TECHNISCHE UNIVERSITÄT
CHEMNITZ

Fakultät für Informatik

CSR-25-06

Development of a Generative Model based Backend of Tutoring Agent

Shahid Ismail · Ummay Ubaida Shegupta · Wolfram Hardt

August 2025

Chemnitzer Informatik-Berichte

TECHNISCHE UNIVERSITÄT CHEMNITZ

Development of a Generative Model based Backend of Tutoring Agent

Master Thesis

Submitted in Fulfilment of the

Requirements for the Academic Degree

M.Sc.

Dept. of Computer Science

Chair of Computer Engineering

Submitted by: Shahid Ismail
Student ID: 579322
Date: 26.07.2024

Supervising tutor:
Prof. Dr. W. Hardt
Ms. Ummay Ubaida Shegupta

Abstract

In this master thesis, which was accomplished in the Research Seminar of Computer Science in Automotive Software Engineering and whose focus is on designing an advanced backend system for an intelligent tutoring chatbot based on state-of-the-art generative language models. This purpose is realized in the creation of a general generative AI infrastructure to personalize and enhance educational experiences, at-scale. This work focuses on the interfacing these models with a chatbot backend to generate natural language responses for questions asked, explanation and personalized feedback. The backend system had to be built out in a way that would allow it to perform important functions like user authentication, session management while users interact with the language model.

Moreover, embedding of the tutoring agent in a front-end interface and providing an emotional avatar was reported to have highly improved user engagement and satisfaction. Another example is the emotional avatar that applied NLP to scan emotions of users and then reply with a more sympathetic manner leading to an enhanced support system which was interactive as well. The thesis also includes the creation of a robust dashboard integration to manage and collect user specific datasets. It gives them personalized recommendations and remarked response considering the data it has been collecting, ultimately leading to improving their learning by acknowledging. This thesis is a valuable contribution to educational technology, as it demonstrates one of the many ways generative AI can make personalized adaptive learning experiences possible. The findings from this body of research not only will contribute to continued development of similar intelligent tutoring systems but are also a further steppingstone towards how AI solutions can improve education in practice.

Keywords: Chatbot, Generative AI, Tutoring System, Large Language Model (LLM)

Content

List of Figures

List of Tables

List of Abbreviations

ITS	Intelligent Tutoring System
NLP	Natural language processing
NLU	Natural Language Understanding
LLM	Large Language Model
GUI	Graphical User Interface
CES	Customer Effort Score
NER	Named Entity Recognition
API	Application Programming Interface
AI	Artificial Intelligence
CORS	Cross-Origin Resource Sharing
GPT	Generative Pre-trained Transformer
PCA	Principal Component Analysis
RT	Response Time
XAI	Explainable Artificial Intelligence
TEL	Technology-Enhanced Learning
EDA	Exploratory data analysis
JSON	JavaScript Object Notation
CNN	Convolutional Neural Network
CUDA	Compute Unified Device Architecture
GAN	Generative Adversarial Network
GPU	Graphics Processing Unit
HTTP	Hypertext Transfer Protocol
LSTM	Long Short-Term Memory
ML	Machine Learning
NN	Neural Network
OS	Operating System
RNN	Recurrent Neural Network
SQL	Structured Query Language
SVM	Support Vector Machine
UI	User Interface
URL	Uniform Resource Locator
XML	Extensible Markup Language

1 Introduction

For the past few years, technology has been incorporated in the education system, which has changed how students and teachers learn and teach. This process started with integrating computers into teaching and learning, followed by the web world and now has come to the era of artificial intelligence. Some of the most exciting possibilities of implementing AI in education is the creation of chatbots, automated conversational agents. These chatbots are gradually finding application not only for administrative functions but also for tutoring students individually. They are specifically developed to help students in their learning process and make it easier, fun, and more flexible per the learner’s requirement. These tools can help manage everyday undertakings, for instance, responding to questions and evaluating scholarly work, hence sparing time for the educators. This enables the teachers to concentrate on the more complex and man-to-man teaching activities that cannot be when done mechanically.

Moreover, with the application of chatbots, students can receive support at any time and from any part of the world due to their availability 24/7. Among the benefits of using the chatbots, it is possible to note the possibility of their further development and scaling. While human tutors can communicate with a certain number of students at a time, chatbots can work with thousands of them simultaneously. This feature guarantees that all the students are attended to, which is quite helpful especially in big classes where the teacher cannot attend to every student. In addition, chatbots can provide convenient learning experiences since they can interact with the students using natural language processing and machine learning to have meaningful conversations based on context. This interaction ensures that learning is fun and not frightening, hence students are encouraged to be more involved in their learning process. Using a chatbot is often cheaper than recruiting new employees or other IT products for education. This gives the education institutions value for their money, mainly when operating under tight financial constraints. the above shows that chatbots can be designed for courses or subjects and therefore, can help students in a way that is relevant to what they are being taught. Such level of personalization is a highly effective approach to learning with a possibility of improving the relevance of the material. This thesis aims to investigate how virtual tutoring assistants can be developed with the help of modern technologies. The objectives include the creation of a chatbot that will be able to offer one on one immediate assistance to students. The combination of the large language models with the state-of-the-art dialogue systems is the goal of the tutoring bot, which aims to provide the best educational experience. Addressing the current challenges faced by educators, this development sets the stage for future innovations in educational technology, potentially transforming how education is delivered and received.

1.1 Problem Statement

Many issues concern contemporary teachers regarding the possibility of providing swift and effective assistance to all students, especially in vast and complex classes. Owing to the ever-high demands placed on educators’ time, it becomes difficult to offer the individual help that is necessary; hence, many students experience a less efficient and stimulating mode of learning. Consequently, such a situation results in low academic achievement and student dissatisfaction. These are some of the problems encountered in the present generation of chatbots. Generative model-based chatbots offer a viable solution to these problems. Specifically, such complicated AI applications can generate text-like outputs and provide non-real-time but instant aid in improving educational content. The use of Artificial Intelligence makes it easy for the chatbots to meet every student's needs and develop the appropriate learning solution for each student. The following are the merits of generative model-based chatbots: they provide instant response regardless of the time or place, promote equality in access to educational materials, especially for students, enhance the level of learning through the interactive system of the chatbots and deliver learning content as per the learner’s need. When included in the educational establishment system, these chatbots will help every student improve their learning experience as they receive the necessary help.

1.2 Motivation

The reason for creating this thesis is to outline and build the backend for a Tutoring Bot that can enhance the quality of education using AI. This spec interview will use the state-of-the-art open-source generative model, making the dialogue flow meaningful and in the proper context suitable for interaction with the students. Evaluating the possibility of utilizing large language models (LLMs) for tutoring is one of the project's objectives. LLMs also have showcased impressive capabilities in grasping and writing human-like text, which is desired in their LLM application in education. Thus, by including LLMs, the tutoring bot would present the student with more accurate and meaningful responses, thereby improving the student's learning situation. Another essential component is the identification and analysis of the affective output. The introduction of emotional intelligence will help design an enhanced empathetic and supportive learning chatbot. This is an excellent feature to help the bot assist students in formulating appropriate responses that may include encouragement and motivation when they are emotionally down, thus helping them achieve better results.

In summary, this thesis will examine the characteristics of real-time communication with Learning Analytics data through a Graphic User Interface (GUI). This way, the data obtained will be analyzed and used by the bot to offer suggestions and information to students, contributing to optimizing their performance and helping them achieve the envisioned academic results. Real-time analytics means that the bot can adjust the feedback it provides to the student depending on the progress the latter is making. The feedback the latter gives is continuously based on the improvement of the course. Technology is closely associated with education nowadays, and automation is one of the significant factors that contribute to the efficiency of the process. These technological advancements, in this case, chatbots, help students with tasks they encounter in their studies, such as presentations, reports and other course documents. These tools allow AI to provide interactive learning methods, making education more accessible and efficient. This thesis is centered on generating virtual tutoring assistants and seeks to construct a chatbot that will immediately assist students.

To this end, the proposed tutoring bot thus aims to improve learning by using current generative models and significant language to deal with difficulties that traditional educators encounter in extending timely and quality assistance. Some potential positive benefits of this thesis are the enhancement of students' interest in the content, the more accessible way students can get information and learning materials, and, most importantly, the individual approach to learning. These contributions can revolutionize the educational practices in the classroom today to the extent that learning becomes more efficient and fun for the learners. Moreover, the usage of intelligent chatbots in the sphere of education can open perspectives for further technological development and could help to improve the quality of education and its access to people all over the world. Thus, by increasing the practical support for and the responsiveness of the learning environment, this work is expected to enhance students' performance and their perceptions of education.

2 Fundamentals

A generative model is a basic method that can provide new real data set strictly from input data. The main sub-technologies are Deep Knowledge Tracing and Reinforcement Learning as the major enabling technologies for this capability. They are basic in the creation and development of ITS, which promotes more unique learning experiences. Intelligent Tutoring Systems (ITS) along with Emotional Analysis Agents are such collaboration frameworks which are intended to facilitate and transform personal learning. These systems incorporate the emotional variation to assess the response according to the learners' emotions to provide a more sympathetic learning experience. Other models, like GPT, forecasting means to produce new data samples belonging to the same type (as text or images) by means of understanding the distribution of a given dataset. NLP can be described as the branch of AI and computational linguistics concerned with how to get computers to process natural languages. Adaptive Learning is a difficult learning process that can be used to replicate and estimate knowledge states to apply educational interventions that are contingent on changes in each learner's knowledge state. Reinforcement Learning as a branch of AI is focused on teaching users about machine learning models, how they work and what they do [1] [2]. The review of the literature discusses both the historical and current stages in the development of AI in education. Starting with rule-based systems first used in the 1980s, these developments came into common usage, especially machine learning, in the last two decades. Some of the major advancements to the field are the creation of the Bayesian Knowledge Tracing and Deep Knowledge Tracing (DKT) allowing for a much better capability when estimating the learning patterns of a student. The recent GPT models are distinguished as a new generation one, with OpenAI's GPT-3 showing the actual capability to generate texts with higher and higher performance that mimic human's writing [3].

The concept of AI in education is a revolution from the usual traditional ways of teaching to more brilliant, innovative, personalized, and more effective methods. This section identifies and explains the key ideas behind AI for learning, the technologies needed, and the research areas in this field.

2.1 Intelligent Tutoring Systems

Intelligent Tutoring Systems are planned to deliver tailored messages of consequent student instruction. These systems use several AI methodologies to tailor learning to each learner, improving their learning process. Emotional features play an essential part in ITS since it enables the system to adapt to the learner's emotions and make the learning process more friendly. Several techniques, such as Bayesian knowledge tracing or dynamic knowledge tracing, are used to define and predict the students' knowledge status and to design individualized and dynamic learning. Another derivative of ITS is the honest time feedback given to the students to assist them in avoiding common wrong approaches and enhance their learning of content. These ITS systems have developed over the years and include different information technologies. Initially, it was rule-based, and it could not correctly deliver personal instructions to a learner. New developments in ML and AI have made ITS systems rather complex and more specific in addressing the needs of learners based on accurate time information. For instance, further analysis in ITS says that the improvement of emotional analysis into ITS can effectively increase the kind and the quality of the engagement of the students, as well as the impacts on the learning process [4]. Also, with these systems, it is possible to incorporate sophisticated algorithms that would enable the measurement and assessment of a student's learning process and provide feedback and suggestions for improvement to guarantee the mastery of the lessons being taught. Flexibility to make numerous paths and adjust the content according to the student's performance degree is changing the existing conventional educational model [5].

2.2 Generative Models

OpenAI's GPT-3 and others refer to generative models, which are the main components of personalized education content. Due to this ability to efficiently learn the distribution of a given data set, these models can create new sample data in text or images. It enables the development of appropriate content within the field of education; students then receive appropriate study materials, which makes the process more effective. Another essential part of AI in education is Natural Language Processing (NLP). Natural Language Processing is the interface between computers and natural language to make computers comprehend and produce natural language. This is important in building interfaces of the AI system through which it can pass messages, understand commands, and respond to student queries.

Many researchers have pointed out that IDP models, especially generative ones, have much to bring into the educational process. For example, the language model GPT-3 introduced by OpenAI can produce text as writing using a stimulus. With this capability, one can design learning material relevant to the learner’s requirements. The literature revealed that integrating generative models into education enhances the learners’ interest and overall learning outcomes as what is processed and provided will be more relevant [6]. In addition, these models can also be applied to tutoring affairs, designing practice problems, and generating explanations of problems, which improve learning ability by making it more animated. One cannot get the opportunity of generative models to modify content proactively according to the student's revolutionary feedback towards educational approaches.

2.3 Adaptive Learning

The technology of adaptive learning organizes educational activities through the application of artificial intelligence to the needs of the learners. These systems use deep learning paradigms to develop and describe the students' state of learning over a given period. Due to their unique approaches, students get timely explanations and instructions depending on the student's performance to help them comprehend concepts in knowledge learning before proceeding to coverage learning. Ongoing assessment of students’ performance enables the provision of appropriate educational support to students, thus enabling them to progress through their learning process more efficiently. From the research studies used in this paper, it becomes clear that the use of adaptive learning methods is very effective in improving the learning outcomes of students.

From the research studies adopted in this paper, it emerges that adaptive learning methods are very effective in enhancing student learning outcomes. In this research [7] proved that there is an excellent potential for adaptive learning systems to enhance the learning achievement of students by giving feedback as per the learning capacity of the student. Also, adaptive learning systems enable identifying and closing learning gaps before the learner proceeds to the next level. Flexible modification of the degree of difficulty and the rate of content dispensing according to the student's outcome is one of the top strengths of adaptive learning systems.

2.4 Reinforcement Learning

Reinforcement learning (RL) is a type of AI that aims at training models to decide about rewarding that come with the decision made. In the context of learning, RL can be applied to create AI instruments that apply feedback to students and help them instantly. RL models are trained through rewards: it will culminate in positive reinforcement and learning results. It assists in the creation of AI systems that can explain, reason, and display the mechanisms of machine learning models hence benefiting the learning of students.

Reinforcement learning has been applied to different educational systems. For instance, there is a study which proposes [8], in which the authors prove that RL could help create ITS with the capabilities to give the appropriate feedback for students. Overall, the study involved the enhanced paradigm of Reinforcement Learning and based on the findings, it was determined that implementation of the RL based tutoring systems among students an increase the students' learning outcomes effectively by using real-time feedback from the system about the student's learning progress. RL models are ideal for the education application since they can learn and enhance their performance thereby getting better through interaction with students.

2.4.1 Transformative Potential of LLMs in Education

Revolutionizing student engagement with educational content and support systems can help the generation of coherent if predictable human sounding dialogue supports more dynamic conversational learning opportunities. LLMs, on the other hand, may offer more in-depth explanation and answer difficult questions even simulate talking to famous people.

In fact, an investigation "Conversational Intelligent Tutoring Systems: The State of the Art" [9] shows that intelligent conversational agents can give more accurate and personalized feedback to help students think better during their online learning process, which leads to increase in student engagement as well as improvements on learning outcomes. In turn, incorporating LLMs into these systems could take their capabilities and effectiveness to the next level by offering richer as well as domain-specific responses.

2.4.2 Benefits of Multimodal Data and Affective Computing

Multimodal systems bring in information from different types of data to help understand more the student requirements and decide on a response accordingly for example, by looking at facial expressions and vocal tone we can see that the student may be feeling confused or frustrated, so it is evident he needs a little help.

We will discuss about a paper here called "Affect-Aware Conversational Agent for Intelligent Tutoring of Students" [1] which proposed integrating affective computing into ITS. Identifying the feelings of students and responding differently to these states can foster a supportive learning environment that is useful in preserving motivation and engagement.

2.5 Ethical Considerations

Explanations and reasoning are required when it comes to the decision-making of AI systems to gain the trust of users. They should embrace techniques that help them eliminate biases from machine learning algorithms to ensure all students will not receive unfair help in education. Regarding the privacy and security of students' information, it remains crucial while designing and implementing AI tools for learning processes. Therefore, it is imperative to discuss these ethical issues that threaten the integration of AI technologies in education. It should also be noted that ethical issues are particularly relevant when creating and implementing AI in learning. For instance, used transparency and fairness to describe the various AI systems. The paper called for warranting fairness in the AI systems to enhance the public's trust in the technologies for use in education [10]. Moreover, the proper personal data management and protection of students' data are crucial to avoid unnecessary intervention and the improper use of AI technologies in learning processes. AI in education must be regulated by ethical standards and norms to prevent misuse.

2.6 Evolution

The application of AI in education has advanced for several decades. Further, moving from the rule-based systems of the 1980s to the modern neural network approaches, AI has continuously developed and improved to provide more efficient and individualized solutions in education. A relatively recent development in making the modelling of students' learning dynamics possible is using Bayesian Knowledge Tracing and DKT. In the previous year, the more sophisticated generative models such as GPT-3 by OpenAI showed high possibilities to generate human-like text and deliver individualized educational material. Education on using artificial intelligence has undergone tremendous growth and is characterized by milestones. For instance, the inclusion of Bayesian Knowledge Tracing in the mid-1990s transformed the flow of modelling and predicting the ready knowledge of students. In the same way, the emergence of Dynamic Knowledge Tracing (DKT) in the 2000s enhanced better modeling regarding students' learning processes and assist them by offering timely feedback and advice [11]. More recent generative models such as GPT-3 have shown much promise of personalization and adaptation of educational content, amongst other achievements. AI technologies are developed at an incredible pace, which allows suggesting new paradigms in learning processes.

2.7 Future Directions

In general, numerous prospects and continuations can be expected for the future of AI in education. This field should also develop and publish new theories like advanced knowledge tracing, reinforcement learning, and generative models to create better and more individual tutoring agents. More development on the models of AI will enhance positive production of meeting the different needs of learners. It is, therefore, required that principles and policies of ethical requisites are set to negate the misconceptions that come with AI in education. When deploying AI tools, the possibilities of further education transformation will be opened better to meet students' individual requirements and needs. Future studies on the application of AI in education will cover the following issues. For instance, knowledge tracing improves learning, reinforcement learning enhances decision-making, and the generative models make newer tutoring agents. Have conducted a study [11] to show that these techniques' effectiveness can be enhanced if integrated to improve student learning paradigms. Besides, as the models used in AI systems are further developed, meeting the requirements of diverse students will be easier. It is also necessary to define the significant and extensive ethical standards and policies that would help to prevent abuse and misuse of advanced tools in learning [10].

In summary, A generative model is a straightforward system capable of creating realistic new data sets from input data alone. Key technologies like Deep Knowledge Tracing and Reinforcement Learning support the construction of intelligent tutoring systems, which enable personalized and adaptive learning. Intelligent Tutoring Systems with Emotional Analysis Agents improves diverse education while implementing an emotional analysis to adapt the response to learner's feeling because of empathetic learning. Generative models like GPT produce new samples like text or images based on understanding the statistical distribution of the input data set. NLP is an integration between natural human–computer language communication. At the same time, adaptive learning applies the registration of deep neural networks to determine a student's understanding state with a view of constituting timely educational interventions. At the same time, AI learning assists users in learning and reasoning about machine learning models, and Reinforcement Learning aids in translating the models to the users. AI development in education can be traced back from the rule-based systems of the 1980s to the deep learning forms of the present period; With the machine learning changes like the Bayesian Knowledge Tracing and Deep Knowledge Tracing, education assessment enhanced the learning dynamics among students.

3 State of the Art

As for the state of research in the area associated, it concerns the possibilities of employing recent generations of AI technologies, including generative models such as LaMini Flan T5 248M [12], in ITS. These state-of-the-art systems incorporate NLP along with ML and are known to issue individualized and experiential learning methods. Recent advancement in the generative models allows ITS to be capable of interpreting and producing response to these set of student inputs in detail and precision to offer student-specific feedback and assistance, if necessary. In addition, ITS studies have shown how AI-enriched might enhance students' participation and learning gains by personalizing the content. Also, including emotional analysis in such systems enhances the possibility of providing kind and friendly communication, improving the learning process. Nevertheless, some of the crucial issues that researchers tackle even today include the reliability of the responses obtained, the issue of maintaining near personification, and the ethical issues surrounding data security and impartiality.
A paradigm shifts from personalized, adaptive, and engaging learning experiences thanks to the recent breakthrough in AI technologies, sophisticated models can now create a customized educational content through real-time feedback and personalized learning paths based on each student's needs and interest. These advances have the potential to confront perennial teaching and learning issues in education including student engagement, individualized instruction targeting proficiency gains, feedback efficiency for ongoing improvement.

Teacher education seems to be one of the primary beneficiaries from approaches reasoning on active AI structures, in which generative AI systems play central role in designing adaptive learning environments accommodating differently capable learners. Studies show AI tools become more engaging and effective than other educational software because they deliver personalized feedback, adjust to the individual user's needs in real time. For example, in e-learning settings machine learning algorithms can personalize the courses by creating personalized learning paths that will lead to an increase of student performance and engagement.

Broad perspectives are provided by comprehensive reviews of applications across various facets of generative AI in education, drawing lessons and design implications from multiple studies. Elements such as AI-authored quizzes, tailored study guides and adaptive learning platforms have provided the nuts-and-bolts for better education versions in all corners. AI can also help the student engagement and comprehensions by integrating skull data, that would bring AI out of its traditional category to provide real time predictions about a student's learning outcomes [13].

Nonetheless, it still has some challenges as well in terms of data privacy, how scalable AI solutions are and is there any sort of ethics being considered while putting use or simply the effect from using an AI on Education. Importance of confronting these challenges to unlock and support the transformative potential of generative AI for education. However, researchers will need to develop further AI systems that are stable and scalable for the technology not only to raise educational standards but also to guarantee data security and ethical usage. Despite all this uncertainty, the current level of investigation means that generative AI for education might have a great future ahead. The contributions of this thesis are directly connected to building on the best attributes and mitigating typical challenges faced by current approaches to lay down foundations for advanced AI-driven educational support systems that can deliver personalized, inclusive learning experiences suitable for all students [14].

3.1 Intelligent Tutoring System with Generative AI

This provides a general outlook on four groundbreaking articles on Incorporating ITS with Generative AI. All the papers are reviewed and discussed in terms of their research findings, methods, and implications for revealing the state-of-the-art and prospects of this innovative learning tool. The present context of ITS and generative models is significantly developed, mainly supporting using artificial intelligence (AI) in constructing intelligent learning environments. ITS is intended to give students information as well as feedback based on the student's needs. Approaches like BKT and DKT are used to model students' knowledge states and adapt teaching methods to these models. Support for these systems involving emotional operation is identifying the learners' emotional state to increase engagement. Educational articles and content are created using generative models. These models can provide regard to teaching and self-tutoring episodes, generate problems to bolster practice, and provide ways to explain perspectives and concepts, thereby enhancing the learning experience. The employment of these progressive elements of AI in ITS and generative models is presently the most progressive advancement in educational technologies that can improve students' performance and enrich their education experiences. This provides a general outlook on four groundbreaking articles on Incorporating ITS with Generative AI. All the papers are reviewed and discussed in terms of their research findings, methods, and implications for revealing the state-of-the-art and prospects of this innovative learning tool.

In the current context of learning organizations, providing favorite education for students, and dealing with several different limitations in terms of resource availability are quite hard. Professional educators face what they consider a huge challenge in the form of student enrollment ratios that have been on the rise in the recent past and which are known to compromise the quality of education offered in class. In this case, students can require instant answers to their questions to stay focused and understand what the instructor is teaching them, especially when in large classes or online classes, the instructor cannot spend much time with each student. This is the rationale why the use of a multiplatform chatbot as an online tutor for university courses is being proposed. The developed chatbot is named as Infobot and its purpose is also to help in reducing the burden of the teaching faculties and help in responding to the normal set of inquiries which students may have in one or the other way across different platforms like Telegram, Facebook Messenger, and Line. This automation makes it possible to provide academic assistance as soon as one clicks for it and is available 24/7 that may usher a new era of learning since the traditional model is being adapted to in this way. Minimizing the time educators spend on rituals of questioning would help the educators to improve on the teaching methodologies and directly address the students on important matters more often improving the quality of education [15]. So, the growth in enrollment in postgraduate facilities is a large new problem every educational institution faces, especially in the sphere of timely and efficient tutoring and counselling services. This growth puts stress on conventional advising solutions, which might result in the delivery of services to students. It may also affect the levels of customer satisfaction and student performance. The problems mentioned above were supposed to be solved with the help of an AI-based chatbot as it offered quick, efficient, and, as far as the machine learning algorithms allowed it, consistent support, thus strengthening the educational pillars in response to the increasing student population [16]. The requirement for effective and efficient search tools in education environments is growing significantly due to the rising number of education resources. In the context of Dublin, where various educational institutions and there are lots of diversified resources in and out of the internet for students, the problem of searching rapidly for exact and pertinent information slows down their learning process. This research is intended to improve the way educational resources are searched for, from the current fundamental techniques to more advanced methods involving machine learning. By creating a chatbot that would be capable of processing and implementing complicated queries for finding specific material, the thesis targets to increase the level of student satisfaction and the speed of their learning process in general [17]. Although this paper originates from the quest for knowledge-based AI and generative AI for shifting new paradigm on educational settings through the concept of personal learning environments and auto-generation of content. They expand the opportunity of experiencing improved, more open, and effective educational processes relevant to the students' and their needs. Therefore, the purpose of this reasoning is to identify how the employed AI technologies can help and enhance the educational process while catering to as many learners as possible [18].

The development of Infobot was structured around several key phases: requirement analysis, design, implementation, testing, and deployment. Firstly, it was necessary to analyze the requirements systematically and to realize the kind of requests students used to make and which means of communication they would use most. The design phase meant the development of a chatbot framework which could reside in multiple messaging channels yet refer to the uniquely updated authoritative knowledge base fed by the faculty. NLP tools were incorporated to facilitate intelligent understanding and answer the questions posed by the students. Implementation was done by coding the chatbot, which was developed using Dialogflow by Google, an NLP tool. This platform was chosen because of its resilience and compatibility to grow with the business. The testing phase was split into two parts: the first was the test of response accuracy with the help of automated testing tools, while the second was the user acceptance testing, done with the help of a small group of students and faculty. Last, the chatbot was introduced across the university; any operational problems that may arise were sorted, and the knowledge base was updated regularly [15]. This methodology not only helped the development team to guarantee the operability and ease of use of the chatbot but also assisted in making sure that the potential questions that students are likely to ask are well accommodated in the functionality of the chatbot. The research strategy utilized in this study included the conceptualization, development, deployment, and assessment of an AI chatbot called Vivian. This chatbot was designed to help the postgraduate students to make more effective academic-related decisions, such as, curriculum choices and planning of study timetable. Vivian was created with the help of the latest artificial intelligence, which involves the ability to comprehend and perform actions by using natural language processing algorithms that allow the application to communicate with students properly. The (Fig. 3.1) explains general workflow and organization of interactions of a student with a chatbot that is intended for processing of different aspects of the conversation and questions from a user [16].

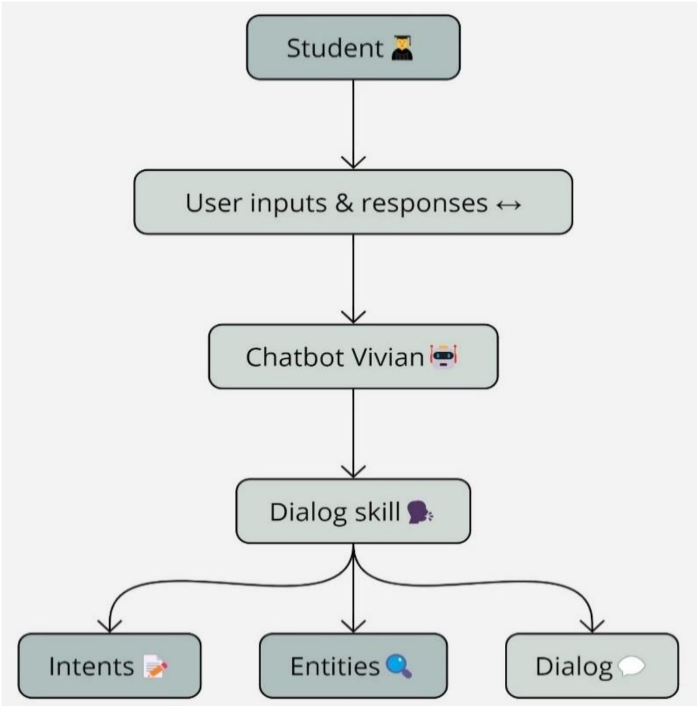


Figure 3.1: Fundamental Architecture of the Chatbot [16].

The specific approach used in this work includes various phases related to the structure of the proposed chatbot that is oriented on serving as a search engine aiding in searching educational resources in Dublin. The current chatbot mainly works based on machine learning algorithms to understand user queries and generate the most suitable responses. The training occurs after creating a set of common questions a student might ask together with relevant sources that it should search through with the purpose of improving accuracy and runtime. The chatbot is embedded in modern students' platforms like university portals and mobile applications for easy reach. Testing is a continuous process where real users' data test the system in different cycles to improve the system. The (Fig. 3.2) illustrates the structure and operational flow of a chatbot system that incorporates NLP/ML with the user and an application backend for data processing and reply. Below is the analysis of the parts and how they operate in the software [17].

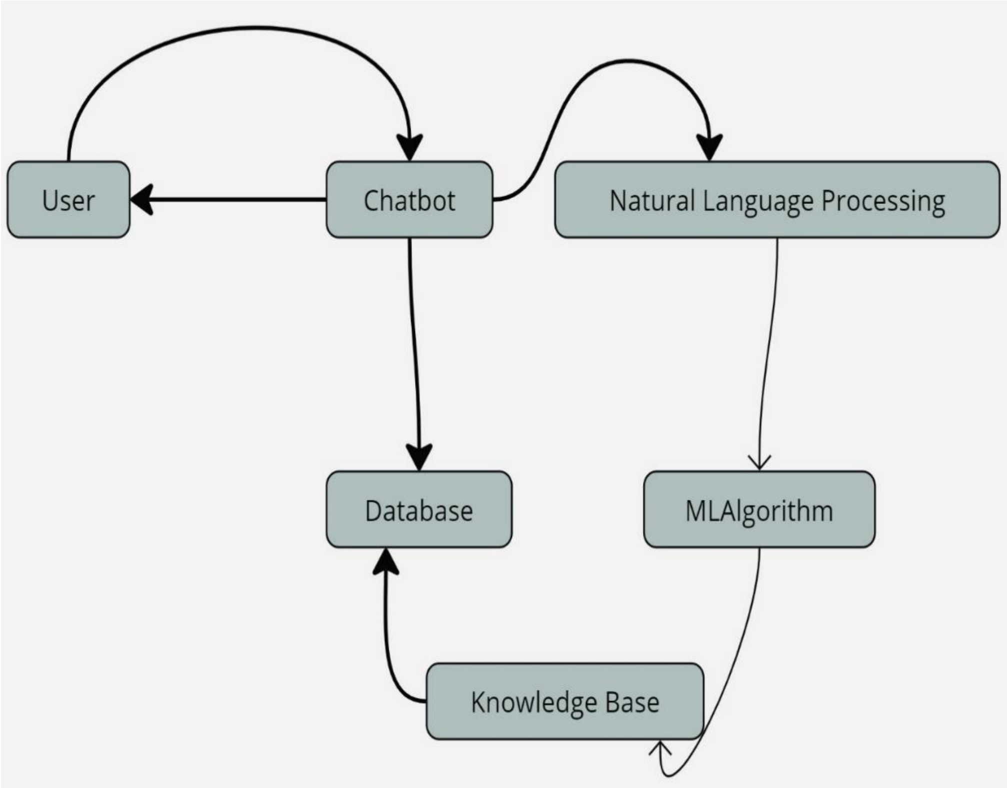


Figure 3.2: Chatbot System Architecture with NLP and ML Integration [17].

This research uses literature review and analysis technique to survey various studies and application of Knowledge-Based and Generative AI in education. The steps include a collection of data from the academic databases, journals, and case studies, where the authors described the application and the results of AI technologies in education. Also, survey questionnaires and focus group discussion are employed to interview experts, and case studying aids in determining various adopting techniques and real-world problems in the deployment of these technologies. The findings of this literature review are sought to give an understanding of the current state of AI in education today, its developments so far, and possibilities in the future [18]. The (Fig. 3.3) illustrates the advance educational technology system for improving the learning process with proper amalgamation of different data and system components. These are the knowledge types that are transformed into personalized, domain specific, and general knowledge which can be employed by a conversational agent. This agent is the most important among all the suggested ones because it communicates with the LSM to store and manage the educational material or track learners' performance.

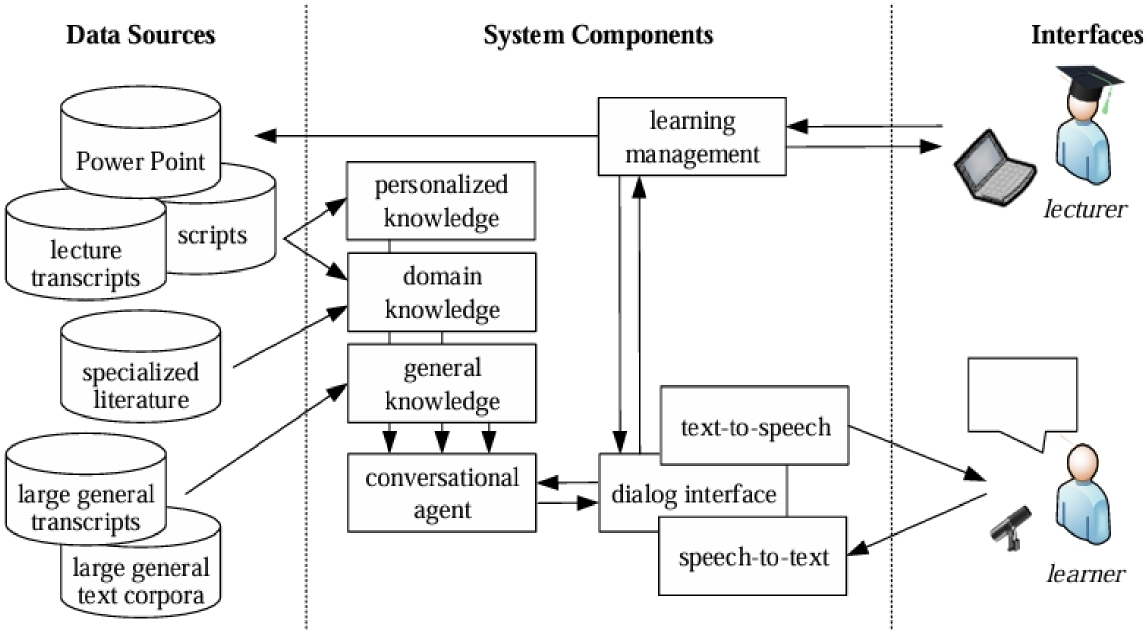


Figure 3.3: Integrated Learning System Architecture with AI Interfaces [18].

The efficiency of the response to students' questions, which was established after the launch of the Infobot system, was very impressive. The assessment of usage information also showed that the chatbot could effectively handle more than 80% of the inquiries without the help of live employees, up from the pre-implementation stage. According to cumulative data collected by the satisfaction ratings that were conducted among students, students were grateful for the instantaneous feedback and that the academic support services were accessible all day, every day; however, a specific emphasis was noted that although the chatbot for course schedules, assignments, and examination protocols was considered helpful for repetitive queries. The Appendix likewise revealed that simple questions posed by students had been minimized substantially, which enabled the members of the faculty to have more hours to prepare for their lectures and more time to enhance the content of the courses. Also, the insights derived from chatbot usage could be beneficial for academic staff as those insights highlighted areas where students might have difficulties or draw their attention to certain areas of knowledge. Further, those insights could be helpful in curriculum changes and instructional approaches. This paper, therefore, shows the effectiveness of the chatbot with the goal of promoting the provision of students' support and boosting the accomplishment of tasks by educational institutions. Figure 3.4 illustrates a conversation in an instant messaging application between a user named “fez” and an instant messaging bot named “infobot.” The former asked the latter for a graph of Ethernet. The bot quickly returned a layout for the Ethernet connections, such as computers, switches, and servers, to help explain a network. The interface entails elements such as message input fields and emojis, as can be seen [15].

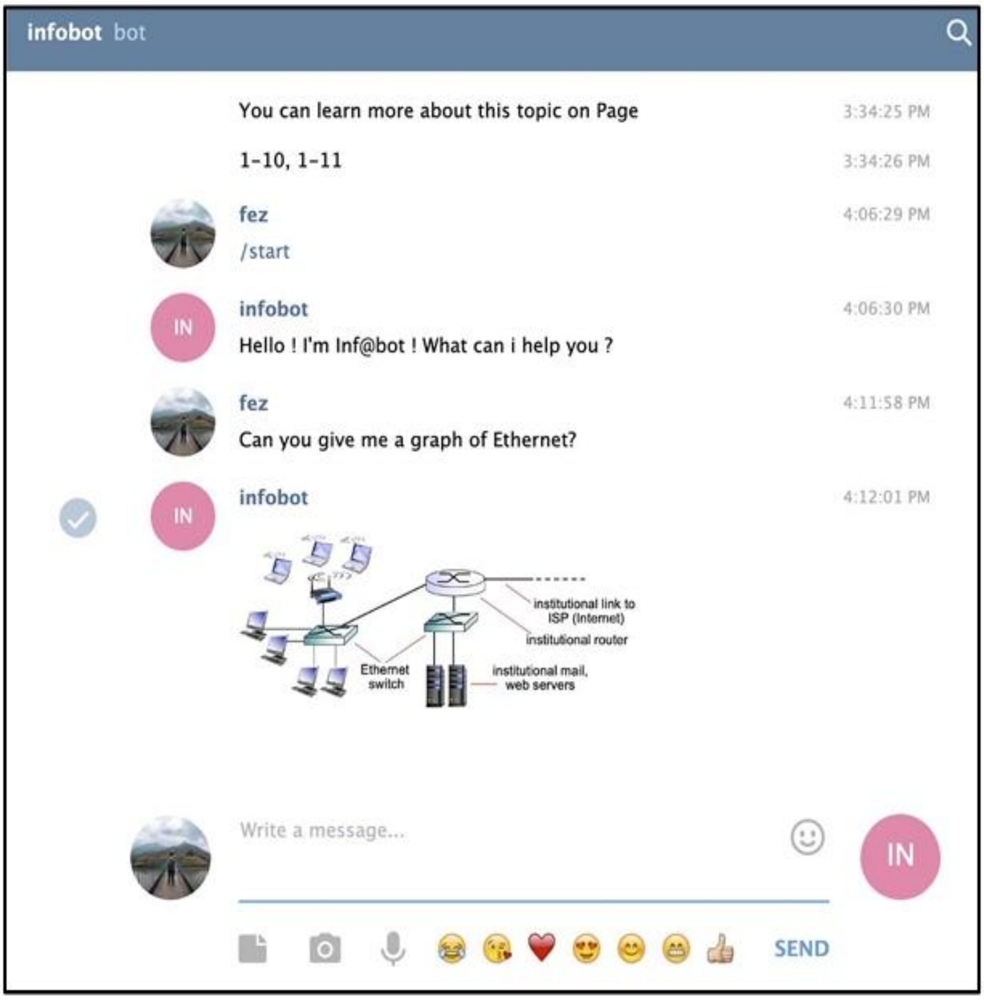


Figure 3.4: A multiplatform chatbot infobot serving as an online tutor [15].

Vivian significantly cut down the time it took to attend to students' inquiries concerning academic advising and ensured that support services for the postgraduate students were available round the clock, thus having solved one of the biggest logistical issues in postgraduate support. But the survey that was conducted among the students revealed that although the chatbot served well in handling routine questions it was not efficient enough in fulfilling specific and complicated counseling requirements. Such research is intended to elucidate the possibilities and drawbacks of using modern AI technologies to mimic the human advisor's subtlety. This part of the (Fig. 3.5) displays a visual representation of a chatbot's decision-making process or workflow, commonly used in designing and managing chatbot interactions. The figure includes various nodes, each representing a step or function in the chatbot's conversation flow [16].

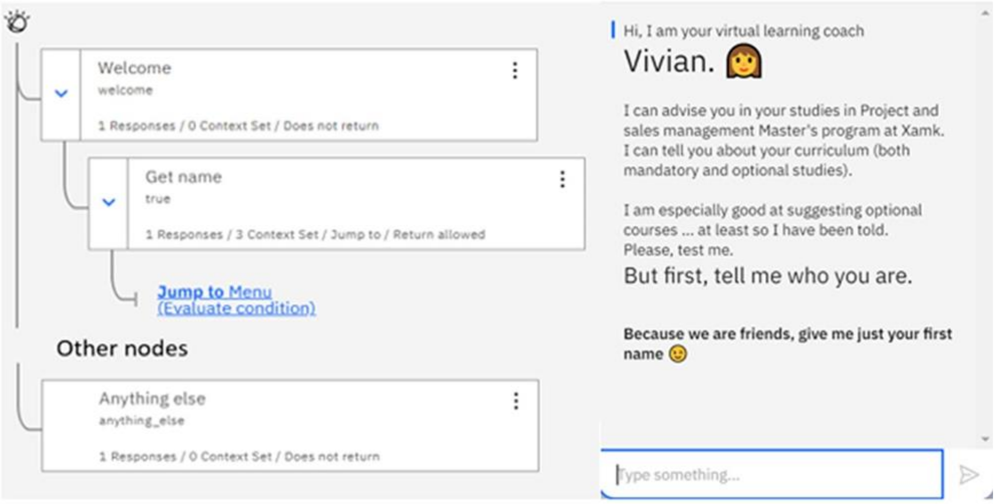


Figure 3.5: Greetings and Retrieve the names of the dialog skill's nodes or dialogs and their results [16].

See (Table 3.1) share actual or proposed user satisfaction measurements and comments on an AI-based curriculum planning chatbot. Customer satisfaction has been depicted to be very high with customers giving a rating of 4. 3 out of 5, and this means that users are per se satisfied with the help of the chatbot. Customer effort score was 2 which is very low. 4 of the criteria out of five therefore the respondent found it easy to use the chatbot [16].

Table 3.1: User Satisfaction and Feedback on AI-Based Curriculum [16].

Metric	Score	Feedback Description
Customer Satisfaction	4.3/5	High satisfaction with the chatbot's assistance
Customer Effort Score	2.1/5	Low effort required to use the chatbot
Net Promoter Score	+38	High likelihood of students recommending the chatbot
Total Participants	57	-
Positive Feedback	93%	Majority found the chatbot helpful for curriculum.

The research suggested better and faster search results than conventional search approaches preferred by students with the help of Chatbot. The interactivity helped the machine learning model to refine its search analytics, which in turn raised the user satisfaction rates. But problems persisted in processing uncertain inquiries and delivering precise and contextually relevant information, which is particularly crucial for educational applications. The study's outcome indicates that despite great benefits arising from implementing machine learning algorithms in the functionality of search, constant improvements are needed to meet the diverse needs of educational search

[17]. The (Fig. 3.6) shows a conversation between a user and a chatbot where user proceeds by asking chatbot its name to which it responds, saying: ‘Hi I’m Bot your helper for Institutional queries in Dublin.’ After this, the user asks about the ‘DBS QAH’ where the chatbot informs the user that this stands for the Quality Assurance Handbook of the Dublin Business School and gives the user a link to find further information. The user also poses questions about how they can they get the timetable of their classes. To know more about personal timetables and personal devices, the chatbot suggests downloading the Outlook App and gives a link to the service desk. This brings out the factors of depth and scope of the chatbot in its response and in explaining possible response forms depending on the action taken by the user [17].

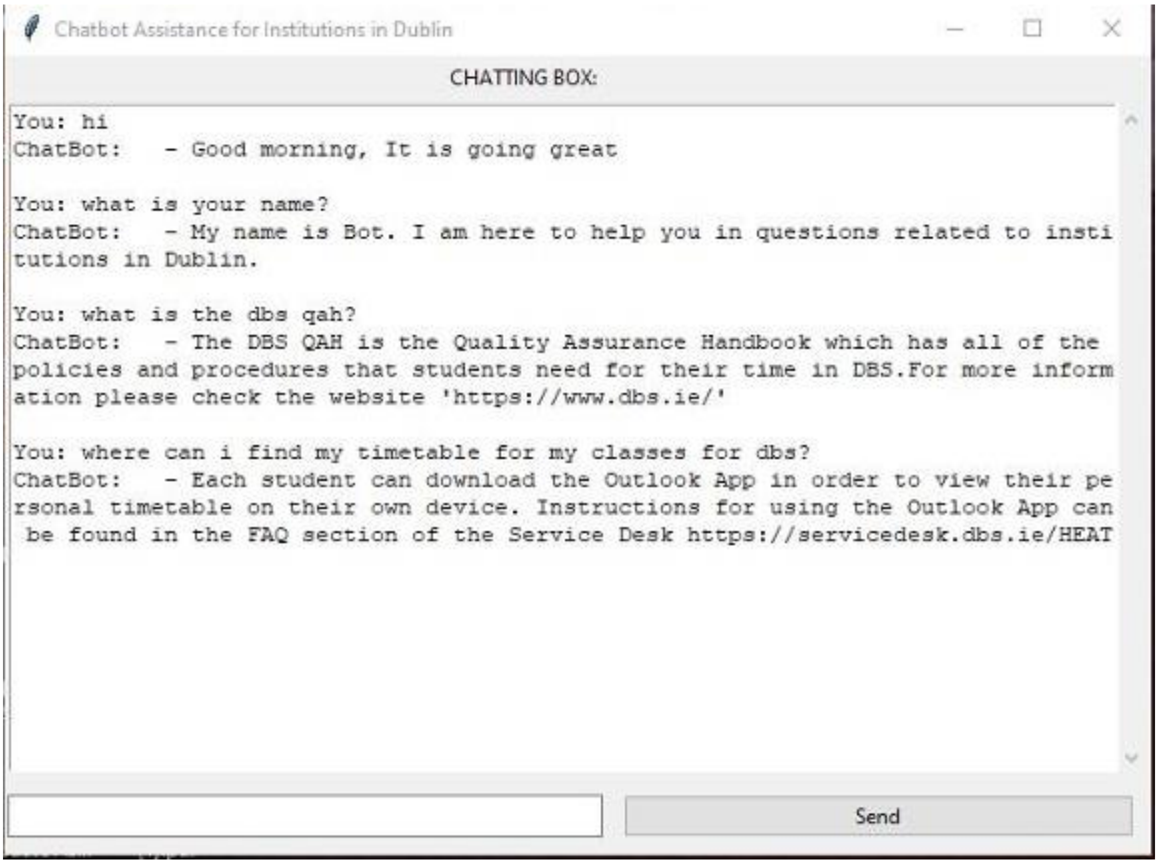


Figure 3.6: A GUI for an educational machine learning chatbot [17].

The Table 3.2 provides an overview of a chatbot’s performance in educational assistance, evaluated by 75 participants. It highlights an 88% accuracy in relevant information retrieval, a 4.7/5 satisfaction rate in voice interactions, and a 4.5/5 score for overall usability. Additionally, 90% of users gave positive feedback on the chatbot's effectiveness. A fitting title for this summary might be "Chatbot Performance Evaluation in Educational Assistance," reflecting the chatbot's proficiency and user satisfaction [17].

Table 3.2: Educational Chatbot Effectiveness Metrics [17].

Metric	Score/Percentage	Description
Search Accuracy	88%	High accuracy in providing relevant educational information
Voice Interaction Satisfaction	4.7/5	Users highly satisfied with the voice interaction capabilities
Overall Usability	4.5/5	Generally positive user experience with the chatbot
Positive Feedback	90%	Majority found the chatbot effective for educational searches
Total Participants	75	Number of users who participated in the evaluation

Based on the results of the study, it is concluded that knowledge-based and generative AI technologies have a great potential in the development of effective educational practices where students receive individualized approaches to learning and content is generated to address the students' needs. But at the same time, the study reveals a few issues such as the relatively high costs of implementation, the requirement of extensive input data for effective AI training, as well as several ethical concerns connected with data protection and the self-learning capabilities of AI tools. However, the general potential of AI in education is highly significant, as numerous successful cases indicate augmented effectiveness of learning and enhanced students' interest in the result of integrated innovations [18].

The ultimate purpose of Infobot was to supplement the learning process and offer a ready and quick source of the desired information to students, relieving the burden of the professor. This objective correlates well with general educational objectives of addressing the concerns associated with access to learning assistance, particularly under conditions where there are relatively many students in classes or when learners are taking online courses, often do not have an opportunity to meet our instructors face to face [15]. The general goal was to identify the potential of offering an AI-based system as an augmenting solution to conventional academic counseling with the hope of increasing the efficiency in an advisory capacity without compromising on the quality of the services provided. The goal of the thesis was to evaluate the reliability of utilizing artificial intelligence to manage different levels of the academic advising activities and to find out the cases when human-interference is Unavoidable [16]. The main aim of this work was to design a such an innovative chatbot that would significantly transform the process of searching for tutorial materials in Dublin through machine learning. This involves more than enhancing the effectiveness of the search as well achieving the relevancy of the results derived to the context and academically beneficial. The discussed purpose is more general in that it aims at determining the applicability of AI in the enhancement of the delivery and use of educational materials, a knowledge that can be beneficial in other areas and contexts of education [17]. Therefore, the main rationale of this study is to establish the applicability and effects of knowledge-based and generative AI in educating. In this way, the study seeks to offer the educators, policymakers, and technologists with the clear conceptual framework of AI's potential for improving the educational practices and results. The more general aim is to intensify the development of new AI technologies and investments in the sphere of education, to enhance constant changes and development in educational approaches and tools [18].

Even though Infobot has proven to be rather effective, the study revealed a significant limitation of the model in capacity to address elaborated questions that may need, for example, utilization of the employees' emotional intelligence or a sophisticated understanding of the learning process. This limitation leads to the common issue with many applications around AI in education. The inability of designed systems to deal with not only the factual questions as input but also the emotional and contextual data present in students' interactions [15]. The study also points out a major weakness of AI systems in relation to the emotional intelligence and more complex forms of advising where human touch is very much needed. This gap reveals the present status of AI in comprehending the instinctive and subjective factors of the human social engagements [16]. The chatbot managed to enhance the process of search; however, the experience put into focus a major flaw of AI: the total lack of contextual understanding as well as the contextually encouraged questions of the educational involvement. This limitation brings the broader concern about many AI technologies in the context education – inability to work in context [17]. Finally, the study unveils a major gap in terms of research when it comes to the consequences of AI technologies in education as they concern learning performance in the long run. Moreover, other important aspects like, ethical concern, privacy and autonomy while using AI in learning environments are still hidden and unrecognized in the literature still [18].

As a result, future research could investigate incorporating higher levels of AI within the chatbot, for example, optimistic emotional intelligence or a better understanding of the context. If such improvements were to be made, they would allow the chatbot to carry out even more sophisticated educational support tasks including identification of the students' affective states and offering more elaborate academic assistance. Moreover, the effective continuation of AI working together with developers, educators, and students will require constant efforts to develop AI applications that will suit the manifold and constantly developing conditions of education settings. They both give and take, refining AI technologies such as Infobot with the goals and issues of education to make those technologies more suitable for learning and teaching aids [15]. Further research could be aimed at the development of the function of detecting the emotional state of the student as one of the capabilities of the chatbot. To implement more effective AI models that could learn from a wider variety of human interactions could also be of assistance in this case, quite probably through using feedback loops whereby the human counselors could assess the chatbot's responses and modify them where necessary [16]. Based on the results that have been provided in this research, future research in the application of machine learning in sport analysis should look at the extension of contextual analysis in the models. This could involve, for example, a specification of the need to use better natural language processing techniques by which, from the command issued, the system can deduce more in relation to the query and pattern responses in real time. In addition, the continuation of the connections with other educational databases and sources could enhance the quality of the given responses and the containments of the contains of the chatbot [17]. In place of these gaps, the paper suggest that long-term studies should be set up to evaluate the effects of AI on various education Type Outcomes are proposed for future research to be done to establish ethical policies and principles of AI use in education. This would include the combination of engineers of AI, teachers, ethicists, and policy makes to have to enhance the best practices in the channeling of AI in education [18].

Explores a real-life example of how the integration of AI improves the educational field's efficiency and results. This specific case indeed illustrates the argument of the thesis statement that AI can enhance support services in education which largely demonstrates opportunities and difficulties. Thus, it can be considered as one of the main components of the thesis, which presents the possibilities and applications of AI technologies in the academic environment and can be used as a grounding for the discussion of the further perspective on AI technologies' use in education [15]. Thus, the discussed paper serves as a critical concern as far it shows how AI has been adopted into postgraduate education support systems. It provides actionable information regarding the application of AI in delivering services in higher learning institution and bring to the fore a major challenge of developing AI-based systems that can cater for the diverse needs of the learners [16]. Specifically, the use case of Education Bot describes a case of how technologies based on Artificial Intelligence work now, how they can be used to solve practical problems of increasing efficiency and improving access to education, which correlates with the materials of the thesis and focus on the efficiency and accessibility of education. The specifics of this case allow for the thesis' more general discussions of AI to be grounded in the actual use of AI in schools, the positive and negative effects of which are demonstrated here as well [17]. This paper is a key component of the theoretical and empirical background for the analysis of the function of AI in education is developed here. It provides evidence to the conclusions stated in the thesis and provides a general introduction as well as more comprehensive descriptions concerning the application of generative AI to address educational issues [18].

3.2 Adaptive Reinforcement Learning in AI-Based Systems

This section provides an elaborate analysis of four main studies on the Application of Adaptive Reinforcement Learning in Systems Comprising of Artificial Intelligence. Aimed at today's advancements and the prospects of the use of adaptive reinforcement techniques turned out to be the principles regarding the innovative methodologies and crucial results of each paper, as well as the discussion of the broader impacts of these findings.

The need to improve the effectiveness of the AI tutors and to introduce innovations such as, for instance, prompt engineering and deep knowledge tracing strategies to train the AI tutors is based on the modern tendencies in learning that more and more focus on learners' individual needs. In traditional educational patterns, the speeds and ways of learning are not taken into consideration leaving the slow absorbers in the group behind thus increasing the inefficiency of learning. AI tutors specially conceived to replace human tutors in terms of adaptability in their learning process have also some advantages and some drawbacks in the best deep and large scale of individualization of learning. This research will seek to fill this gap by using state of the art recognition and response methodologies of AI in identifying and attending to students' individual needs. The effectiveness of incorporation of prompt engineering improves the quality of interaction between the AI tutors and students in the manner that positively impacts personal educational contexts. While deep knowledge tracing established a stable strategy for analyzing the student performance over the course of learning; the AI in this solution can proactively diagnose and respond to the specific student performance pattern at any point in time accurately [19]. Due to the pandemic education has been a big challenge and at the same time it has opened a lot of opportunities that require education sector to come up with new approaches to impart knowledge. This paper is based on the idea that generative AI can open a new era within the educational environments, contributing to making the processes of teaching and learning more effective and enjoyable. Thus, if AI is integrated into the educational frameworks, there is a good chance to get rid of some chronic problems like accessibility of education, individual approach to learning, and final assessment. The motivation behind this study is to utilize generative AI abilities as a way of making education more communicative and simultaneously more sustainable in terms of meeting students' needs and future educational requisites [20]. This paper aims at discussing the emerging conversation concerning the application of AI chatbots, particularly ChatGPT, in education. Being more embracing of learning technologies, education emerged as one of the most actively implementing spheres when it comes to AI trends, making it possible and easy to discuss the chances and threats of disruption of the conventional learning paradigms here. As a result of the overly optimistic and pessimistic stances taken regarding AI as a progressive and revolutionary element in the educational system or as the opposite, a threat and a source of negative influence, this study aims to identify and categorize the widespread range of users' sentiment and experiences on interactive forums such as Reddit. The aim on the practical level is to offer a more sophisticated view of people's perceptions, which can help to improve the corresponding AI approaches to learning [21]. The motivation for this research stems from the desire to improve the quality and availability of tutoring for higher learning facilitated by conversational agents. The situation unfolds as educational requirements become more challenging and varied, and more efficient tutoring models are needed. AI agents especially those mimicking human behaviors would be the one to help in assessment and learning to address barriers of education to make learning effective to everybody [22].

The research uses a highly complex experimental architecture that involves the use of both qualitative and quantitative research methods to evaluate the performance of improved AI tutors. The given methodology is based on the use of AI tutoring systems with the elements of prompt engineering and the application of deep knowledge tracing in various contexts of education. The students who use these systems are also observed for over a semester period and results are taken on how engaged they are, performance profile that they exhibit as well as the overall results against those students who used the traditional tutoring systems. The study employs the use of machine learning methods on the gathered data targeting on response rates of the students, enhanced efficiency in answering questions as well as enhancement in the difficulty levels of the problems. This approach also verifies the increase in effectiveness because of the AI additives, as well as the specifics of students' absorption of text versus visuals, usage of tutoring prompts, and learning path modifications [19]. Qualitative research methodology is used in the study to examine the use of generative AI tools in an academic environment. The quantitative data is collected in experimental manner by comparing how classes that are taught with the help of AI tools work and how they perform with the traditional teaching methods by considering the aspects of student performance and other indexes. Large interviews with students and teachers have also been conducted and questionnaires were presented to get an idea of the perceived advantages and difficulties of implementing AI. These procedures facilitate the identification of the potential of generative AI to transform different segments of the learning process for students, as well as optimize the functioning of educational institutions' administration [20]. To address the issues of diversity, the present study used natural language processing techniques to obtain the views on and experiences with the educational technologies Discussion threads were selected from several subreddits that are related to ChatGPT. The method of sentiment analysis is used to determine the quantity of positive, negative, or neutral attitudes of the users and thematic analysis to reveal the major issues and topics. Thus, the work jointly combines quantitative sentiment analysis with qualitative thematic analysis to provide a more nuanced view of the general public's opinion on the involvement of ChatGPT in education [21]. This entails creation of an interactive tutoring agent under the RASA architecture acknowledged for its versatility in NLP. Discrete training data are generated from the typical student queries as well as the course content related to the computer engineering programs at Chemnitz University of Technology. Quantitative appraisal of the tutoring agent is as well done through scales of simulation and live pedagogical lessons with a student to determine its effectiveness in actual learning environment [22].

The research conducted in this paper shows increased learners' achievements that are attributable to AI tutors that have been expanded with prompt engineering and deep knowledge tracing, coated efficient, better recall, quicker grasp or concepts, positive attitude among students when it comes to complicated concepts. More so, the results indicated that dynamic involvement is enhanced when engineering prompt as students initiated more communication with the tutor to complete the tasks assigned to them. Deep knowledge tracing also helped in the early identification of students who were likely to struggle in a subject by providing suggestions for timely compensatory interventions which largely boosted students' performance. These results present the effectiveness of continuous improvement of educational tools with the use of complex AI solutions not only for student's profiling but for overall improvement of educational outcomes and students' satisfaction [19]. These insights of the research reveal that generative AI enhances learning through altering its method and content to reflect the learner's parameters hence enhancing the learning process mainly, from the standpoint of participation and efficiency from the learners or students. Some routine clerical chores are recorded less often thereby relieving teachers of time to ponder how to develop ideas of teaching. However, the paper identified some limitations such as the absence of proper means of using data for misuse prevention and potential overdependence on technology, which show the possible paths for future studies [20]. The findings shown in the analysis portray a rather diverse opinion picture. Several users appreciate ChatGPT for its ability to give people equal opportunities for enhancing knowledge and making the learning process individual and unique. However, there is a controversy over whether its use decreases students' motivation and increases such negative advantages of cheating such as the depersonalization of education. It also points out that; there is a growing voice demanding ethical frameworks and educational policies on the use of such AI chatbots [21]. The feedback analysis of the agent revealed a very high level of accuracy of the tutoring agent in responding to the students' queries thereby decreasing the elapsed time of the resolution of the query and making more tutors available to students. Students' feedback showed a certain level of satisfaction with the quality of interactions and usefulness of shared information. However, the findings also point to some of its weaknesses, especially in the manual operation, especially when dealing with more elaborative questions that require contextual analysis [23]. In the (Fig. 3.7), there is a webpage, which contains a description of an Interactive Tutoring AI Agent as one of its elements for students and their questions in a Research Seminar, namely Haupt seminar. The description of the page has a unifying heading that introduces the AI tool indicated and a brief description that reflects its main purpose, namely, its use for answering students' frequently asked questions about the to the right, there is a chat interface showing a student conversing with the AI, the latter including questions concerning the student's enrollment in the seminar and his or her condition. The student is active in the friendly dialogue and gives a response within it [22].

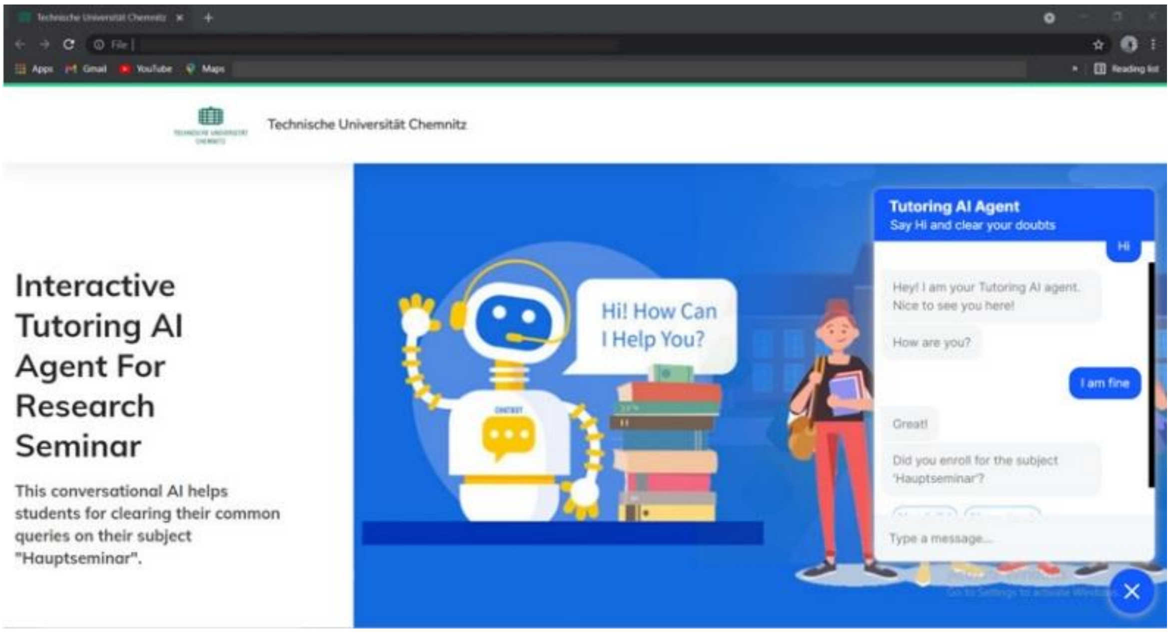


Figure 3.7: GUI of an Interactive Artificial Intelligence Agent [22].

The purpose of this study is to substantially expand the domain of AI-enhanced education by incorporating the modern AI technologies, namely, prompt engineering, and deep knowledge tracing in AI tutors. This includes not only improving the direct reactions of AI tutors to inputs provided by the student but also providing the latter with the ability to predict the learner's needs and adjust the content of education. Another goal is to prove the necessity of using these kinds of improvements in the conditions of real learning environment to promote the development of IT systems able to work in complex and implement various forms of intelligent interactions [19]. Thus, the purpose of this study is to identify and assess the innovative effects of generative AI on educational practices and performances. The present study also seeks to address gaps in the literature by systematically evaluating the implementation of AI tools in learning environments with an aim on presenting useful practical options and recommendations based on empirical evidence in order to assist educators and policy makers in the implementation of AI technologies in order to boost the educational effectivity and inclusion [20]. Therefore, the purpose of this research is to employ systematic steps to collect and analyze the users' insights about ChatGPT to identify the perceived advantages and limitations of using it in education. This will help the key stakeholders like educators, policymakers and AI developers know the possible ways of effectively incorporating AI Chatbots into education while observing the right ethic [21]. To create an application of an artificial intelligence tutoring agent that will enable direct effective academic facilitation to learners in higher education hence improving learning achievements and institution performance [23] [22].

It is seen that although there has been a great deal of improvement in the field of artificial intelligence in educational technology there is still a significant lack of advanced artificial intelligence technologies that can be easily incorporated into the common education environment and implement a truly personalized model. Again, there are some strains of AI tutors that have the capacity to adapt its teaching technique in response to the performance by students, an aspect not many tutors possess thus making AI learning blunt and less effective than it should be [19]. Despite the swift development of the AI technologies, there is still a lack of empirical research dedicated to the analysis of the generative AI in educational context, not mentioning the impact of the generative AI on long-term educational achievements and with the regard to ethical issues [20]. Currently, there seems to be limited massive-scale research that collects and examines the raw perception people have regarding the education application of AI technologies with special attention to social media platforms like Reddit [21]. Still, there is a lack of published studies that depict the utilization of intelligent tutor agents that can be implemented into the real academic environment consistent with different learners' requirements and supported by current academic frameworks [22].

As such, this research addresses these deficiencies with the proactive use of state-of-art AI techniques; namely: prompt engineering and deep knowledge tracing within AI tutors. The purpose of the study is to show how these technologies can be applied to deliver the content adaptation for learning styles and requirements for every student, all the time, to improve the learning provision [19]. This study addresses the critical research gap need by offering data on the advantages and disadvantages of generative AI in learning. It also presents guidelines to avoid the misuse of AI applications and assure that such solutions are introduced to the educational systems only for their improvement [20]. This research addresses this gap by utilizing Reddit data to discuss the user's early adopter perception and a real-time sociopolitical discussion on the incorporation of the artificial intelligence chatbot with educational systems [21]. This research fills this gap by not only building a technically effective AI tutoring agent but also by implementing it into a real education environment, which is important for indicating how AI may improve students' learning at a large level [22].

The emphasis of this paper on the advanced approach to making the AI tutors work even more effective makes the topic highly related to the thesis, which is devoted to the impact of the new forms of artificial intelligence in education settings. This study is hence line with the thesis proposal by applying practical advantages and possibility of the prompt engineering and deep knowledge tracing for enhancing the dynamics of learning through the application of AI [19]. This information is used in this research to support the ideas mentioned in the thesis statement where the primary topic is on how AI can help improve or reinvent education. Thus, the present paper contributes to the support of the specified thesis as the result of the detailed focus on the multiple opportunities that generative AI can provide for the modernization of educational practices [20]. Thus, this paper contributes to the thesis by offering qualitative data about the public attitudes toward the use of AI in education and augmenting the discussion of how AI alters educational processes and what key approaches are needed for the effective and equitable implementation of innovative technologies [21]. The positive outcome of the interactive tutoring agent is a good example within the thesis and supports the practicality and difficulties concerning the application of AI in education. It supports the thesis's argument about the impact of AI on personalizing learning processes to augment the effectiveness of delivered knowledge [22].

3.3 Generative AI for Emotional Analysis

This section provides an extended analysis of critical works dedicated to Generative AI in Emotional Analysis addressing the context of Artificial Intelligence Systems. To do so, it presents the state-of-the-art aspects of such technologies and their future possibilities in the assessment and management of human emotions appropriately. Some of these studies highlighted how Generative AI is being incorporated into systems to improve appreciation of emotion, which is vital in developing these better systems. Thus, as revealed through this section, generative AI perspectives on the transformation of the emotional analysis shall fit the needs of contemporary AI environment.

The motivation for developing a real-time multimodal intelligent tutoring system is necessary proposal to have a tool that will engage the learners more and will be able to respond to their needs. It depicts conventional learning spaces as not being sensitive to or accommodating the inequality of students adding that they negatively affect the learning process and achievement. Leveraging voice, text, and gesture inputs into this work, the goal is to provide more natural and perceptive means to enhance educational applications' effectiveness in assessing students' feelings and response [24]. This research is initiated based on the analysis of the existing online tutoring systems that give the conventional way of learning that rarely covers the characteristics of the learner. In this regard, the paper presents a new system called 'Seis Tutor', which is a learning model that this paper has engineered to go beyond just conducting online classes but with highly flexible and personalized tools in learning. This system seeks to bring out the education dynamism by matching the learner by his/her preferred style hence boosting learner interaction and successful learning. The rationale is to fill the apparently lacking aspect of personalization and adaptivity in traditional settings of online tutoring focusing only on the availability of education instead of its individual effectiveness [25].

This methodology implies the usage of four AI technologies: speech recognition, natural language processing, and emotion recognition to design the tutoring system capable of operating with the students via different channels. Due to the relatively simple design of the system, and the fact that it works based on real-time feedback received from the student, the system responds to student inputs correspondingly quickly. Thus, this approach makes the tutoring particularly effective and relevant to the needs of each of the students [24]. The (Fig. 3.8) provides a general framework of education aiming at improving learning and teaching practices within the context of integrated models. This framework's core is the Interface Model, which works as the primary point of communication between the student and the teacher. This model is linked to four other specialized models: the Student Model that employs numerous techniques to address student differences in learning styles; the Affective Model that deals with such things as facial expressions or even tone of voice or text to tailor educational strategies; the Domain Model responsible for offering the relevant and up-to-date educational contents; and the Pedagogical Model that provides feedback depending on a student's progress in learning. In combination, these models foster a functioning organizational system of education that pays attention to students' emotional and mental requirements for learning, as well as their responsiveness to teaching techniques.

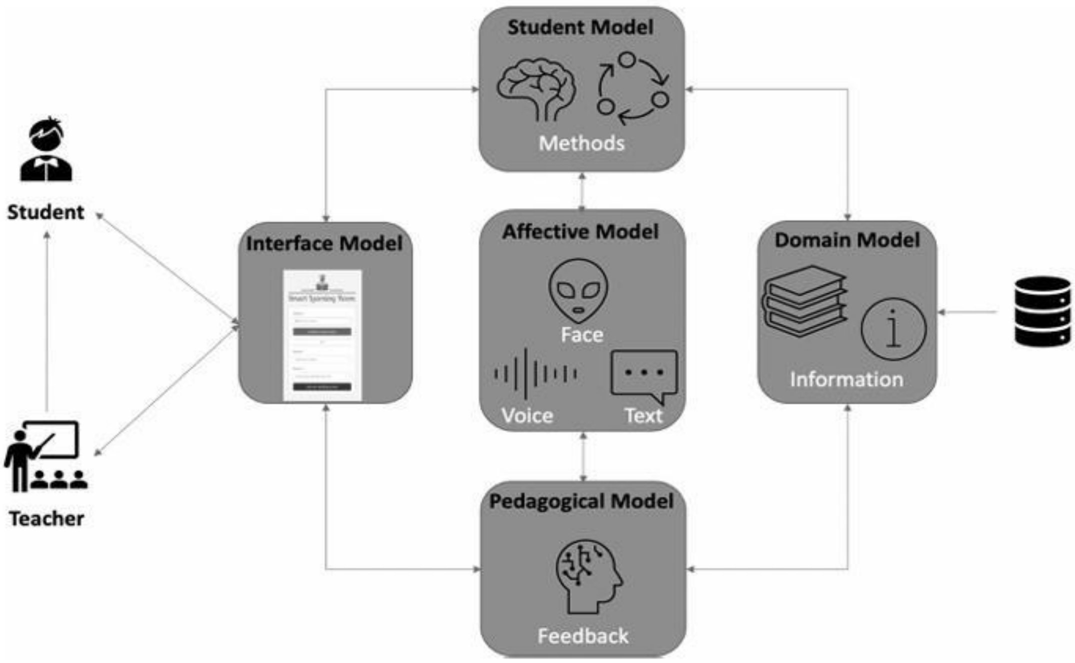


Figure 3.8: Integrated Learning System Architecture [24].

The study employs an empirical approach of a systematic evaluation of ‘Seis Tutor’ with various other competitors in the online tutoring business. In this comparative analysis, the areas of special interest are to what extent existing sites personalize for the learners, how learner engaged the sites are, how adaptability is addressed, and the overall learning outcome achieved. The research uses both the quantitative approach by measuring and evaluating the data of users’ interactions with the systems and the use of the qualitative approach to elicit users’ feedback concerning the effectiveness of each system. In this way, through such two-faced evaluation, this study expect to gain essential and numerous understandings of how the each of these systems have been catering the different learner population adequately and also to provide a record of the effectiveness of the proposed one in providing more proactive leaning environment [25].

The outcomes suggest that the real-time multimodal intelligent tutoring system effectively simplifies learning and increases the students’ interest. Responses from students indicated that they establish a closer link with the learning resources and felt supported from the tutoring approach as it is flexible in accordance with students’ learning styles and speed. Also, the system examined that such factors as understanding, and identification of student emotions were correctly and to the point responded by improving the learning process of the subject [24]. See Table 3.3 that generates a classification report on the performance of a sentiment analysis model for an emotion-wise distribution of the text such as Neutral, Joy, Sadness, and Anger. In each of the categories of emotions under study, evaluation metrics are precision, recall, the f1-score, as well as the number of sentences considered (support). Precision scores that show the extent of the accuracy of the prediction vary and range from 0. 94 to 0. 98, while recall values, which measure the model’s capacity for the correct identification of all instances of an emotion, stands at 0. 97. As a fair check, between precision and recall, the f1-scores are around 0. 96 or 0. 97.

Table 3.3: Emotion Classification Performance Report [24].

Metrics	**Precision**	**Recall**	**f1-score**	**Support**
Neutral	0.98	0.97	0.97	1873
Joy	0.97	0.96	0.97	729
Sadness	0.94	0.97	0.95	316
Anger	0.96	0.97	0.96	488
Accuracy	-	-	0.97	3406
Macro avg	0.96	0.97	0.96	3406
Weighted avg	0.97	0.97	0.97	3406

Figure 3.9 [24] depicts a snapshot of an online conferencing possibly in the context of academic or teaching and learning activities with emphasis on live video conferencing and sentiments analysis. In this case, the primary sample screen shows a video feed of “Ines” and beneath is the emotion recognition result that labels her as ‘angry,’ among other tags. To the right-hand side of the figure, there are options of chat. The platform combines video communication with emotional analysis for improving the interactive mode of communication with customers and responding correspondingly. In macros-level the interface is clean and well-organized, it shows some functional abilities like message sending, and others like tracking emotion.

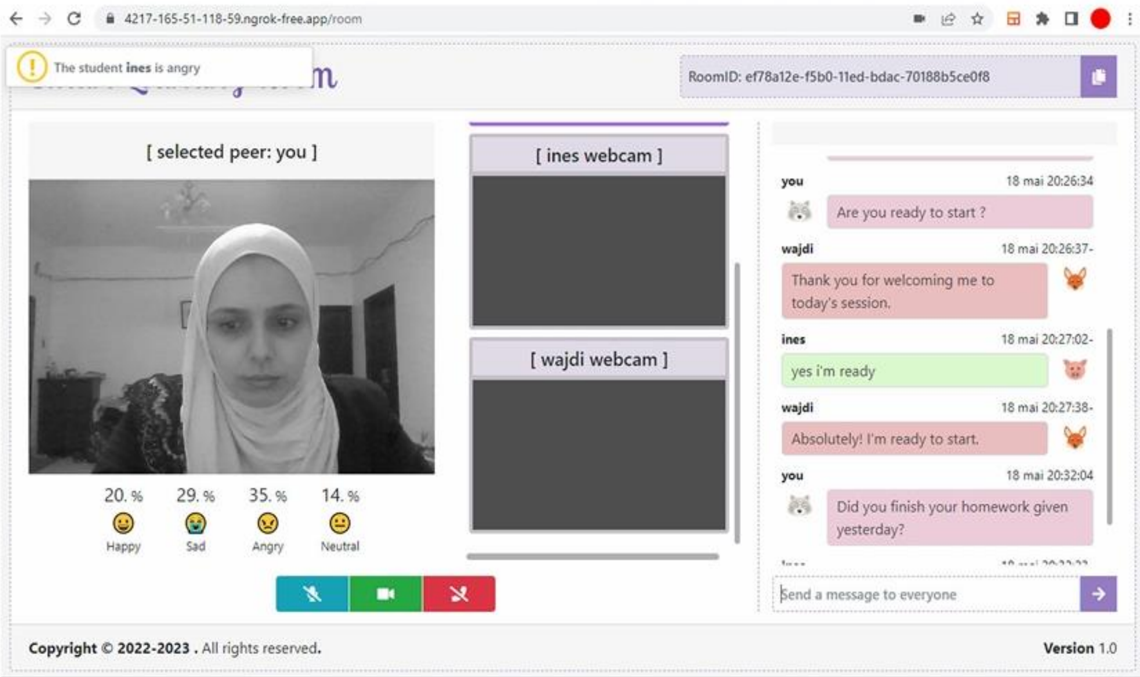


Figure 3.9: Smart Learning Room with Enhanced Educational Communication Platform with Integrated Emotional Analysis [24].

The results of the study are rather encouraging, proving that ‘Seis Tutor’ is superior to many currently available online tutoring systems in such significant respects as personalization and adaptivity. In general, the key findings for students using ‘Seis Tutor’ demonstrate the efficiency of the system in terms of its ability to address the need for differentiation of learning needs and modalities of learning under the umbrella of technology enhanced learning. From these outcomes it can be said that such adaptive features of ‘Seis Tutor’ do not only provide improvement of the learning process, but also creates more interesting and effective educational environment, thus confirming the possibility of the further development of the advanced adaptive systems for the online education [25]. The (Fig. 3.10) represents a digital screen of an e-learning platform that combines the video analysis and lessons. To the left, a video feed shows a frame with the student inserted for Emotion Recognition and labeled ‘Neutral’ with a blue emoji to Applying emotion recognition technology to determine engagement level of the student. The right side has an e-learning area that includes the title “Introduction” of the course, course progress bar, navigation tools, and “Mark as Complete” button. This arrangement creates a friendly learning atmosphere, in which the dissemination of knowledge is influenced by a student’s feelings, thus increasing its effectiveness.

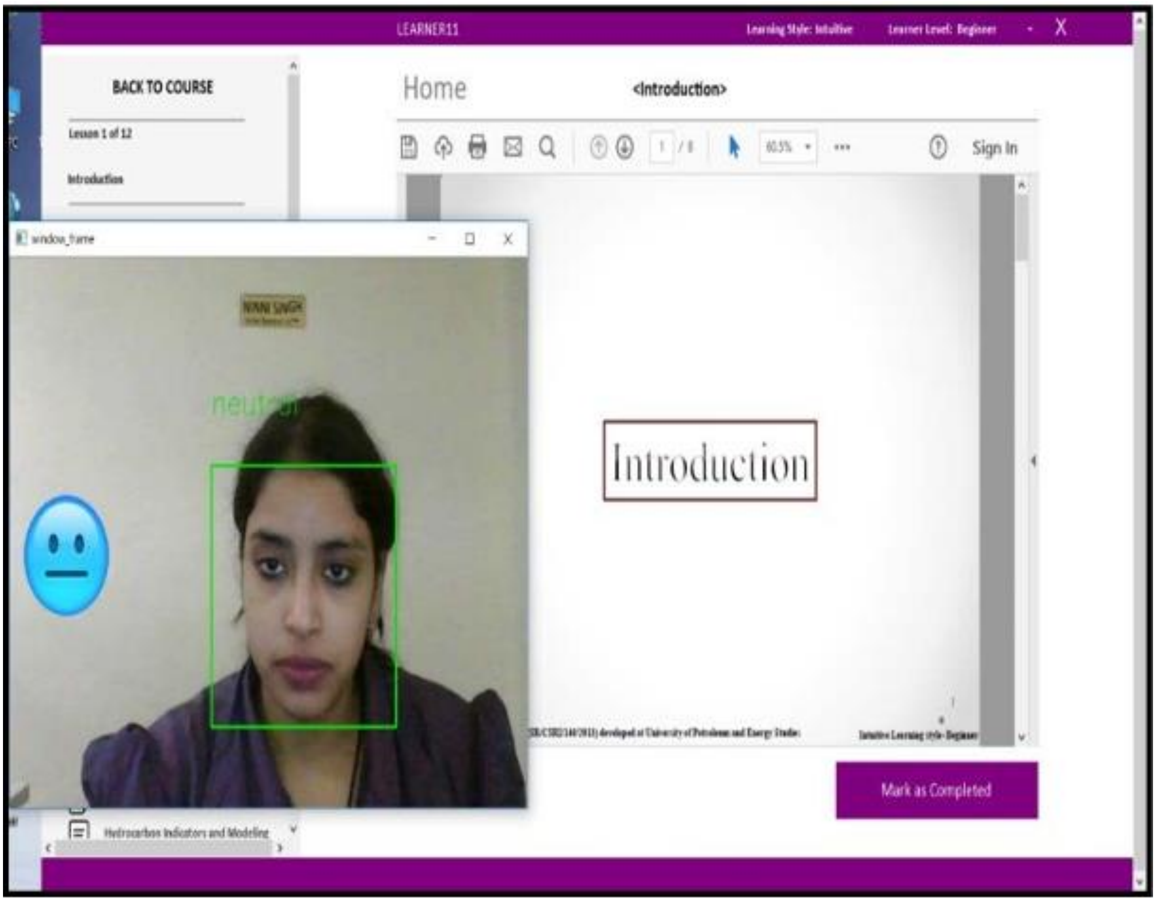


Figure 3.10: Educational Communication Platform with Emotional Analysis [25].

The primary goal of the present research was to design and evaluate a multimodal ITS that can apply real-time data to facilitate education. Using an APO model, the system is designed to meet the learner’s immediate instructional requirements so that teaching and learning becomes more effective [24]. Therefore, the aim of this research is to establish the efficiency of the ‘’Seis Tutor’ as the effective online tutoring system than the existing online tutoring systems. As such, this paper aims to advance the claims of ‘Seis Tutor’ and Other related educational technologies that are embedded in the learning management system, by proving that the altered algorithm improves adaptivity and the level of personalization. In addition, the presented research also intends to present a clear and replicable evidence on similar technologies’ implementations in more general educational contexts so that the current education systems become more individualized, and learner centered [25].

Many existing technologies in use in educational setup do not have the capacity to adapt and merge different kinds of student feedback, especially in real-time. This is more so if dealing with students' emotional and behavioral signs because these are critical in facilitating learning. It is worth to notice that most developed systems are focused on the static model of the communication and do not take into the account the situation when a student has a change his/her mood or behavior during the learning sessions. This gap called for a need of more complex AI-based systems that can interpret and respond to many inputs, and one that can tailor learning in real time in a real way [24]. Nevertheless, the increase of web-based tutoring systems still exhibits a significant lack of genuinely effective individualization. It is worth emphasizing that most existing systems do not contain a proper and clear-sighted scheme that can adapt to each learner's requirements and successfully navigate through the IDs without an individual approach. This gap in the literature underlines the importance of an extensive comparative study that could presumably data-show the advantages of quite progressive adaptive and personalized education systems as opposed to more traditional forms of e-learning environment [25].

Thus, this work fills the gaps by demonstrating the example of a multimodal and real-time architecture that employs modern AI solutions to design an effective and 'smart' tutoring environment. Data from emotion recognition and behavioral analysis are combined with natural language processing and featuring machine learning to adapt to individual inputs from the students. This makes it possible for highly individualized learning environment to be achieved, and this shows that it is possible to integrate these capabilities in the improvement of the flexibility of the learning technologies [24]. This paper rises to this challenge through a highly structured parametric comparative study which exposes the enhanced effectiveness of the 'Seis Tutor' while also laying out a road map where the effectiveness of personalization and adaptivity interfaces in education can be measured and mapped. Thus, the study provides practical evidence that basically substantiates the implementation of such systems, proving their responsiveness to actual educational environments [25].

The research correlates well with the thesis of the thesis by focusing on the application of advanced AI technologies in live educational environments, shedding light on novel ways of enhancing learning through technology. It is more specific in demonstrating how real time AI systems can enhance teaching and learning by presenting content of the first type thus serving the purposes of the thesis which aims to establish the role of technology in enhancing teaching and learning. That is why this research is not only applicable but also becomes one of the essential parts of the thesis, proving AI's significance and possibilities in the sphere of learning [24]. First, it is necessary to state that this research is closely related to the topic of the present thesis, as the major directions discussed in the thesis are aimed at improving the usage of educational technologies. Thus, the study helps to enrich the ongoing discussion on the efficacy of adaptive and personalized learning approaches by looking into how 'Seis Tutor' was applied, and the results obtained. This relevance is enhanced by the explicit utilization of such findings within the context of the thesis, which seeks to enhance educational practice by applying information technology selectively; hence putting the research at the heart of the thesis' rationale and suggested outcomes [25].

In summary, the integration of generative AI into the field of education holds transformative potential for fundamentally improving both educational experiences and outcomes. This integration promises to revolutionize the way education is delivered, making it more personalized, inclusive, and effective for students across diverse learning environments. Generative AI refers to advanced algorithms capable of producing content, such as responses and solutions, based on the data they have been trained on. In educational settings, this capability can be harnessed to create dynamic learning materials that adapt to the needs and learning pace of each student. For instance, generative AI can analyze a student's previous responses to tailor subsequent educational challenges, ensuring that each student is engaged at the optimal level of difficulty. This personalized approach helps maintain student interest and promotes more effective learning. Furthermore, the inclusivity offered by generative AI in education is another significant benefit. Traditional educational models often cater to the 'average' student, potentially neglecting those who may require more support or challenge. Generative AI systems, however, can adapt to cater to a broad spectrum of abilities and learning styles, making education more accessible to students with disabilities or those who might otherwise feel marginalized within the conventional education system. To fully realize these benefits, several existing challenges must be addressed. One major challenge is the ethical use of data: ensuring that student data used to train AI systems is handled securely and responsibly to protect student privacy. Another challenge is the potential for AI to perpetuate existing biases found in training data, which could lead to unfair or ineffective learning experiences. Tackling these issues requires robust ethical guidelines and transparent AI systems that are regularly audited for fairness and accuracy. By overcoming these hurdles, future advancements in AI technology could lead to its widespread implementation in educational systems worldwide. This widespread adoption would mark a significant step forward in education technology, offering systems that not only adapt to individual learning needs in real-time but also provide consistent, supportive, and engaging educational experiences.

In this evolving scenario, educators would be equipped with tools that augment their teaching, allowing them to focus more on pedagogy and less on administrative tasks. AI-driven systems could handle routine educational interactions, freeing teachers to engage more deeply with students on complex topics and personal development.

4 Methodology

The methodology involves ensuring smooth integration with the front-end to create a seamless user experience by aligning backend functionalities with the front-end interface. This thesis maps out a plan to incorporate such enhancements on conversational AI with the front-end application. There is the establishment of APIs, middleware, and efficient incorporation of emotion recognition technology for improving the tutoring system. The methodology is tightly focused on integration with the front-end which helps to have a web application interface with the user through the usage of the modern web development frameworks and team cooperation of the front-end and backend. APIs and middleware for System Interface ensures that all interactions between the different systems occur through Restful APIs and middleware for processing the Users authentication credentials, data processing, and error handling. Robust data management is required on how to keep up data accurateness, privacy, and efficiency when using the relational or even non-relational database, and standard techniques like normalization, indexing, and caching. Emotion recognition technology is to improve the outcome of user experiences, which targets the emotional part of users, applying the machine learning models with data containing various emotions. Scalability and performance are maintained using scalable architecture designs, load balancing, regular performance monitoring, and optimization.

They consist of Contextual Understanding which relates to the ability of the system to process Natural Language and respond contextually, and Personalization, in which the ability refers to customized interaction as per the preference profile of the user which would also require Machine learning. Functional requisites concern inclusion of natural language processing for natural language understanding of the input data through the application of modern NLP techniques and libraries, application programming interface for invoking external resources and services with the help of secure and optimized API. That way, there is a guarantee that an intricate, user-adapted, and technically sound first conversational AI is created. This methodology ensures the development of a sophisticated, user-centric, and technically robust conversational AI system.

4.1 Research Design

The research design of this study focuses on the development and evaluation of a generative AI tutoring agent, as illustrated in the (Fig.4.1). The architecture of the tutoring agent is divided into two main components: the front end and the back end, each serving distinct functions to facilitate seamless interaction between users and the machine learning model. The research methodology encompasses several stages. It consists of several steps: introduction and problem formulation, selection of a model type, data acquisition, data preparation, model learning and tuning, integration of the model and its deployment, and finally, model assessment. Each is instrumental in establishing a solid, scalable, effective system that caters to individuals' learning styles. Firstly, I shall review prior work about ITS and generative models to determine current enabling technologies. This exploration will help choose the right machine learning model, for instance, Llama2, that will enhance the core of the backend system. Collecting and cleansing data are important preliminary steps that consist of acquiring materials related to education and preparing for the model's training. This is done by pre-processing the data where activities like data cleaning, handling of missing data, and lastly, transformation are performed. The following step is model training the model the model further or fine-tuning it. As mentioned, with the prepared data, the next focus is training the generative model to capture and produce suitable responses to a given user's query. It further enhances the given model to be used solely for an educational environment and offers proper educational-related answers. At the same time, integration and deployment are the processes of developing the API and middleware using Python and Flask to enable frontend-backend cooperation. The backend system shall interface with a web application written with HTML, SCSS, Bootstrap, Typescript and Angular. The API will also interact with the local file system and a database containing files needed for the application's functioning and user data. Last, the evaluation of the system will be done to compare the results thus achieving educational outcomes to determine the efficiency of the system and the possibility of making additions and improvements as a result of the tests and evaluations.

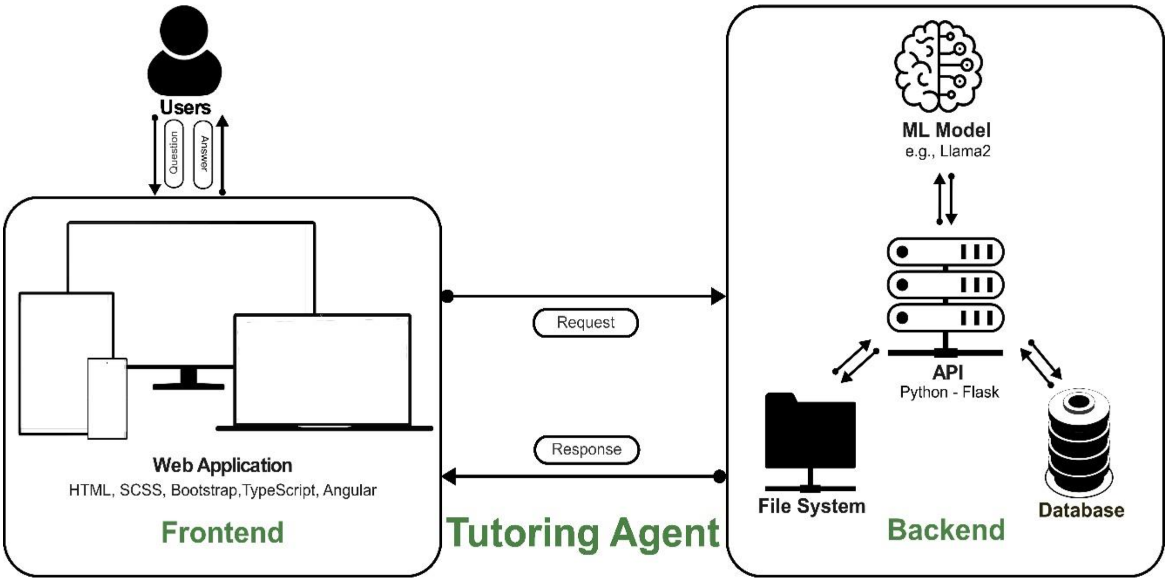


Figure 4.1: Concept of a Backend of Generative Model AI Tutoring Agent.

4.2 Initial Exploration

4.2.1 Rivescript

Initially tried using Rivescript, which is a particular scripting language focused on chatbots creation. Thus, even though Rivescript was easy to implement and allowed building simple conversational agents, it was not sufficient for our tutoring agent as it lacked the necessary capabilities. Namely, it was unable to produce innovative and situationally relevant replies, which might be crucial when it comes to educational application. This limitation delegated us to look for more complex models in machine learning algorithms. Because Rivescript is based on rules alone it failed to address the richness and nuances of natural language that will be needed for an educational environment. Even though the experiment was somewhat primitive, and the conversational agent created with the help of Rivescript is quite simple, one can learn about the basic needs and difficulties of creating such an assistant [26].

4.2.2 Extractive Models

Next, we looked at extractive models including, for instance, RoBERTa that was pre-trained to work with the SQuAD datasets. This model did well in answering questions that directly translated from the questions asked when an exact match was obtained. However, it had difficulties in producing imaginative answers or in dealing with questions that asked for an inference not indicated in the text. It was due to this limitation that we had to consider applying generative models. Hence, extractive built models like RoBERTa fall short when it comes to such settings with strict and often recurrent reliance upon the presence of exact information in the dataset , and where inference and even creativity often play a much bigger role [27].

4.3 Transition to Generative Models

4.3.1 Custom Machine Learning Models

Our initial attempts with the custom machine learning models developed with assistance from PyTorch and TensorFlow at the start of development did not turn out as efficient as anticipated. These models called for a lot of resources for training and were known to give least relevant and sometimes wrong answers. This was actually a good experience to show how the need for even better pre-trained models is a necessity. Specific models required considerable time and resources aimed at parameter tuning and enhancement, which was not efficient taking into consideration the constantly evolving context of education content and communicational processes.

4.3.2 OpenAI's GPT-3 and GPT-4

We then looked at OpenAI’s GPT-3 and GPT-4 [28] that showcased remarkable ability in producing text that resembled or even duped human-written text. However, the subscription-based model and the company’s complete reliance on the outside infrastructure were the major disadvantages. This meant that we needed something that could be hosted in DSM and could be managed by us without external help, prompting the search for other products. Thus, although GPT-3 and GPT-4 demonstrated an impressive level of performance in language generation, their availability with a rather high cost and strict dependence on the infrastructure of the OpenAI company raised certain issues in terms of their applicability to the conditions of our educational environment and their independent and mass use.

4.4 Model Selection

4.4.1 LLaMA Models

We compared the LLaMA 2 and LLaMA 3 models and showed that even though they worked well, their computation was heavy. The above-mentioned models proved useful but were utterly unattainable in our limited resource context. What we wanted was the performance combined with the rationality of using as few resources as possible. Studies have revealed that different LLaMA models work for terms, but their computational complexity is high, which revived the search for lean models for real-world applications [29].

4.4.2 FLAN-T5 Model

Finally, it brought us to the decision where we determined that FLAN model with specifically MBZUAI and LaMini-Flan-T5-248M is the most suitable for our case. It is a specific variation of Google's FLAN T5 that has been optimized for the task of instruction following. This characteristic ensures that it displays a good combination of performance and use of resources, which is ideal for our tutoring agent. These are due to instructional dataset with which FLAN-T5 has been fine-tuned, architectural modification for following instructions that boost its performance in educational tasks with high accuracy without stressing out the resources.

FLAN-T5 was chosen as a base for creating an intelligent tutoring system because it corresponds to the nature of educational tasks and their training. The present paper describes in detail how FLAN-T5 works and why it was chosen and integrated into our system. FLAN-T5 is built on the T5 (Text-to-Text Transfer Transformer) architecture in which every NLP task is reformulated as text-to-text transduction. This concept makes training and application easier across various tasks including translation, summarisation and question answering. This model is a more advanced version of this model known as the FLAN-T5 as it contains a new fine-tuning instruction-following exercise that makes it very appropriate for academic use.

The fine-tuning of the MBZUAI/LaMini-Flan-T5-248M model was conducted with reference to a variety of instructional datasets: The fine-tuning procedure improved the model's capability of receiving and understanding complicated directions in the procedure of interaction with clients that can be considered as one of the most crucial characteristics of an ITS agent. These datasets can comprise descriptions of several steps for solving different subjects, the instructions and solutions for the problems, the learning questions and the corresponding answers, the context for the unknown terms or expressions, and so on. Due to such a wide variety of Learning Content, the training of FLAN-T5 provided options for various educational situations with very high Accuracy and Contextual Relevance. It should be noted that the problem of the correlation between the model's performance and the resources used to achieve it is the primary strength for FLAN-T5. Unlike the big models such as GPT-3 which requires a lot of computational power, FLAN-T5 can perform very well, and at the same time is easy to handle in terms of resources. This makes it possible to implement the model in environments that are less powerful than the one used in this study such as the conventional server platforms [30].

4.5 Data Collection and Preprocessing

Data collection, conducted by a master's student, involved sourcing a diverse range of educational content [31]. The data included various question-answer pairs, explanations, and feedback relevant to the educational domain. The diversity of sources ensured comprehensive coverage of topics and the inclusion of multiple perspectives, crucial for creating a versatile and adaptable tutoring system. Custom datasets were particularly valuable in tailoring the model to specific educational contexts and user needs, ensuring relevance and applicability.

Data preprocessing is an essential step that requires cleaning of data in order to prepare the dataset that will be need for training the generative AI tutoring agent. Data preparation started with data cleaning since this process aimed at eradicating duplicity, irrelevancy, and mistakes in the dataset. It is here that consistency was automatically checked by scripts to some degree and repaired where necessary; the data quality was then manually checked. It is necessary to have clean data at the model training and performance level to avoid corrupt or irrelevant pattern learning by the model. The second step involved the normalization of the text where the format of the text to be analysed is adjusted for analysis. This comprised of capitalizing all characters to lowercase, eliminating the special characters as well as applying stemming. This step helped in making sure that the data was consistent in the set, and this was important in preprocessing of data. Normalization also included work with CONTRACTIONS and regularizing of ABBREVATIONS for consistency. This way, the solutions will be more polished, and the data formatting ensures that the model learns to recognize patterns much more effectively, thus enhancing the generalization factor. Data enhancement procedures were then used to expand the dataset and bring in more variety to it. Techniques employed included replacing words with synonyms, rearranging the sentence's order, and the random addition and removal of words within the sentence and its context were utilized adding practicality and generalization of the model to the data collected. Augmentation brought in the aspect of variation, which allowed the model to understand different interpretation of the same idea and make it more resilient. Also, the technique of paraphrasing together with contextual data augmentation was used to broaden the horizon of the given data set for the model [32]. Last but not the least, based on the pre-processed data, the data was divided into training data, validation data and the test data in the ratio of 80:10:10 respectively. They keep the evaluation process handy by making this split; it guaranteed a rigorous evaluation of the model. Therefore, the training set was employed for model training, the validation set was employed to fine-tune hyperparameters as well as to decrease it overfitting on the training data, while the test set was used for model generalization performance assessment. This kind of splitting of data set minimizes the risk of; bias as all types of set were represented proportionately as a measure to enhance the performance of the model.

4.6 Model Training and Fine-Tuning

The model training process involved fine-tuning the pre-trained MBZUAI/LaMini-Flan-T5-248M model on our custom dataset. The training process was managed using the Seq2SeqTrainer from the Hugging Face Transformers library.

4.6.1 Training Configuration

The following hyperparameters were used during training:

- Learning rate: 0.0005
- Train batch size: 128
- Eval batch size: 64
- Seed: 42
- Gradient accumulation steps: 4
- Total train batch size: 512
- Optimizer: Adam with betas = (0.9,0.999) and epsilon = 1e-08
- LR scheduler type: linear
- Number of epochs: 5

All these hyperparameters were selected after preliminary experiments to enhance the stability and performance of the training phase. Learning rate and batch size was adjusted to control the speed of training model and convergence, while the optimizer and it is setting where chosen to optimize the gradient descent.

4.6.2 Training Process

The knowledge generation for the intelligent tutoring agent's generative model was a careful process involving the following steps. First, the FLAN-T5 model, which can be downloaded from the Hugging Face's Transformers library was preloaded. Subsequent specialization was built upon this pre-trained model because this model was trained intensively on a vast number of datasets.

Further, since developing the model, the pre-processed dataset was then tokenized for the input data of the model. This process entailed fragmentation of the text into more manageable portions, like words or subworlds, in order to ease the model's operations and interaction with the information provided. In addition, aim after that, the dataset is preprocesses in the way that FLAN-T5 model requires, in order to have consistent environment while training and using it.

Once the data was ready, the model went on to the stage of fine-tuning. This important step I used to train the model on the prepared dataset keeping several appropriate hyperparameters. These hyperparameters such as learning rate, batch size, and optimizer setting that were used were determined through a trial-and-error approach to enhance training effectiveness and outcomes of the created model. It was done with the help of the Seq2SeqTrainer, an element of the Transformers library, which controls the backpropagation of errors and modifier to the model's weights. This phase enabled the training of the model from pre-schooling of its learning to the characteristics and finer details of the educational dataset to be more precise in the generation of responses for the tutoring agent in each educational context.

To achieve the model's generalizing capability and avoid overfitting, a validation was introduced at the end of each training iteration. In this stage, the complex was tested on a new, unseen data set to assess the model's performance. Evaluation of the model was done with accuracy, precision, recall and the F1 score for the purposes of deducing its efficiency in producing relevant responses. Apart from that, this regular testing and validation helped to give general information about the model's learning process and acted as a measure of monitoring for overfitting so that necessary changes to the training process could be made if needed. Such careful compliance with all the described steps and the constant tracking of the model's performance during training allowed fine-tuning the FLAN-T5 model to meet the requirements of the intelligent tutoring agent, thus increasing its capability to deliver efficient and individualized learning support to users.

4.7 Integration and Deployment

The system architecture comprises three main components: frontend, backend, and database. These components are connected through APIs and installed to servers, and there is arrangement for the best performance and security.

4.7.1 Frontend

In this thesis, the front end, developed with an emotional avatar, was ordered by the student [33]. The front end implemented using HTML, CSS, and JavaScript, allows the user to interact conveniently. Based on this, the design is relaxed and built to be user-friendly across as many devices as possible. Web standards of a responsive design were adopted to be compatible with different devices and improve usability. The front end also featured emotional avatars, sliders, and forms that enabled user-friendly interactions on the website, depicting two emotional avatars. One avatar is portrayed as a female character, while the other represents a male character, as shown in (Fig. 4.2).

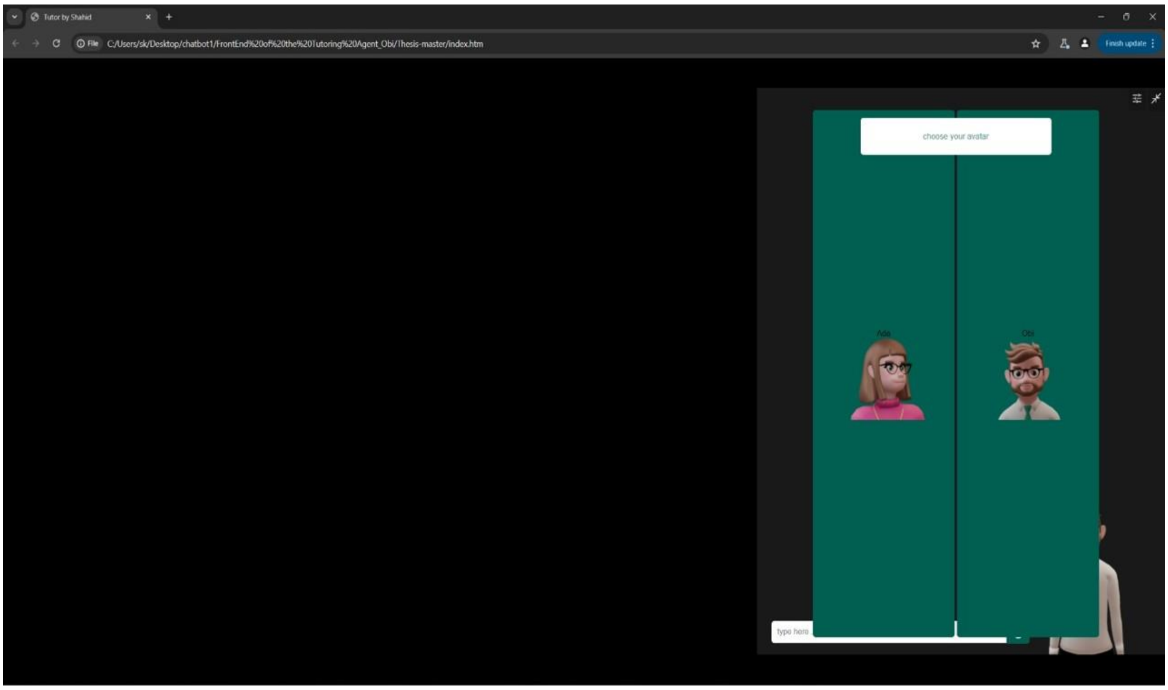


Figure 4.2: Front End of a Generative AI Tutoring Agent with Emotional Avatar [33].

4.7.2 Backend

The backend which was developed in Flask receives API calls and interfaces with the model. The backend activities include the computation of user's query, the formulation of the results of the query and session control of the users. Features in Flask's architecture also include lightweight and flexibility and therefore made it easier to develop and deploy the API. The backend also performed some data preprocessing within the query, to align the query with the model before it was processed.

4.7.3 Database

The database stores user data, session information, and interaction logs. It is held within the database and includes the users' details, session details, and interactions. The use of this data also becomes crucial for addressing people's queries while serving as a strong basis for the enhancement of the system. We kept using Relational database Management System (RDBMS) to maintain data accuracy and query the data. Different types of data about the users and the logs of the interactions are provided in the database schema for better manipulation and storage.

4.7.4 RESTful API

A RESTful API facilitates communication between the front end and back end. Key endpoints include:

- ***/chat:*** Handles user queries and returns model-generated responses.

- **/train:** Initiates model retraining with new data.
- **/auth:** Manages user authentication and session handling.

RESTful API, the best practices were implemented to achieve scalability, security, and maintainability of the system. Different measures were taken to put into practice the API security and this included the use of services such as authentication and authorization in the API usage and access. This was accompanied with various documents that helped in the use of API.

In summary, the methodology outlined in this section provides a comprehensive framework for developing a generative model-based backend for an intelligent tutoring agent. By employing state-of-the-art models, rigorous data preprocessing, and robust evaluation techniques, the system aims to deliver personalized and adaptive learning experiences. The approach taken at the generative AI tutoring agent's backend level concerns the methods of proposing and assessing the system's efficiency that should be used to manage user queries and generate the corresponding answers. The backend is built using Python with the Flask framework, which ensures good API communication between the front end and the machine learning model. The centrepiece of the backend is the ML engine and large language model LaMini-Flan-T5-248M, designed to provide truthful and semantically sound answers to the posed question. Also, the backend includes a file system for storing and managing important data and model files to make them easily accessible and well-processed. Behind this application is a database that stores user interactions, model responses, and other appropriate data to make data persistence and performance analysis possible. Due to the backend architecture, interaction is facilitated following a smooth interaction flow in processing user requests, implementing the ML model to generate the required responses and returning these responses to the front end. This methodology makes it possible to ensure that the backend caters for the tutoring agent’s objective of offering efficient and appropriate tutoring agent. This methodology ensures a systematic and thorough approach to developing an intelligent tutoring system that meets educational standards and user expectations. The iterative process of testing and refinement, coupled with advanced machine learning techniques, positions the system as a robust solution in the field of intelligent education. Future enhancements will continue to leverage user feedback and technological innovations to maintain the system's relevance and effectiveness.

5 Implementation

The design and development of the backend for a Tutoring Bot can be understood as a combination of complex AI algorithms and user interface possibilities that allow the design of a sophisticated and fully functional tool for effective education. Closely connected with this is a conversation flow generated from an open-source, powerful generative model that allows the generation of proper and interesting responses. Complementing the system's capacity is a large language model and its significantly improved ability to understand and process multifaceted queries regarding different topics within the sphere of academics. This not only helps the bot give the best tutoring responses to each student but also makes these tutoring responses based on the learners' learning aptitude and speed. In the same way, the backend features emotional intelligence incorporated into the design of the system for the identification and management of emotional responses. This enables the intelligent bot to steer its interactions with a very elastic preparedness, with constructive and comforting messages that create a good training atmosphere. Also, the backend efficiently deals with real-time interaction with learning analytics data from the Graphical User Interface (GUI). This feature helps the tutoring bot get real-time feedback to change the teaching strategy in what a particular user wants to learn and be taught at that moment. This is an inflow step that involves a user entering his or her credentials into the GUI. The entered details are then validated against a database to grant permission. After the authentication process succeeds, the system fetches user queries; they may be keyboard input or a set of queries already set by the system, which undergo further elaboration in the backend. At this stage, the bot requests and receives the user's information from the database, creating a context for further communication. This data is used to form what essentially is a personalized search statement for the Question Answering System, which involves the input of data pertaining to the user in question to increase query match relevance. Thus, with the help of the LLM, the tutoring bot formulates a correct answer to the given query based on the formulated query. At the same time, the second ML model identifies the user's emotional state by processing sentiments derived from the user's query and the produced bot's response. This two-tier processing allows the bot to respond with the relevant facts and the response that would fit the user's emotional state. Responses together with their emotional evaluations are then sent to the frontend and shown within the chat application. This not only helps to answer the user's query but also helps to reflect the emotional tone that improves the interaction quality. It also changes the emotional state of the bot in the chatbot depending on the sentiment analysis to match the user's emotional context when responding. This extensive design guarantees every aspect of the Tutoring Bot's backend, including user identification, real-time intelligent learning and affective computing, is optimized to provide a personal, effective and emotionally sensitive learning process.

This thesis implementation has incorporated some tools and libraries to enhance our natural language processing system. To tap into the efficient and readily available pre-trained models in the transformer’s library, we used AutoModelForSeq2SeqLM and Auto Tokenizer to load, fine-tune, and save the models and tokenizer. The Datasets library helped load, sort, and even format the data to a form suitable for training and validation. Another component of our system was the pipeline function of the Transformers library, which allowed generalizing the application of our models to tasks ranging from emotion perception to text production. For debugging and to monitor further protocol, we decided to use Python's native logging, which comes in handy in abating events and searching for problem areas. We implemented a web-based application for the graphical interface due to Flask to incorporate endpoints for our trained model. To allow the web application to work, we use Flask CORS to enable it to serve requests from origins. In this process, PyTorch was used to handle the model training and deployment of the work, with the reference of the transformers' Seq2SeqTrainer. Several PyTorch modules: GPU support helped speed up training and inference time. As it would be observed, the Seq2SeqTrainer was crucial for the finalizing of our sequence-to-sequence model on the data that we created. Finally, we employed JSON for data handling to load our question-answer pairs from a JSON file. It helped regulate and control the data flow, setting up and improving the quality of the given data that was the input for our model.

5.1 Development Environment

The creation of an effective and strong development environment was one of the most important conditions for the successful completion of our work. This environment was systematically set up for the various stages of natural language processing model creation, training and implementation. The setup mainly concentrated on the use of efficient computational tools, programming languages, and frameworks that are well suited for the large-scale data processing and machine learning jobs. This section focuses on the precise definition of the development environment, the languages and tools used for constructing and implementing the NLP models, as well as the required hardware and software environment for the project's effective implementation.

5.1.1 Tools and Languages

Many of the tools and programming languages we've chosen for the software foundations of our thesis are optimized explicitly for NLP and support thousands of other resource-heavy NLP tasks. At the core of the proposed toolkit, we borrowed Transformers from Hugging Face, famous for having one of the most extensive sets of pre-trained models and tokenizers. From the discussed library, Obsidian stands out as one of the most potent solutions widely used in the NLP community across various tasks, including text classification, summarization, translation, etc. It was chosen for its high speed and simplicity, as well as the versatility of use, which excludes the presence of many factors that complicate the NLP model deployment.

Python, with inherent simplified syntax, was the language of choice for implementing the thesis. It enables fast installation and comes with large libraries and frames, which are crucial for AI and machine learning. Thanks to the compatibility of Python, which allowed the use of these pre-trained models with no issues and little interference from the Transformers library in our system. Python language is friendly, widely used, and actively developed. This means that today's generative AI advancements will always be included in the updates.

Furthermore, another helpful tool by Hugging Face is the Datasets library aimed at the effective loading, sorting, and preprocessing of large datasets needful for feeding models. This library played a crucial role in data management; hence, we used this library to prepare our datasets for both the training and validation of our models. Its integration with the Transformers library helped ensure that the interoperation allows for a high degree of efficiency in data stream processing model training and other associated processes.

The pipeline feature available in the Transformers library was used to properly utilize our models for specific NLP tasks to utilize our models for specific NLP tasks properly. This powerful feature can help apply models to tasks with few boilerplate functions, including preprocessing, model adding, and post-processing. Regardless of the specific task of emotion detection or text generation, the described pipeline could deliver accurate solutions with a limited amount of code.

Python's native logging module was used extensively for solid application development and maintenance throughout this thesis. It offered a dependable means of recording crucial features in the system, observing the application's behavior, and remedying problems as they occurred. This functionality was critical throughout the system's development and deployment to ensure its high reliability.

Regarding the program's overall structure to be designed, Flask stood out as the most appropriate web framework within the context of building a GUI that would be effective for the user and easy to navigate. Because of its lightness and elasticity, it was perfect for what we wanted: rapid development of web apps without having to bear the overhead accompanying the use of other, more rigid frameworks such as Django. Because of Flask's built-in simplicity to handle web requests and the receptiveness of numerous plugins like flask-CORS, it became the ideal choice for our project; therefore, Flask made it ensured that our application could be accessed from multiple domains securely.

Furthermore, the reason for selecting PyTorch as the deep learning framework was its flexibility due to the capability of dynamic computation flow graphs and better support for GPU. This made it easy to develop models where changes and even optimization are often needed, hence making the thesis more research oriented.

The specific selection criteria used to choose these tools and technologies adapted to a consistent development environment based on the inherent strengths of each tool and technology and the strength of the link between them. This synergy was very useful in developing a sound, scalable, and optimal NLP solution relevant to make Tutoring Agent with generative AI.

5.1.2 Setup Configuration

Due to the computational expenses that are required to train deep learning models, especially large data sets and complicated algorithms, we adopted the use of computationally intensive hardware. In our configuration, we referred to the generation of NVIDIA GeForce RTX 3060 GPUs, the performances of which are highly appreciated for their parallel computing. These GPUs expanded the speed of training and especially of inference, which is paramount for real-time NLP.
The selected CPUs were Intel Core i7-12700H because of their efficiency and raw computing ability. The choice was dictated by the requirement for high calculation speed and the possibility to perform multiple operations without any impact on the performance that can be critical during machine learning models' and data processing tasks' execution. Multitasking and large-scale operations typical under machine learning tasks require efficient systems, for this reason we fitted our systems with 32 GB of DDR4 RAM. This rich memory was important in mitigating against data processing and model training timings. Storage choices were made with a reference to the high speed in both reading and writing we require when working with huge datasets characteristic to NLP jobs. Fast storage also enhanced our system's extensible total throughput since it decreased the loading and preprocessing time for data, thus improving machine learning operational efficiency.

The software environment was set up for the efficient execution of NLP models in the development, testing, and the deployment phase. We utilized Python 3. 8, decision that must do well with the support of the said language in the machine learning community, as well as with compatibility with all the key data science libraries. We set up a virtual location for each venture so that dependencies could be coordinated as well as guarantee that the development setting was free from system-wide Python installations. The fundamental libraries including PyTorch and Hugging Face Transformers were installed using pip, Python's package manager. These libraries were core to our model training and deployment plan. PyTorch met the following criteria as a dynamic framework for model development, and for experimenting with various new developments, while the Transformers library allowed us to use ready-made state of the art pre-trained model which helped us significantly reduce the time for developing models. Regarding GPU acceleration, we installed Nvidia drivers and CUDA toolkit to make sure that PyTorch models could benefit from our GPUs. This setup was critical for improving the two elements critical to the operation of our machine learning models – iteration speed and model training. Flask was set up to be the backbone of this application as it enabled interaction between users and the models through a web application interface. We chose Flask because for our needs it was lightweight enough to do exactly what we needed without the additional features of more complex frameworks. Flask-CORS was also enabled to work on cross origin resources; thus the web apps were safe to request from different domains.

5.2 Implementation Steps

The implementing process of deploying the generative tutoring agent was a series of activities followed to enhance the system's functionality to achieve the intended goal. This detailed guide of the interaction process from the user's authentication to the final delivery of a response involves intricate language and sentiment analysis by machine learning. This is a sequence of activities followed in the implementation process of putting the tutoring bot into practice.

5.2.1 Fine-Tuning of the Model

The method of fine-tuning of the model LaMini Flan T5 248M [12] is clear when it comes to any machine learning based system especially for precision difference making application such as the tutoring bot. For this, we used the Hugging Face Transformers library's Seq2SeqTrainer because it is engineered to help fine-tune the sequence-to-sequence models. It is in those use cases where many sequences need to be transformed into another, such as, for instance, the translation of the text or generation of answers to questions, which are functionalities of our tutoring bot. To elaborate further on the method outlined in step one of our training process, we used checkpoints of pre-existing autoencoder models. These models have been pre-trained on such large datasets for general tasks like the ones we wanted to execute, so this forms a good starting point to fine-tuning. Beginning with these pre-trained models reduces the amount of training that needs to be done because the model as learned a lot of information that is relevant. The next form of data preprocessing that was applied was tokenization of the input data. The process of converting the raw textual data in a way that can be easily interpreted by the sought model known as tokenization where the words or any other segment of text can be represented by numbers. This step is important as it determines the nature of the generated text with respect to the training data. After tokenization, the next step that the model went through was the backpropagation where the weights of the model were adjusted. This is a concept or an algorithm that is common in the usage of neural networks especially in cases where accurate estimates are being sought for. In this approach, the model's internal factors are recalibrated in relation to the current error rate of its outputs. It is important for the model to go through these iterations so that the performance can be optimized.

Finally, the model's accuracy was checked at every stage of the training steps. Unlike other evaluations that are primarily concerned with the correctness of a model while using it, this evaluation was based not only on the accuracy of the model but also on the time it took to process and make the responses. This made it possible to compare the performance of the model to an ideal distribution achieving the intended goal of identifying possible problems or weaknesses that can be corrected on the fly in order to improve the performance of the model. Due to proper management of each of the above phases, it was possible to develop a model capable not only of correctly interpreting and answering the user's query but also of doing it within a reasonable time.

5.2.2 Data Management

Data management is a critical factor in generative AI as the input data affects the final model's efficiency. In the case of data management started with the import of data kept in a JSON file format. JSON is a data format used for transmitting data as text between a server and a web application; It is the short form of JavaScript Object Notation. The data contained within the JSON file was then divided into two distinct sets: training and validation are two virtues of descent methods. The training dataset is used to instruct the model necessary information for its work, that is why this data is also called training data. The validation dataset is, therefore, used to assess the status of the model during the training phase. This is important as it minimise the chances of overfitting wherein the model will do well on the training data but will not do well on any other data. Preprocessing is the next important procedure of data management that involves preparing data for its usage in the model. For our thesis, this meant processing of the dataset by tokenizing it with the help of tools available in the Transformers library. Tokenization is a technique that involves segmenting the text into parts that are easier to work with and include words, phrases, among others. This also involves processes of working with the textual data and transforming such data into a format that is suitable for the model. This made it easy for our model to parse through our data as we made sure that it was appropriately managed, loaded, and pre-processed. This preparation is important in training efficient machine learning models as this contributes to the efficiency of the model's learning and its capabilities to perform tasks accurately.

The Table 5.1 explains the segregation of data into different subsets, which are followed in any machine learning project and are training set and validation set. To use the training set, 80% of total data are used for teaching the model complexity in its distinguishing pattern. The remaining 20% is called the validation set and it is utilized in the model evaluation, in the process of optimizing the model's parameters and to minimize the risk of overfitting. Such a structuring in data helps the model learn it better and apply the learned knowledge to the unseen data.

Table 5.1: Training and Validation Set Table.

Subset	Proportion of Total Data	Purpose
Training Set	80%	Used to train the machine learning model, allowing it to learn and adapt to complex patterns.
Validation Set	20%	Used to validate the model's performance, tune parameters, and prevent overfitting.

5.2.3 Model Inference Pipeline

In a model inference, an already trained model interprets new input data, to make some decisions or draw some predictions. In the case of our tutoring bot, model inference will entail taking the inputted user string tokens, feeding them into the fine-tuned model and returning the output string that will be in turn relayed back to the user through the API. This process starts at the time the user enters a search query through the UI. Tokenization is done first on the given input which leads to converting the text into a format suitable for the model to read. It is then passed through the trained model, the model being able to generate the right output according to the parameters that it learnt. The inference pipeline has been optimized to be very fast so that the users do not end up waiting for responses for very long. Such efficiency is necessary to keep the customers continuing their interactions with the tutoring bot. The response elicited by the model is then passed through the backend API layer where they are delivered to the frontend layer to be displayed to the user. In this case, all the various steps from input, processing, storage and response are efficient and are processed in a short time span, thus making the use of the application fully interactive.

5.2.4 Backend API Development

The backend API can be described as the working core of our tutoring bot as it contains all the calculation and the processing for the answers to the users' questions. Our back end was written in python with Flask and was built to be scalable in order to handle a large volume of connections at once without being affected in terms of the speed with which the system processes the incoming requests. Python was selected as the primary language for backend coding because it is easy and has tremendous backing from libraries supporting web development and machine learning. Flask lightweight web application framework was used to construct the backend architecture. It let us define endpoints that were capable of processing the requests from the frontend in the most optimal manner. The backend API featured two primary endpoints: It is divided into two sub commands, which are chat and train. The main chat endpoint was to get user queries, pass these queries to the machine learning model, and the model provided the responses. This endpoint is paramount to the tutoring bot as it updates the relevant responses in real-time, as requested by the user. While the train endpoint was active for updating the model with new data that is collected frequently. This functionality is crucial for the teaching model to always remain up to date since it is capable of learning from current inputs. Making strong the backend API was necessitated to facilitate the functionality of the tutoring bot while matching its capability of processing and researching queries from any user. Thanks to the choice of the tools, we were able to create a system that can be developed further according to the user load dynamics.

5.2.5 Integration with Frond-end

As for the thesis, the emphasis was made on creating a strong backend that would work in harmony with a comprehensible web interface, thus making user's interaction with the tutoring bot smooth and effective. The backend developed mostly for dealing with data processing, queries and response processing and delivery was well and methodically constructed to have a compatibility with the frontend and to work in proper coordination with it for receiving and transmitting data in real time.

The frontend development of the web application was designed with current web application technologies such as HTML, SCSS, Bootstrap, TypeScript, and Angular. These technologies were selected to provide users with a visually pleasing design and layout to the interface with the processes of navigation and feedback. Angular specifically allows for the creation of a single page application that makes real time interactions possible, for lack of having to refresh the page. Flexibility of the backend and the frontend is achieved using a clear API through which data interchange and requests are issued. This API is the channel through which the questions typed by the user in the frontend form and passed to the backend to be processed. Once the backend processes these queries with the help of algorithms incorporated in tutoring bot, the answers that are received are pushed back to the frontend and displayed to the user. This is essentially the point of contact which the users have with the bot, or more appropriately, the interface component through which communication takes place. It is user friendly, and this makes users to be able to type in their questions without struggling. This is the section where the backend responds to the user question after the user has typed the question and submitted. It helps in increasing the level of utilize engagement and satisfaction due to the immediacy of the response. The communication between the backend and frontend is thus always active, which means real time. This is made possible through a feature in Angular that will allow real time data binding and update from the backend to the frontend interface.

The structural and end-to-end integration between the backend and frontend is necessary in creating an exceptional user interface. In the systems that employ a distinct backend and frontend design, the technical performance of the backend processing is done in a way that seamlessly merges with user interactivity in the front end while making the system not only efficient but also well designed.

The (Fig. 5.1) presented the Concept of the User Interface of a Web Application for a Tutoring Agent, which would focus on enhancing the interaction between the student and the online tutor. After the due time, the interface highlights an animated character on the right which improves the interactive feature of the tutoring agent. At the bottom there is a plain text box with the text input area and a send button with which the user can type in messages and send them to the agent without much difficulty. This setup is designed with an aim to ease a learning process by providing a means of continuous interactivity.

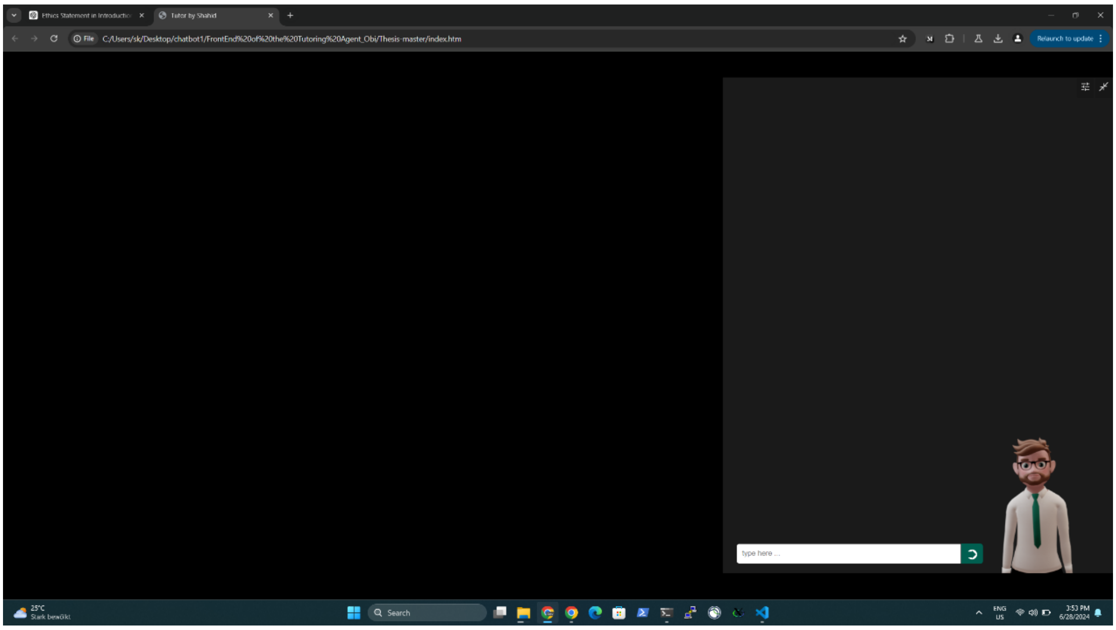


Figure 5.1: User Interface of Web Application of Tutoring Agent.

5.2.6 Emotion Classification

The aspect of emotion classification in the context of the tutoring bot analysis focuses on the emotional tone of the typed responses by the model j-hartmann/emotion-english-distilroberta-base [34]. This process involves the use of an emotion model that is trained in recognizing different emotions from text, thereby enabling the system to change the nature of its prompt in relation to the emotional tone involved in the communication. This is because incorporating the emotion classification into the system enables users to communicate to the system as with a human. The tutors shall also work towards identifying emotional signs from the inputs made by a user so as to produce consonant and hence, a meliorated interaction between the user and the tutoring bot. The emotion classification model is also integrated in the response pipeline where it examines each of the produced responses before it is returned to the user. If an emotion is registered, the system can adjust the tenor or the message conveyed in response to capture the user's mood, thus making the encounter more pleasing. This is the added layer of emotional intelligence that makes our tutoring bot different from simple question and answer AI applications. This makes it possible to have interactions that are deeper which are very important especially in educational settings, where recognizing and responding to emotional states affect learning in a huge way. To sum up, every step of the implementation, starting from the model training, data management, user interface design, and backend development had been designed and developed with high attention to its functionality as well as its effectiveness in making the tutoring bot serve as high-functioning and effective way of creating the tutoring experience.

The (Fig. 5.2) reveals a feature of User Interface of a Web Application for a Tutoring Agent and more interestingly a feature related to emotion avatar. In this interface, the users are offered a choice between two characters Ada and Oli, and both these characters have different facial expressions that tend to make them a little more emotional. This choice gives users a chance to control the form of interaction with the tutoring agent by choosing avatar which seems more suitable for the users or corresponds to their current mood. Introducing emotional avatars as an improvement of the frontend is another strategic addition that intends to bring the tutorial closer to the learner. In addition to positively stimulating the user's participation level while using the application, the inclusion of avatars which can display emotionally expressions add a value to the social interaction with the tutoring agent since such interactions feel more lifelike. This design choice is highly perceptive to the user, where the key to the educational technology is the connection prospective users have to the application.

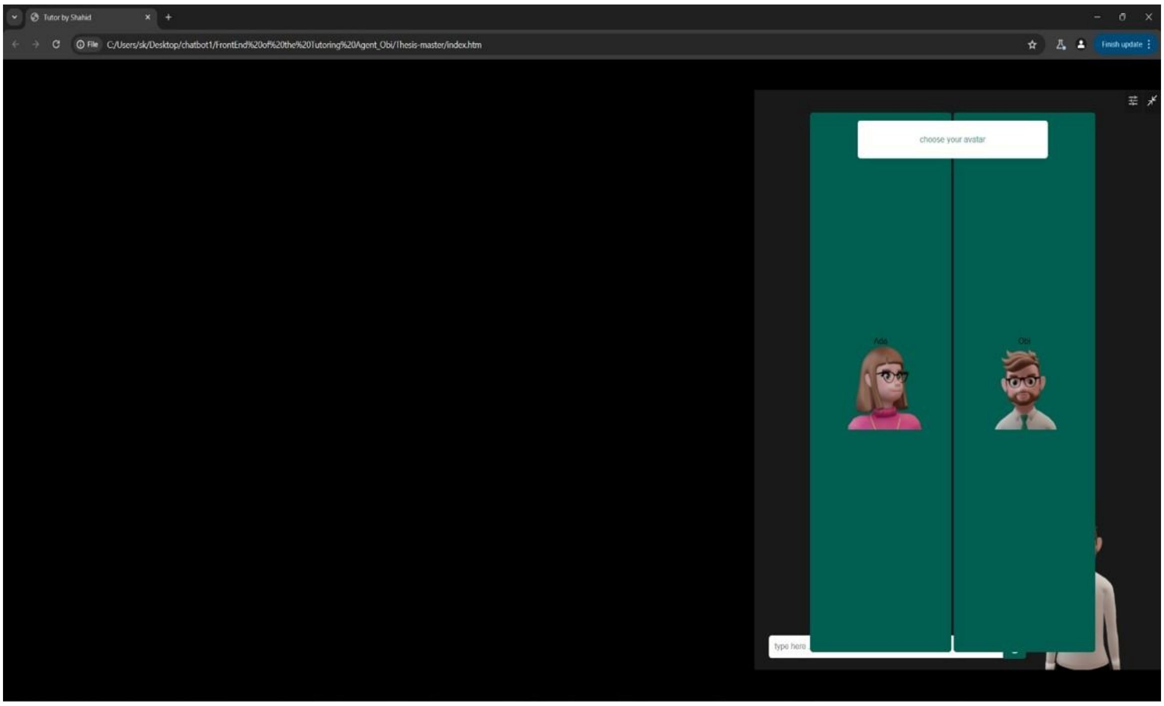


Figure 5.2: Emotion Avatar Integration in Tutoring Agent Web Application.

The Figure 5.3 shows the screen of a chat window, which was employed for the discussion on the incorporation of ethical statements into academic research. Two of them are named, male and female avatars, and they discuss usual procedures to integrate ethical aspects in the documentation of the research.

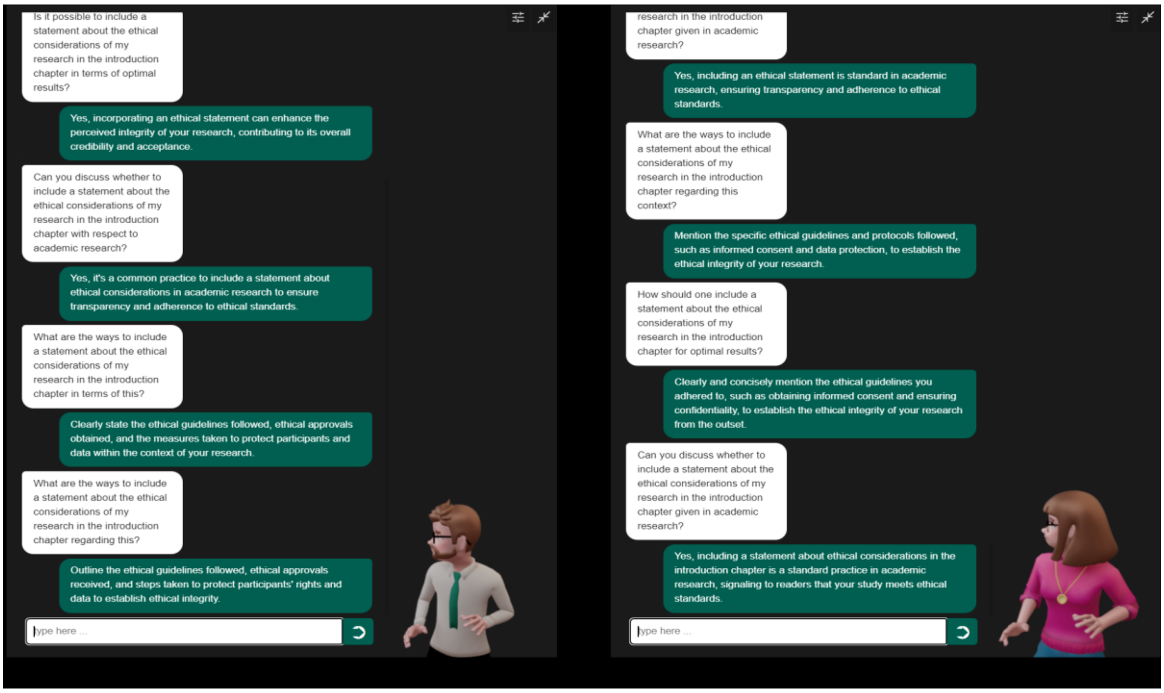


Figure 5.3: Response of Emotion Avatar.

5.2.7 Dashboard Integration

Integrate a session-based authentication system using cookies between a React dashboard [35] and a JavaScript-based chatbot, the process is streamlined and secure. When a user logs into the React dashboard, the Flask backend generates a session, which is stored in a session cookie sent to the browser. This cookie is automatically included in every request made from the React app or the chatbot, eliminating the need to manually pass tokens or credentials.

In the React dashboard, the chatbot is embedded as an iframe. Since the browser automatically sends the session cookie along with any requests from the iframe, there’s no need to explicitly share authentication details. The chatbot’s JavaScript can then send a request to the Flask backend, and the session cookie is included. Flask verifies the session and responds with user-specific data, such as their profile or relevant course information. The chatbot uses this data to personalize its interactions with the user.

To enhance security, the session cookie should be configured with attributes like Secure to prevent access from JavaScript, ensure cookies are sent only over HTTPS, and mitigate cross-site request forgery (CSRF) attacks. This setup ensures that both the React dashboard and the chatbot share the same login state seamlessly, providing a secure and personalized experience for users without the need for complex token handling.

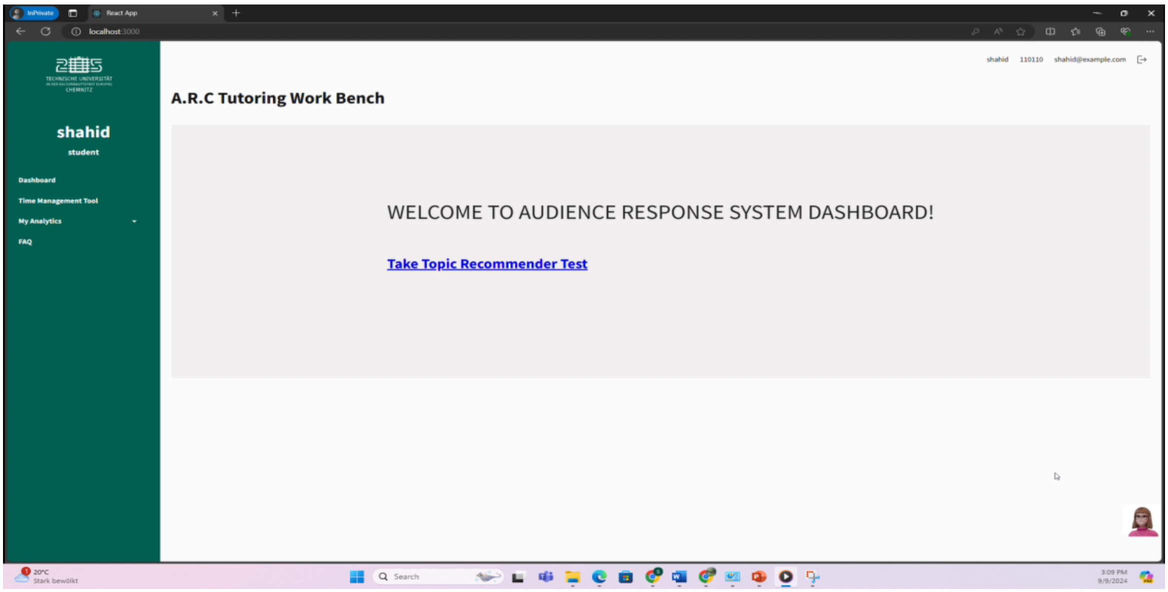


Figure 5.4: A.R.C Tutoring Dashboard [35].

5.3 Testing and Debugging

Performed through a test and debug of our system to make sure that everything was working properly. Prepare: Here are the major tactics and techniques we used, as well as suggestions for adding project shots. Thorough testing approaches -unit, integration, system and performance tests that guaranteed the systems reliability. Those tests certified that our system components were just doing the right thing, interacting as expected altogether and able to withstand full range of operating conditions. Solid debugging solutions used publishing, exceptional handling as well as a reside keeping track of. With these practices, issues were solved quickly, and it was easy to know if the system is still up without downtime.

5.3.1 Unit Testing

As for Unit testing, the main objective was in confirming that each fragment of code functions was operating effectively. Specifically, we relied on the Python unit test framework to develop and run tests specifically optimized for the respective part. This phase involved testing of modules such as the data loading, pre-processing and the API handles. What this essentially meant was that we could test one segment of the system in isolation and gain insights on how to correct it if at all it in compiled in the wrong format since every module was independent of the other.

5.3.2 Integration Testing

After Unit testing, we proceeded towards Integration Testing in which the integration and cooperation of the different components in the system was tested. The specific aspects of testing features were accomplished with the help of the pytest framework and the Flask extension for this purpose. For example, we validated the possibilities of the system when answering a question entered through the web application by receiving an answer from the backend API. This was essential mainly to facilitate an assessment of the level of viability in the integration of the front-end and the back end of the system to support the interaction between system components.

5.3.3 Performance Testing

Last but not the least; Performance Testing was performed to test the competency of the system to work at different condition and in different capacity. To test response time and system stabilities, we implement heavy user loads by using locust and JMeter. This phase made it possible to establish the maximum capacity of the system and ascertain when the system’s capability started declining. So, the performance testing had a key role in determining that the created system can correspond to real heavy traffic loads and remain stable in its work during the critical usage.

All these testing phases are used in our development cycle and were very helpful to systematically explore and resolve possible weaknesses. Where as in Unit Testing it made sure that all the components involved were working in the right modes and manners, in Integration Testing it confirmed that all the sub systems selected for integration worked in harmony with each other, and in Performance Testing it ensured how our system could endure in the best possible manner, these methodologies proved to be very valuable and effective in giving a real touch to our system before it would be actually put to use.

5.3.4 Exception Handling

Regarding Exception Handling, we aimed to having error control to avoid the termination for various or unexpected circumstances. For exception handling, we used Python, which provides multiple possibilities for creating unique exception handling systems, according to the problem. With the help of try and except blocks used in the program, we managed to see how exceptions look like and where they are generated. This way of structuring also eliminated the abrupt termination of applications and enabled the reporting of detailed error messages that proved helpful during debugging. This approach of exception handling enabled our application to run smoothly when it encountered some runtime errors thus increasing convenience and robustness of the application to the users.

5.3.5 Debugging Tools

As for Debugging, the idea was to quickly identify and resolve problems within the code of the given program. We used various strong tools such as pdb (Python Debugger), and other inbuilt debuggers which are incorporated in the Python IDE systems such as PyCharm and VS Code. These tools aided us in being able to freeze the code and look at variables and the flow of the program execution. This capability served as useful while identifying the nature of problems, including their exact coordinates and the causes of mistakes. This made us to fine tune our code repeatedly and to improve on the system performance and stability with the above advanced debugging tools.

This ability of exception handling and debugging in the development process is helpful in making sure that the tutoring agent can execute under different circumstances. It allowed identifying the potential problems and eliminate them and improving the system step by step; thus, we achieved the goal of providing users with the well-designed tutoring environment [36].

5.4 Challenges and Solutions

We ran into a few challenges during the development and deployment of our project, each one we needed to work around for it not to cause issue at runtime. This is the detail of challenges encountered with respective solutions, our system requirements and future work.

5.4.1 Model Selection and Resource Constraints

Initially we adopted the Llama2 model (which is a sophisticated one with several capabilities) Nevertheless, the computational resources needed to execute Llama2 or even further optimize it were out of reach. Llama2 needs a high-end GPU with at least 24GB of VRAM. We then moved onto the Mixtral model [37] which again needed heavy resources 16GB of VRAM that we did not have. Afterward, we decided to use LaMini-Flan-T5-248M a lot because there were hardware limitations. Llama 2 is an advanced language model that has been developed by Meta (formerly Facebook). It belongs to the family of transformer-based models which are built for a wide range natural language processing task. Mixtral, a language model from Mistral AI designed with efficiency and adaptability for various natural language processing tasks, while Llama 2 and Mixtral are more performant model for optimized real-time applications on constrained resources; LaMini-Flan-T5-248M is a less powerful, but better suited to be used in environments that have limited capability. Thus, the LaMini-Flan-T5-248M model is recommended in constrained-resourced infrastructures, despite offering inferior performance than Llama 2 and Mixtral.

The (Fig. 5.4) shows a pseudocode describes a procedure that inquires whether a system has 24 GB of VRAM to reach an application called “Llama 2.” A function to obtain the available VRAM is present and returns 6 GB in this case. Another function checks whether this amount is greater or equal to 24GB. If insufficient, the main function outputs a message saying that Llama 2 cannot run because of the lack of VRAM and thus prevents the application to run on under-equipped systems.

```
// Pseudocode to infer Llama 2 failure on 6GB VRAM

// Define the minimum VRAM required for Llama 2
MIN_VRAM_REQUIRED = 24  // Llama 2 requires at least 24GB of
VRAM

// Function to get the total VRAM available on the system
function getSystemVRAM():
    return 6  // Example: Returning 6GB for this example

// Function to check if the system VRAM is sufficient
function isVRAMSufficient(vram):
    return vram >= MIN_VRAM_REQUIRED
// True if VRAM is sufficient, else False

// Function to attempt running Llama 2
function runLlama2():
    vram = getSystemVRAM()
    if not isVRAMSufficient(vram):
        print("Llama  2  did  not  work:  Insufficient  VRAM
(minimum required is " + MIN_VRAM_REQUIRED + "GB)")
    else:
```

Figure 5.5: Pseudocode of Insufficient VRAM Error for Llama2.

As a way of solving this challenge, we came up with a strategic solution that entailed a change of our selection criteria as well as carefully regulating the resources available to the model. First, we planned to use the high performance Llama2 model, however, its demands were highly beyond of our computational power. We then tested the Mixtral model which we found equally drained a lot of resource from our infrastructure. At long last, proposed the LaMini-Flan-T5-248M which, though not as powerful as the others, would be far more appropriate given our current assets. By applying this change of a model from the previous architecture to LaMini-Flan-T5-248M, we were able to get used and optimized all the available resources without having to invest in new equipment. Additionally, due to the concern of using this model within the capacity of our system, we aimed at enhancing our resource management to enhance the implementation of this model. Thus, the problem of integrating and the operation of the LaMini-Flan-T5-248M model did not affect the system's efficiency and performance due to the proper organization of computational resources.

5.4.2 Data Limitations

We have an approach of building our model with TensorFlow and Keras (our target is to pass the test using this trained TF Keras, but it was not giving any output) The only issue here is that the dataset being so small, caused problem for we in scoring and generalization. While setting up the machine learning model, TensorFlow and Keras were selected based on their great performances in most complex tasks related to neural networks. The objective was to not only train a model with these frameworks, but to also follow it with a set of tests for its verification. However, we encountered a significant hurdle: The situation in the testing phase was that the model was not giving any outputs. This problem was identified to have stemmed from our data collection where the size of the data was small, which is a typical problem in machine learning and artificial intelligence applications.

We therefore took an approach to controversially deal with the problems emanating from use of limited data. Firstly, we changed to the LaMini-Flan-T5-248M model that has been trained on sets significantly greater than ours. Having such a preset model comes with the benefit of transfer learning from the larger datasets to smaller and specific ones. This shift was made to focus on the generality of the model so that the model could do better with what little data we have now because of the other learning it has been given. Secondly, the set of data augmentation transformations was used to artificially increase the volume and variety of the training data set. This method implies creation of new instances from the existing ones by virtue of various transformations and modifications, thus increasing the size of the training set and diversifying it. This diversity is valuable for increasing its capacity to predict more varied outcomes and inputs from the current model. By means of data augmentation, the volume of dataset was augmented beyond the simple increase in the amount of data but in variability as well, which positively impacted the overall performance as well as ability to generalize of the model in real life applications. In combination, the applied approach, which combined the use of a pre-trained model and data augmentation, allowed for the development of reliable approaches to addressing the issues arising from the initial data scarcity, thus contributing to the enhancement of the model's performance in practical applications.

In summary, the development and implementation of the tutoring agent were characterized by detailed planning and execution across software development and machine learning domains. This comprehensive approach resulted in a sophisticated educational tool that provided personalized, interactive learning experiences, significantly enhancing user engagement and satisfaction.

Before the implementation of the project, there is a planning phase that carefully establishes the functional requirements as well as the technical specifications of the tutoring agent. Moving on this phase involved an identification of appropriate technologies and frameworks that would be utilized in the event to allow the agent to enhance its advanced machine learning with sound software settings. Special emphasis was placed on the choice of the machine learning model that would serve as the basis for the formation of the agent—LaMini Flan T5 248M, which proved itself well as a performer of sequence-to-sequence transformations that are paramount for understanding and answering the user's queries. During the implementation phase, the creation of the machine learning environment was done through the Hugging Face Transformers library's Seq2SeqTrainer for the chosen model's fine tuning. This step involved drawing on existing information that is inherent in pre-trained autoencoder models this greatly reduced the training time. This was accompanied by sophisticated methods in data handling which preprocessed and arranged the input data in the most suitable form for model learning. The signal was also converted to the machine-readable form for analysis and the process known as backpropagation was applied to the parameters to improve upon the model. Parallel to the aforesaid machine learning setup, the software development team, with an aim to provide an easy and interactive graphical user-interface has been developed using HTML, SCSS, Bootstrap, TypeScript, and Angular. Indeed, this interface was initially designed in such a way that it could easily interface with the backend API which, in this case, was developed using Python and Flask, to process and deliver back the responses depending on the modeled results obtained. This backend setup was significant in controlling the traffic in the system as well as in making the system highly available despite high loads from the users. To enhance the user interaction capabilities, an emotion classification model was incorporated for determining the affective tone of the user inputs, and thus the tutoring agent was capable of not only providing correct answers but also answering with appropriate emotion. This feature was meant to individualize the interactions and improve activity engagement by taking into consideration the mood of the model when answering, thereby improving efficiency of learning. Testing was also strictly followed in the project, where unit testing took place for each component respectively, application integration testing to check the integration of the different applications and performances testing to assess the efficiency of the operations. This meant that the system had undergone extensive testing which made it virtually sound, efficient and capable of providing quality education.

In conclusion, the given and detailed and integrated ITS design and implementation process came to the creation of a very effective tutoring agent to be used in education. It not only supplemented the capability of interaction and customization of learning but also greatly boosted the engagement and satisfaction level of its users thus effectively revolutionizing the educational interactions by use of Technologies.

6 Results and Evaluation

This section of the study gives an in-depth indentation of processing results, model performance measures, confusion metrics measures, visual evidence and user feedback, along with an evaluation of the overall system performance for tutoring agents. This section of the study gives an in-depth indentation of processing results, model performance measures, and user feedback, along with an evaluation of the overall system performance for tutoring agents. The model performance metrics are centered around the Model LaMini-Flan-T5-248M, for which we compute Accuracy, Precision Recall and F1 Score as key performance indicators. Various components of the detailed confusion matrix were visualized to better understand how well its model can discriminate between correct and incorrect responses, highlighting where it can be improved. Surveys were conducted to collect user feedback on the perceived ease, response time, accuracy, and emotional relevance through which the Tutoring agent was responding. This was critical in knowing how satisfied users were and what areas needed it most. Based on the survey results, users rated different parts of the system, such as satisfaction with the overall experience and navigation ease along with Response speed and Design of web application or website. Most users found the system easy to use and were generally pleased with its design, although some areas for improvement were noted, as well as customization solutions. We have evaluated the performance of our system (e.g., computational efficiency, resource utilization) to ensure that it can meet underlying project objectives. It was based on extensive system latency measurements, resource utilization and the effect sizes of some data augmentation techniques. This made it optimal from a performance standpoint, also demonstrated the need to have a trade-off between accuracy and computing time so that the system can answer as fast as possible in an accurate manner but not consume too many resources. This section will evaluate the performance of the tutoring agent system based on model evaluation and user feedback. The results emphasize some of the success factors and best practices in this implementation, such as performance balance and usability; however, at the same time, areas have been identified for further future improvements to keep up with system evolution and adhere to better responsiveness towards user requirements [38].

6.1 Visual of Tutoring Agent with Emotional Avatar

For describing the results of the study conducted in the framework of the master’s degree thesis concerning the development of an educational chatbot, can provide a detailed description of the evidence of work in the results chapter. All ‘figures’ depict unique scenarios in which the actual interaction with the chatbot occurs and demonstrates related components along with the overall efficacy of the proposed system within an actual learning environment. The first (Fig. 6.1) showcases the initial user interface of the chatbot, featuring a welcoming message from the chatbot. Being greeted by the message “Hello, how may I assist you today?” it creates a rather friendly and welcoming environment to the user. This figure shows that the developed an easily navigable and attractive chat interface, with the input bar at the bottom for type in the inquiry. The simplistic overall look of the design is optimal and means that it meets the functionality without any complications and is accessible to users with any level of technological literacy.

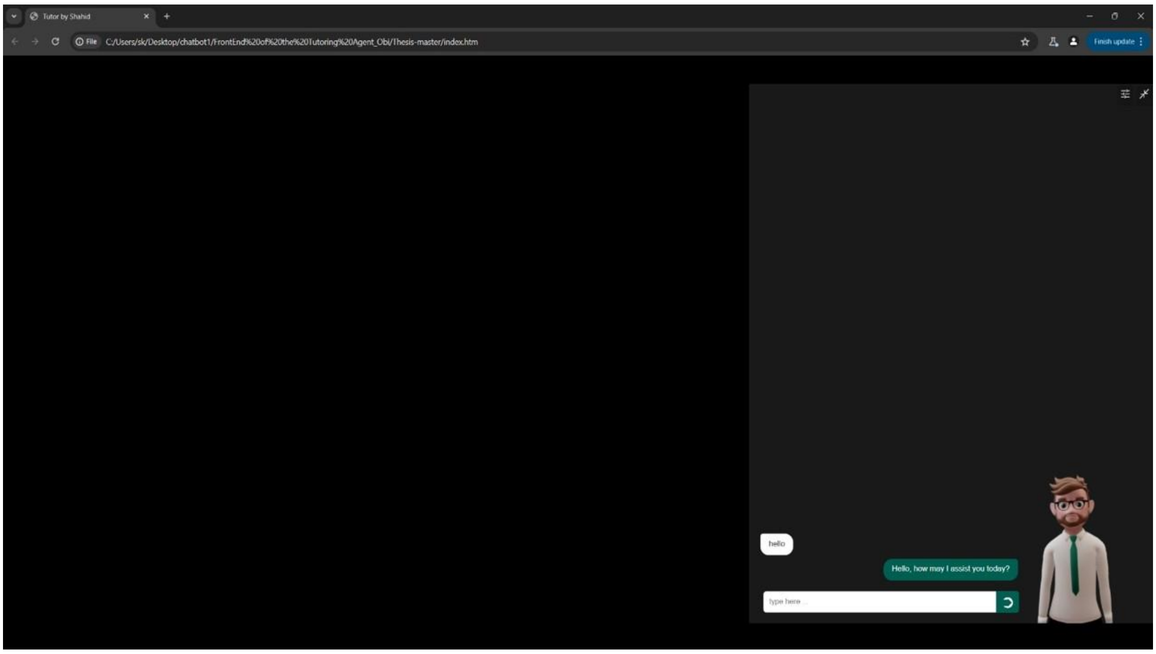


Figure 6.1: Initial Interaction and User Interface.

This (Fig. 6.2) is important since it reveals the complexity of the advice that it can offer to the users, which can be useful for navigating the ethical questions related to research, thereby asserting the educational function of the chatbot. Basically, there is a notion elaborating how ethical principles should be applied within the educational process and academic research. It offers a detailed answer concerning the need to seek ethical clearances, participant’s rights and the significance of relevance.

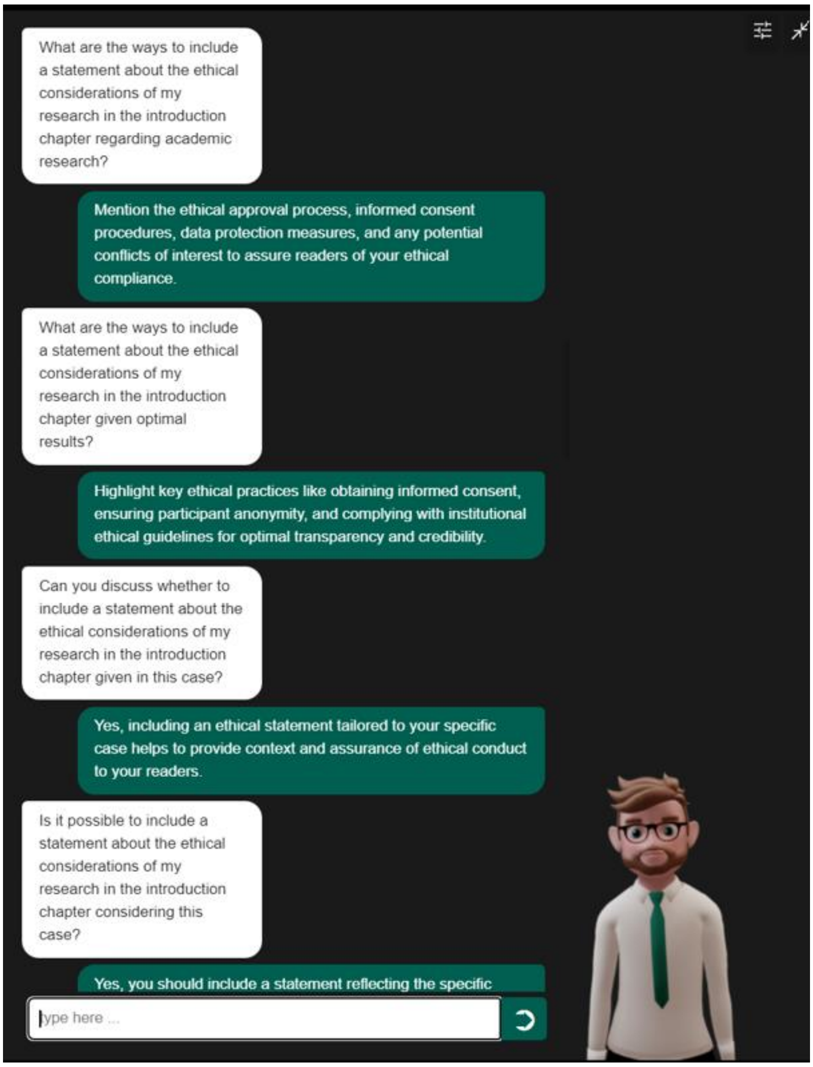


Figure 6.2: Response about Continued Ethical Standards Discussion.

The table 6.1 below summarizes the animations and gestures used by the emotional avatars, detailing the intended words or phrases that trigger these responses and illustrating how they are connected to the backend. This table is crucial as it demonstrates the interaction dynamics and emotional intelligence of the tutoring agent, providing insight into how the system interprets user inputs and generates appropriate emotional responses. The emotional avatars in the tutoring agent are designed to enhance user engagement by responding with human-like emotional expressions. These avatars, named Ada and Obi, can display various emotions such as welcome, happiness, idleness, sadness, surprise, and goodbye. Each emotion is triggered by specific words or phrases detected in the user's input. The backend processes these inputs, classifies the emotions, and maps them to predefined animations and gestures, which are then rendered in real-time on the user interface. This integration of emotional responses ensures that the interactions are dynamic, engaging, and emotionally intelligent.

Table 6.1: Animations and Gestures and Their Backend Connections.

Question	**Generated Answer**	**ML-Based Sentiment**	**Keywords**	**Sentiment**	**Emotion**
Hi! Good morning! Hey there!	Hello! Morning! Hey! Good, you?		Hello, Hi, Hey, Morning Afternoon, Evening etc	Welcome	
Did you like the food? How was your day?	It was fine.	Neutral 😐	-	Idle	
Why are you crying? What happened at the meeting? Are you okay?	I just feel sad. It didn't go well, I'm disappointed.	Sadness 😭		Sad	
Why are you smiling? Are you happy about promotion?	I'm just happy! I got some great news. I'm thrilled!	Joy 😀		Happy	
What just happened? Did you hear the news? Why do you look shocked?	I can't believe it! Yes, it's unbelievable! It really surprised me!	Surprise 😳		Surprise	
I am leaving now. It is time to go. I will be heading out soon. We will catch up later.	Goodbye! See you later! Take care! Talk to you later!		Goodbye, See you See you later Take care	Bye	

We use the Hugging Face model Emotion English DistilRoBERTa to detect sentiments based on the questions asked and the answers given. The model outputs sentiments such as anger, disgust, fear, joy, neutral, sadness, and surprise. We then map these sentiments to our predefined emotions, as shown in the chart. Additionally, we use specific keywords to map welcome and bye interactions.

Here is an example of how this mapping is done, illustrated in the table:

- **Question:** The input question from the user.
- **Generated Answer:** The system's response to the question.
- **ML-Based Sentiment**: The sentiment detected by the Hugging Face model.
- **Keywords:** Specific keywords used for detecting welcome and bye interactions.
- **Sentiment:** The final sentiment category used in our system.
- **Emotion:** The corresponding emotion displayed by the avatar or animation.

Combining the model's output and keyword detection ensures accurate and contextually appropriate emotional responses.

The backend system processes the user's text input received via the chat interface. Initially, the input is pre-processed to remove any extraneous characters or noise that might affect the accuracy of emotion detection. Once pre-processed, the text is fed into the emotion classification model (j-hartmann/emotion-english-distilroberta-base). This model analyzes the text to detect any underlying emotions by identifying keywords, phrases, and the overall tone of the message. Based on the detected emotion, the backend categorizes the input into one of the predefined emotion classes (Welcome, Happy, Idle, Sad, Surprised, Bye).

Each detected emotion is then mapped to a specific animation and gesture of the avatar. The backend contains a predefined set of animations and gestures corresponding to each emotion class. The mapping process involves selecting the appropriate animation sequence based on the emotion category. Once the appropriate animation and gesture are selected, they are sent to the frontend to be rendered in the user interface. The frontend displays the avatar’s response in real-time, ensuring the avatar’s expressions align with the user’s emotional state. The backend continuously monitors user inputs to dynamically update the avatar’s emotional state and responses as the conversation progresses.

The emotional avatars used in the tutoring agent, named Ada and Obi, are designed to enhance user engagement by providing human-like emotional responses. These avatars can display various emotions such as welcome, happiness, idleness, sadness, surprise, and goodbye. Each of these emotions is triggered by specific words or phrases detected in the user's input, allowing the avatars to respond in a manner that feels natural and emotionally intelligent. Here’s a detailed description of each emotional response and how it is triggered and displayed:

6.1.1 *Welcome Emotion*

When the user greets the tutoring agent with words such as "Hello", "Hi", or "Hey", the backend system processes this input to detect a welcoming emotion. The emotion classification model identifies these keywords as indicators of a greeting. Consequently, the backend triggers the avatar, either Ada or Obi, to raise their arms in a welcoming gesture accompanied by a smile. This initial interaction creates a positive and inviting atmosphere for the user, setting a friendly tone for the conversation. The corresponding (Fig 6.3) illustrates this welcoming gesture, emphasizing the avatar’s role in making the user feel greeted and acknowledged right from the start.

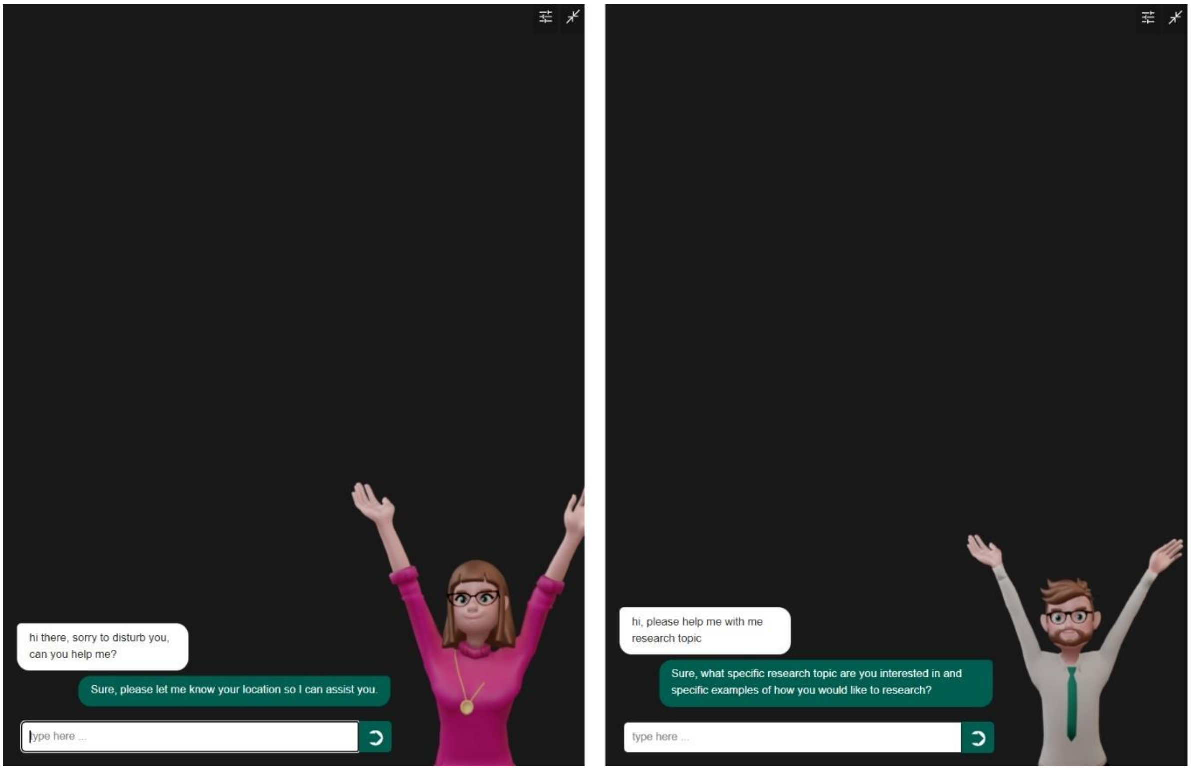


Figure 6.3: Welcome Emotion.

6.1.2 *Happy Emotion*

Positive feedback from the user, such as "Thank you", "Great", "Awesome", or "Good job", prompts the emotion classification model to detect a happy emotion. The backend maps this emotion to a predefined animation sequence where the avatar displays a joyful expression with a wide smile and animated features. This reaction enhances the user experience by mirroring their satisfaction and reinforcing the positive outcome of the interaction. Figure 6.4 captures this precise moment, showcasing the avatar's ability to respond with happiness. The image illustrates the avatar with a wide smile, animated facial features, and an overall demeanor of joy. This visual representation highlights the advanced emotional intelligence of the tutoring agent, demonstrating its capability to engage users in a more meaningful and emotionally rewarding way.

This happy response is designed to make the interaction more engaging and emotionally satisfying for the user. By mirroring the user's positive emotions, the avatar creates a feedback loop that encourages further positive interactions. The user is likely to feel more connected to the tutoring agent, perceiving it as more empathetic and responsive. This emotional resonance is particularly important in educational settings, where a positive emotional state can significantly enhance the learning experience.

In summary, the backends' ability to map detected emotions to appropriate animations and the frontend's capability to render these animations in real-time are crucial for creating a lifelike and engaging interaction

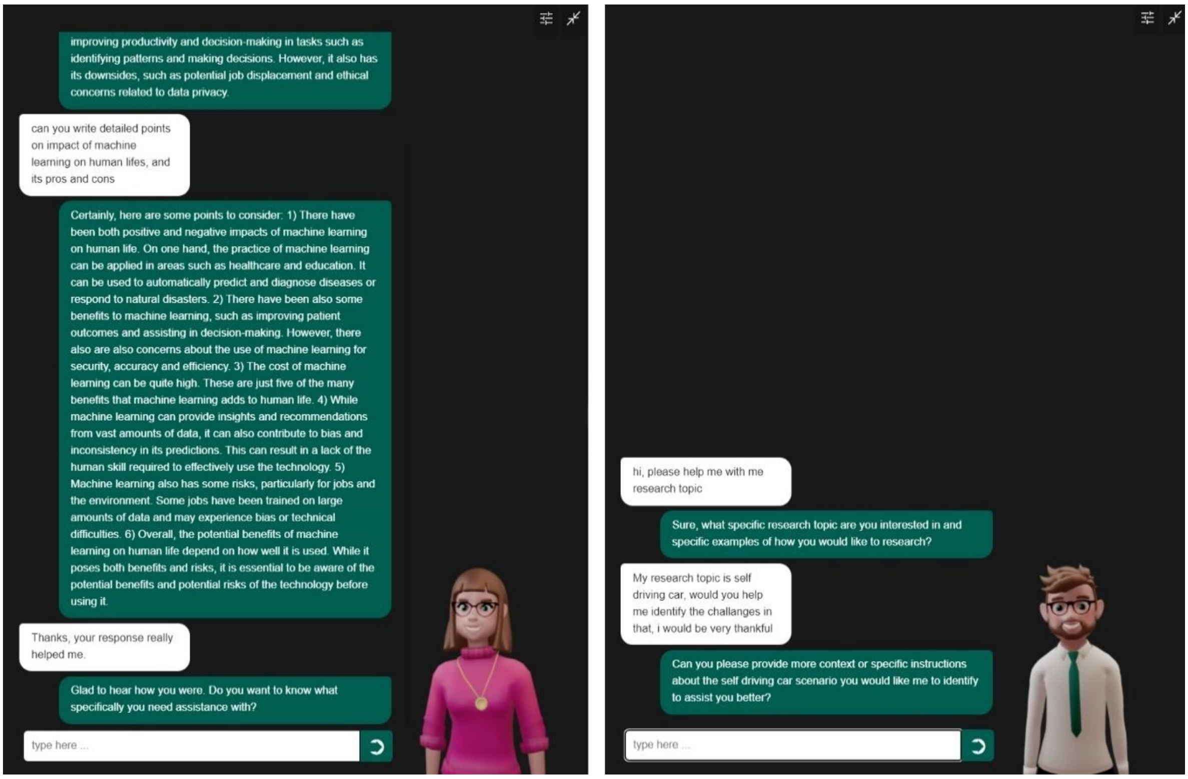


Figure 6.4: Happy Emotion.

6.1.3 Idle Emotion

When there is no user input for a specific duration (e.g., 30 seconds), the backend detects inactivity and triggers the idle emotion response. The avatar does not show any positive or negative attitude to a particular client and offers a composed look that shows the avatar is willing and ready to help in anyway but with preference to the user present. This state makes the user always feel that there is a companion who is with them, even when there is no active conversation being carried out. In this position, the (Fig. 6. 5) shows the avatar in this stable and dependable position in the chat interface, waiting for the user to return to chat, for a reply. Such a transition is beneficial in that it maintains continuity throughout the user's interaction with the agent. The benefit of transitioning to a neutral pose and staying in a neutral pose until the user re-engages also contributed positively to the over quality of interaction and decreed the tutoring agent is more natural and smarter in some way. Therefore, the idle emotion response as part of the tutoring agent's interaction dynamics is very important. It is useful in establishing a more permanent and credible form a foundation, which improves the utility and appeal of the tutoring agent. Figure 6. 5 is a good example of this middle ground where the focus is put on the avatar as the supporter of a conversation in the chat window.

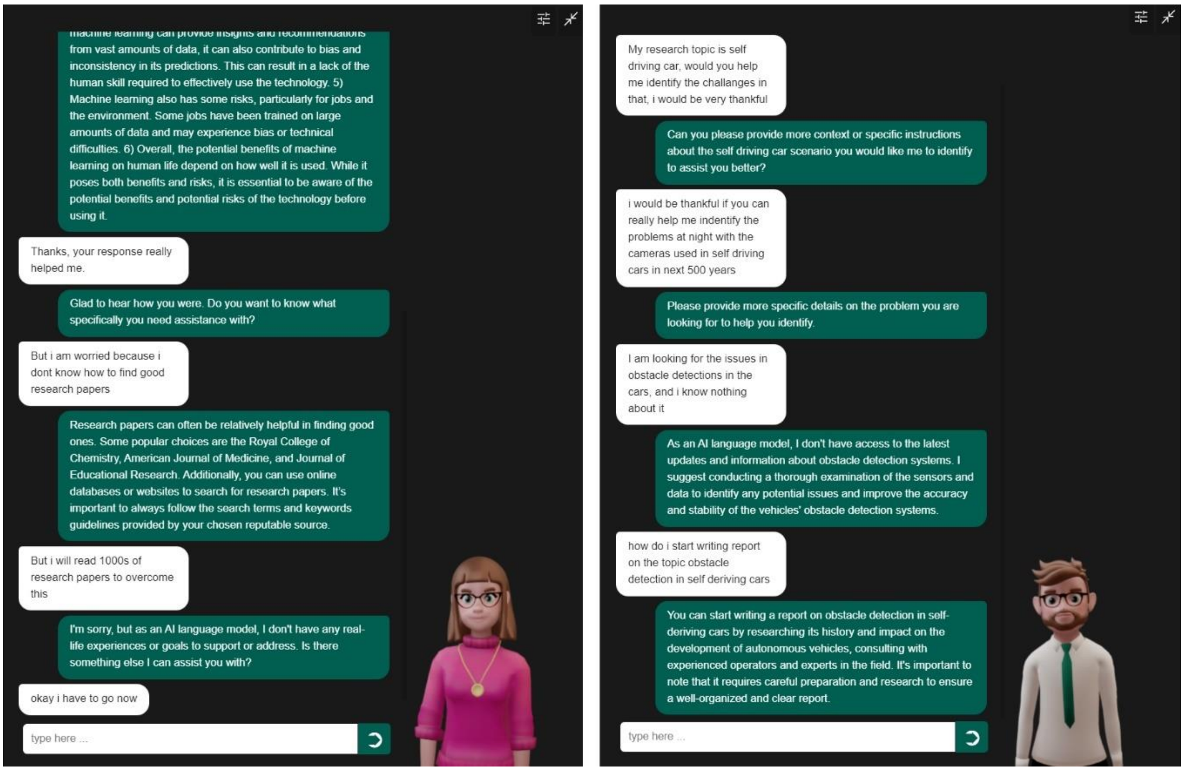


Figure 6.5: Idle Emotion.

6.1.4 Sad Emotion

Any instance where the user types something like, 'I don't understand', 'This is hard', or simply types 'I am frustrated' is flagged by the emotion classification model as a case of the sad emotion. The backend may link this emotion to an animation in which the figure pulls the corners of its mouth down and the opening of the eyes just a little to portray sadness. Indeed, such a response fosters a more profound relationship with the user since the avatar acknowledges and shows concern for the user's difficulties. Thus, if the user type things like I don't understand, this is hard or I am frustrated the emotion classification model takes these as a sign that the user is sad. This input is then parse by the backend system which looks for the emotion and when it is determined to map to a sad expression, the avatar's appearance will reflect this. According to this aspect, the eyes should be slightly drooped, and the mouth is frowning, which represents sympathy and comprehension. This empathetic response can go far in developing rapport with the user because the avatar has identified with the user's pain. The real-time rendering of this animation makes the avatar's reaction proximal and pertinent, so the interaction continuity and emotional care are not disrupted. The Figure 6.6 shows the use of an avatar when the user needs someone to comfort them during difficult stages.

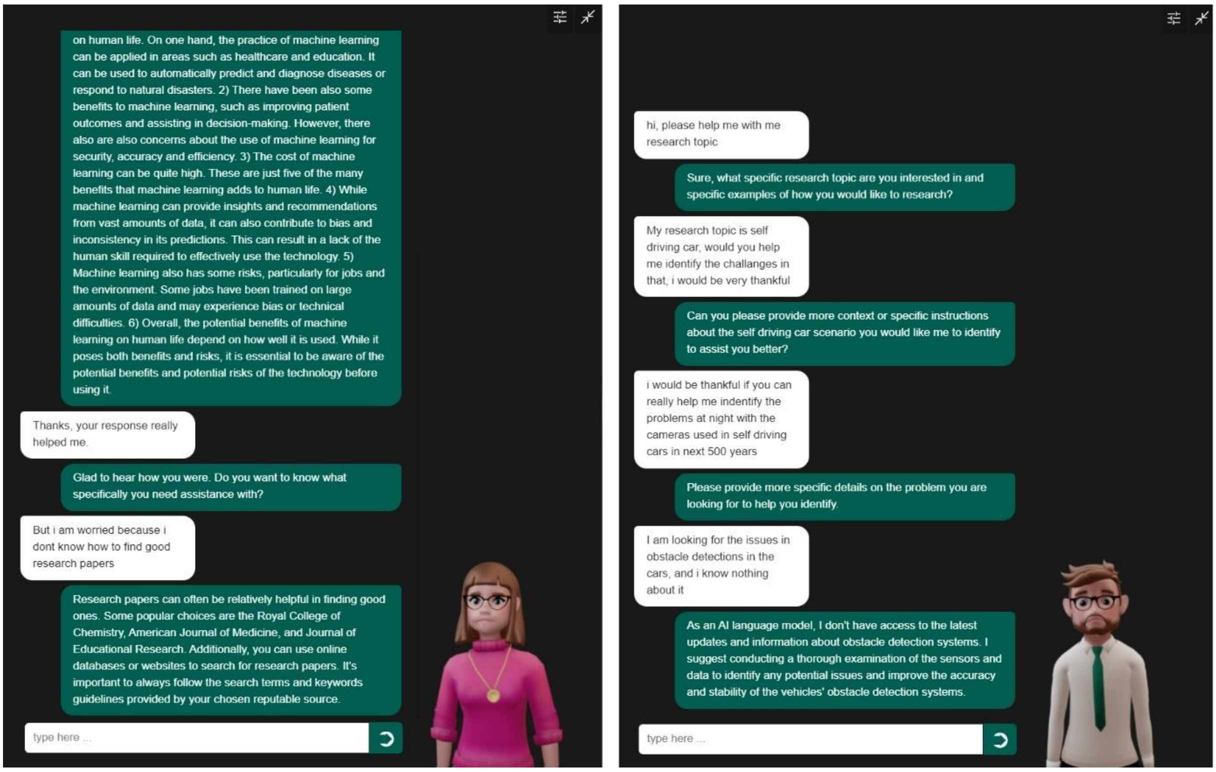


Figure 6.6: Sad Emotion.

6.1.5 Surprised Emotion

Unexpected or unusual inputs from the user, such as "Really?", "Wow", or "I didn't know that", trigger the surprised emotion response. The emotion classification model detects these keywords and maps them to a surprised animation. The avatar responds with wide eyes and an open mouth, expressing surprise and acknowledging the user's input dynamically. This reaction adds a layer of engagement to the interaction, making the conversation feel more lifelike and spontaneous. The surprised emotion response is a key feature of the tutoring agent's interaction dynamics. By detecting unexpected or unusual inputs and responding with an animated expression of surprise, the avatar enhances the conversational experience. Figure 6.7 effectively demonstrates this reaction, showcasing the avatar's ability to acknowledge and reflect the user's emotions, thereby making the interaction more engaging and lifelike.

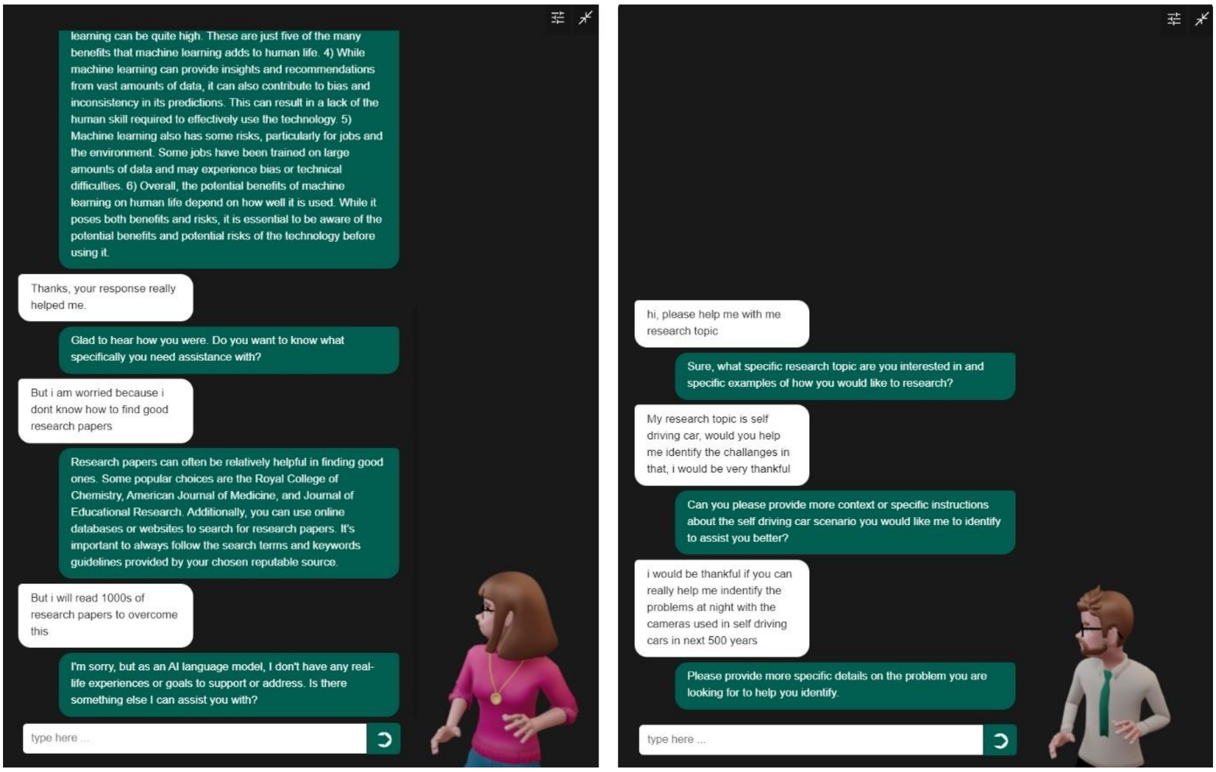


Figure 6.7: Surprised Emotion.

6.1.6 Bye Emotion

When the user types of farewell phrases like "Goodbye", "Bye", or "See you later", the emotion classification model detects a bye emotion. The backend then triggers the avatar to wave goodbye with a friendly smile. This gesture signifies the end of the interaction in a personal and considerate manner, leaving the user with a positive impression. The (Fig. 6.8) depicts the avatar's friendly farewell gesture, ensuring that the user departs feeling valued and appreciated, which can encourage them to return for future interactions.

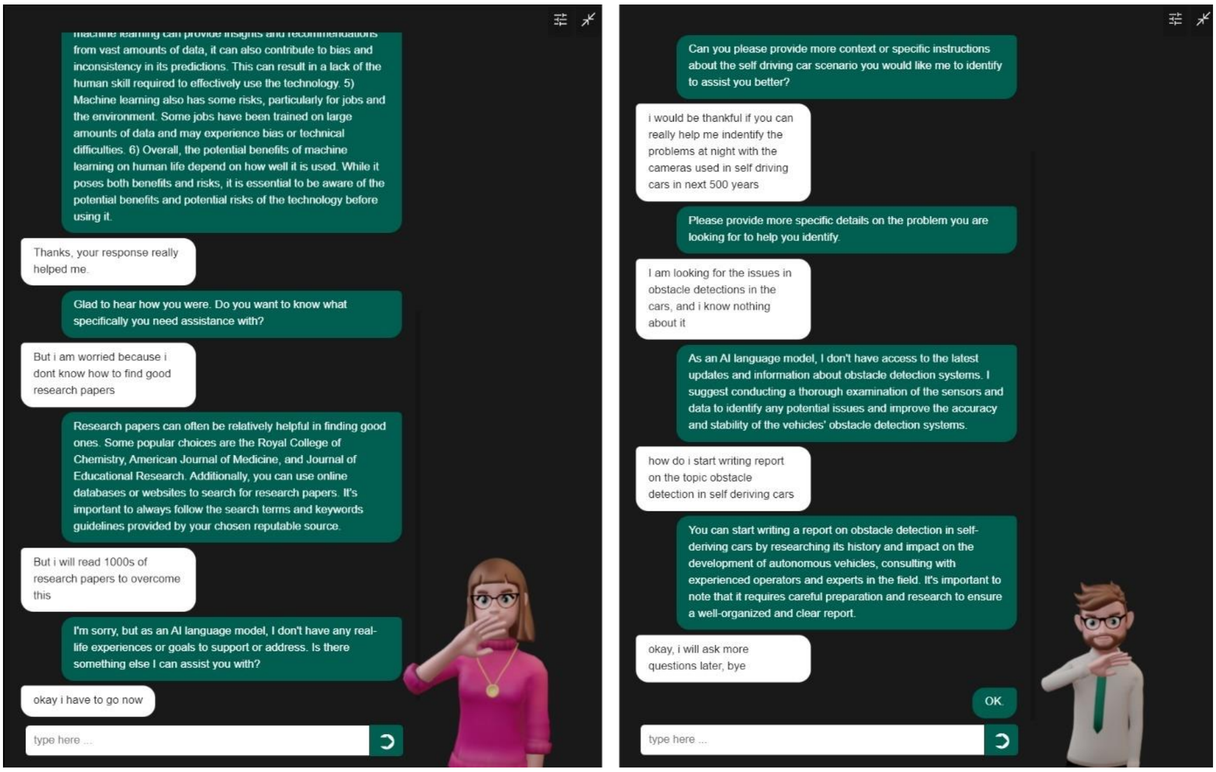


Figure 6.8: Bye Emotion.

By incorporating these varied emotional responses and animations, the tutoring agent effectively enhances user engagement and provides a more human-like interaction experience. The combination of natural language processing and emotion classification ensures that the avatar can respond appropriately to the user's emotional state, making the educational interaction more personalized and effective. The detailed integration of these emotional gestures showcases the advanced capabilities of the tutoring agent in creating an immersive and supportive learning environment.

6.2 Visual of Dashboard

The (Fig. 6.9) shows the final version of the A.R.C Tutoring Work Bench within the Audience Response System dashboard. This interface provides a clear and user-friendly layout designed for students, with the user's name ("shahid") displayed on the left side, alongside a menu that includes options like Dashboard, Time Management Tool, My Analytics, and FAQ. These features help students manage their academic activities effectively.

One notable feature in this dashboard is the chatbot icon located in the bottom-right corner, designed to provide real-time assistance and enhance the overall user experience by answering questions or offering guidance. This user interface balances ease of navigation with essential tools to support academic progress.

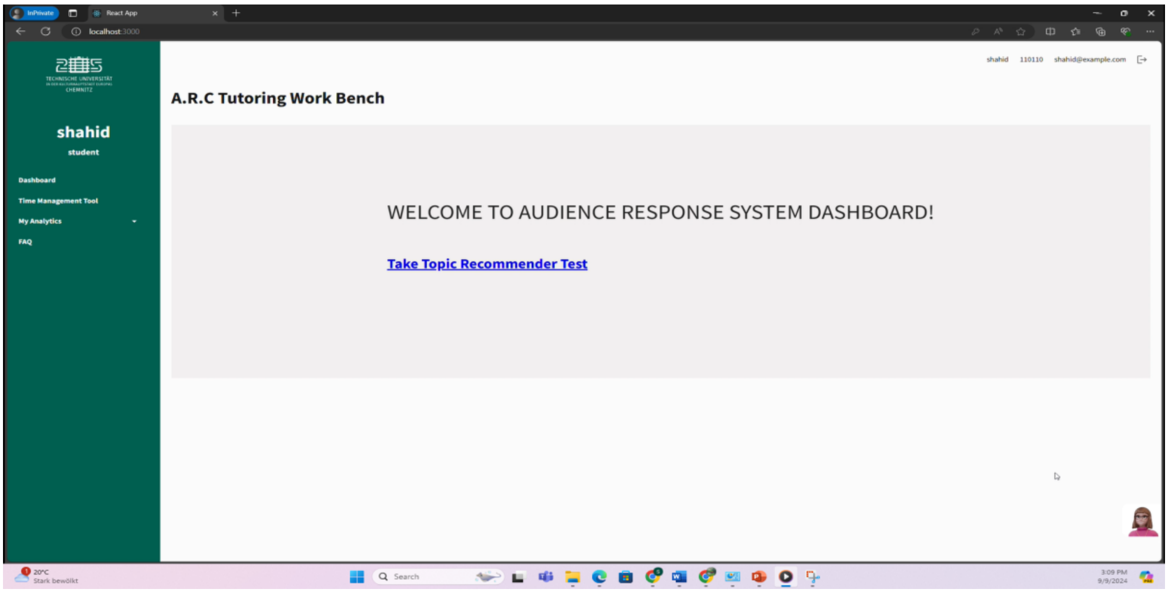


Figure 6.9: Dashboard with Generative Ai Chabot [35].

6.3 User Survey

For the thesis topic on a generative model AI based tutoring agent concept using a Google Form to take a survey [39] and additional data on the survey results can be found in Appendix A. The present chatbot was designed to work as a tutoring agent and the given survey aims at its assessment. The questions are designed to cover issues like user satisfaction, efficiency, reliability, convenience, and specific operability which are crucial factors to determine the efficiency of educational chatbots. It is common for similar surveys to be used in other research [40] with slight differences in certain questions to better fit the subject of a study, although its major purposes are to gauge users' experience, satisfaction, and learning achievements. This approach is quite useful in determining the extent of usage of the chatbot in fulfilling educational needs as well as in finding out areas of improvement to develop the abilities of the chatbot. The goal of the survey was to get feedback on different parts of chatbot like Accuracy, Response time, Usability and Emotional relevance (see Table. 6.2).

Table 6.2: Summary of Survey Responses.

Question	Response Summary
Overall User Experience	4.0 (average rating)
Response Speed	4.5 (average rating)
Ease of Navigation	4.4 (average rating)
Accuracy of Responses	3.7 (average rating)

Relevance of Answers	Most of the time (60%), Sometimes (40%)
Emotional Relevance	3.0 (average rating)
Ease of Use of Input Form	3.8 (average rating)
Satisfaction with Design	3.4 (average rating)
Encountered Bugs	No (80%), Yes (20%)

The initial (Fig. 6.9) shown survey addressed the usage of the tutoring agent. Users have also been given a scale of 1 to 5 to give their feedback of the event. These findings indicated that respondent rated the experience of the study highly with almost a quarter of them giving it a score of 4 or 5. Likewise, the rate the response speed was established to be highly appreciated since all the respondents chose 4 or 5, showing that the agent had acted quickly and effectively.

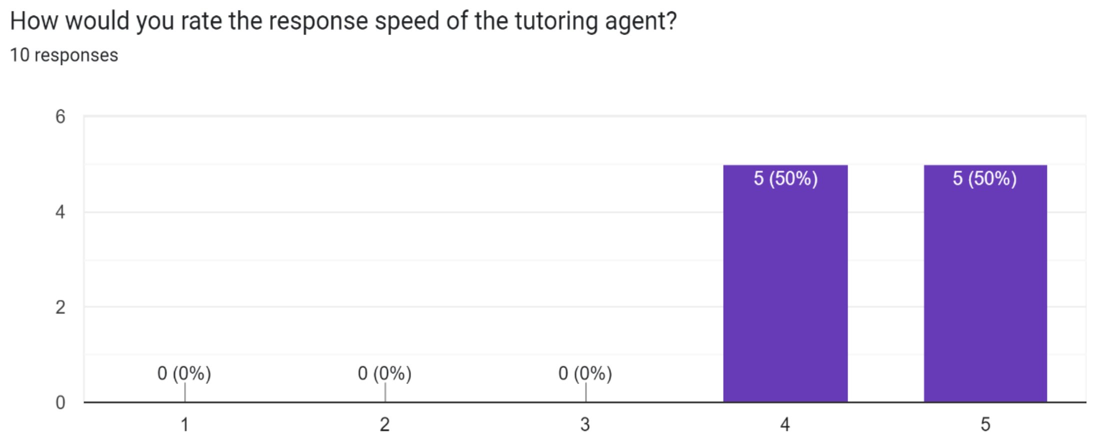


Figure 6.10: Bar chart response speed of tutoring agent.

The (Fig. 6.10) performance and accuracy of the tutoring agent were also evaluated. Half of the respondents found the responses to be highly accurate, rating them a 5, while the other half rated them a 4 or 3. Regarding the relevance of the responses to the questions asked, 40% of respondents indicated that the agent provided relevant answers most of the time, while 60% felt it did so sometimes or rarely. The emotional relevance of the responses was rated more variably, with 30% rating it a 5, 20% a 4, and the remaining 50% giving it a 3 rating.

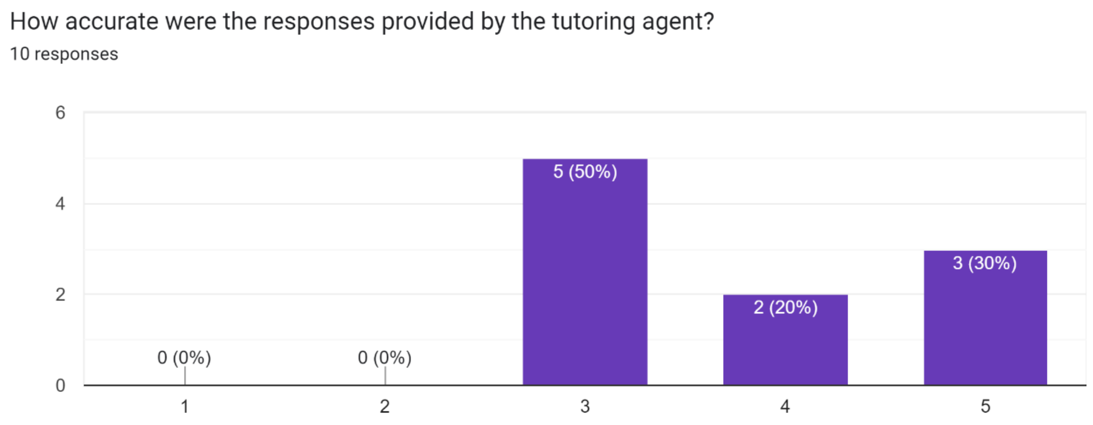


Figure 6.11: Bar chart of Rate the response Speed of Agent.

The pie chart (Fig. 6.11) shows that 40% of respondents found the tutoring agent always provided relevant answers, 30% said the answers were relevant most of the time, and 30% felt they were relevant sometimes. No one indicated that the answers were rarely or never relevant, indicating generally positive feedback with some room for improvement.

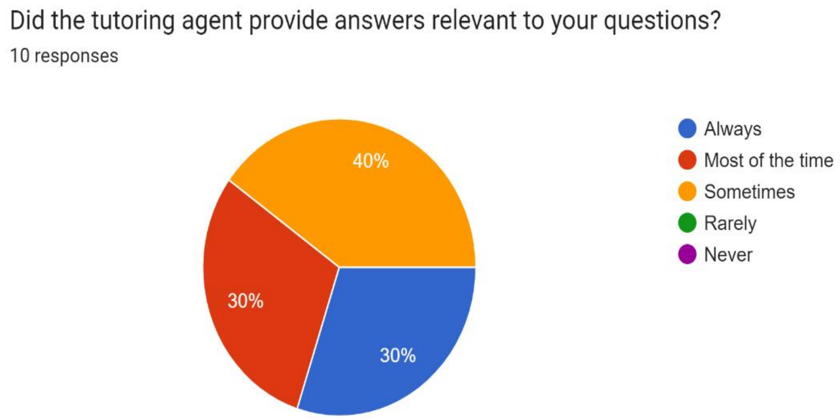


Figure 6.12: Pie chart of tutoring agent provide answers relevant to questions.

The (Fig. 6.12) bar chart reveals that among 10 responses, 6 rated the overall user experience of the tutoring agent as good to excellent which has a rating of 4 and 5. Three of the respondents (30%) assigned a neutral working (3) to the statement while one respondent (10%) gave a somewhat unsatisfactory working (2). The rating table did not reveal any very low ratings (1), meaning that the reviews were generally positive with a small area for development.

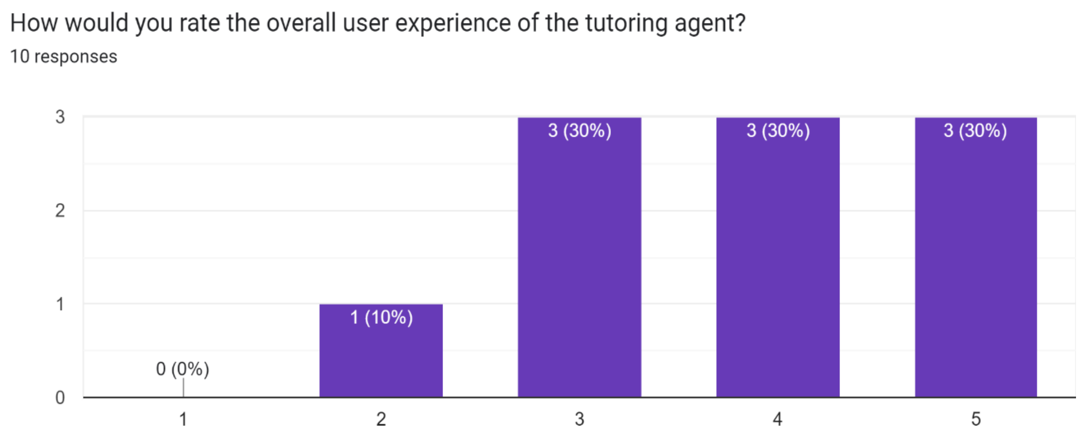


Figure 6.13: Bar chart of overall user experience of the tutoring agent.

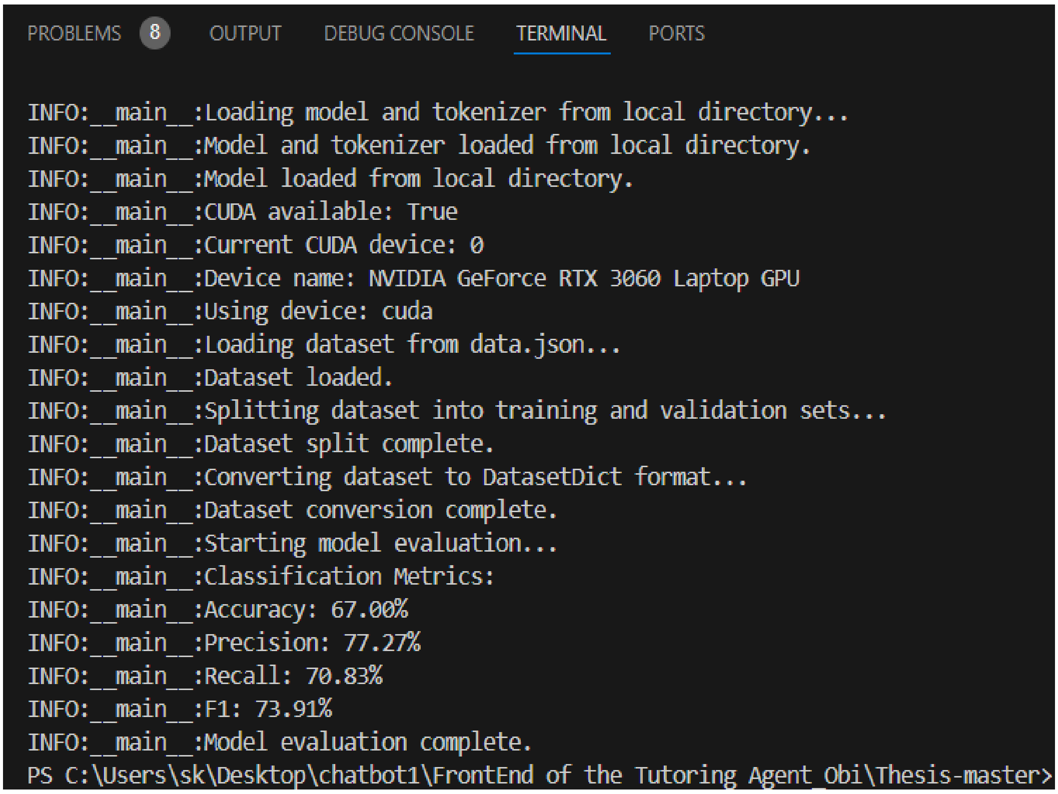
Therefore, from the results of the survey, it can be concluded that the tutoring agent is rather effective for the most difficulties, based on the findings of the survey, it can be proposed that the tutoring agent appears to be efficient in offering appropriate responses to the common questions posed by users. More specifically, 40% of the participants believed the tutoring agent responded with relevant information all the time, while 30% of the participants responded saying that the information provided was relevant most of the times. Also, regarding relevance, 30% of users said they made the answers relevant some of the time. Another indicator that attests to the efficacy of the agent is that none of the respondents indicated that the answers were rarely or never relevant. From these outcomes, it can be inferred that the tutoring agent is efficient in dealing with users' queries, even though there is a margin for an agent to always be relevant.

6.4 Evaluation

The tutoring agent model performance was measured by some of the main metrics like accuracy, precision, recall and F1 score [41]. These metrics are meaningful in the sense that they give us an overview of how well our model is doing at predicting what we want it to predict. Furthermore, a confusion matrix was plotted for detailed understanding of classification performance by the model.

6.4.1 Performance Metrics

The evaluation process begins, during which several key metrics are calculated to assess the model's performance as shown in (Fig. 6.13). The terminal output shows the evaluation of the fine-tuned LaMini-Flan-T5-248M model. The process begins with loading the model and tokenizer from the local directory, confirming that CUDA is available for GPU acceleration, specifically on an NVIDIA GeForce RTX 3060 Laptop GPU. The dataset is loaded from data. json, split into training and validation sets, and converted into the DatasetDict format. The evaluation metrics are then calculated, revealing an accuracy of 67.00%, precision of 77.27%, recall of 70.83%, and an F1 Score of 73.91%. These metrics summarize the model's performance, indicating its effectiveness and areas for improvement.



```
PROBLEMS 8   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

INFO:__main__:Loading model and tokenizer from local directory...
INFO:__main__:Model and tokenizer loaded from local directory.
INFO:__main__:Model loaded from local directory.
INFO:__main__:CUDA available: True
INFO:__main__:Current CUDA device: 0
INFO:__main__:Device name: NVIDIA GeForce RTX 3060 Laptop GPU
INFO:__main__:Using device: cuda
INFO:__main__:Loading dataset from data.json...
INFO:__main__:Dataset loaded.
INFO:__main__:Splitting dataset into training and validation sets...
INFO:__main__:Dataset split complete.
INFO:__main__:Converting dataset to DatasetDict format...
INFO:__main__:Dataset conversion complete.
INFO:__main__:Starting model evaluation...
INFO:__main__:Classification Metrics:
INFO:__main__:Accuracy: 67.00%
INFO:__main__:Precision: 77.27%
INFO:__main__:Recall: 70.83%
INFO:__main__:F1: 73.91%
INFO:__main__:Model evaluation complete.
PS C:\Users\sk\Desktop\chatbot1\FrontEnd of the Tutoring Agent_Obi\Thesis-master>
```

Figure 6.14: Evaluation of Model Performance Metrics [35].

The (Table 6.3) shows the performance of the LaMini-Flan-T5-248M model. The accuracy is 67. 00% which is the percentage of correct predictions out of all the instances. Precision is 77. 27%, this is the percentage of true positive cases out of all the positive cases that were predicted. Recall is 70. 83%, which indicates the proportion of correctly identified positive observations out of all actual positive ones. The F1 Score that considers is 73. 91%.

Table 6.3: Performance Metrics of model LaMini-Flan-T5-248M.

Metric	**Value**
Accuracy	67.00%
Precision	77.27%
Recall	70.83%
F1 Score	73.91%

- **Accuracy:** The model achieved an accuracy of 67.00%p. This metric shows that the model was right in two out of three predictions that were made by the model.
- **Precision:** The precision score is 77.27%. Accuracy is defined as the number of correct positive predictions divided by the total number of positive predictions. This means that among all the instances that the model classified as positive, 77.27% were correct.
- **Recall:** The recall score is 70.83%. Recall measures the proportion of the actual positives that have been correctly predicted by the model out of all the actual positive values in the data set.
- **F1 Score:** The F1 Score is 73.91%. The F1 Score is the measure of the average of the precision and the recall, which gives a single value that considers both the precision and the recall.

The (Fig. 6.14) bar graph shows the performance metrics of the LaMini-Flan-T5-248M model, with four key criteria: Mentioned in Table 6.2 are the above, Accuracy 67.00%, Precision 77.27%, Recall 70.83%, and F1 Score 73.91%. Each of those is depicted in a different coloured bar; thus, Accuracy is depicted in blue, Precision in green, Recall in red, and F1 Score in purple which emphasizes model proficiency in these aspects.

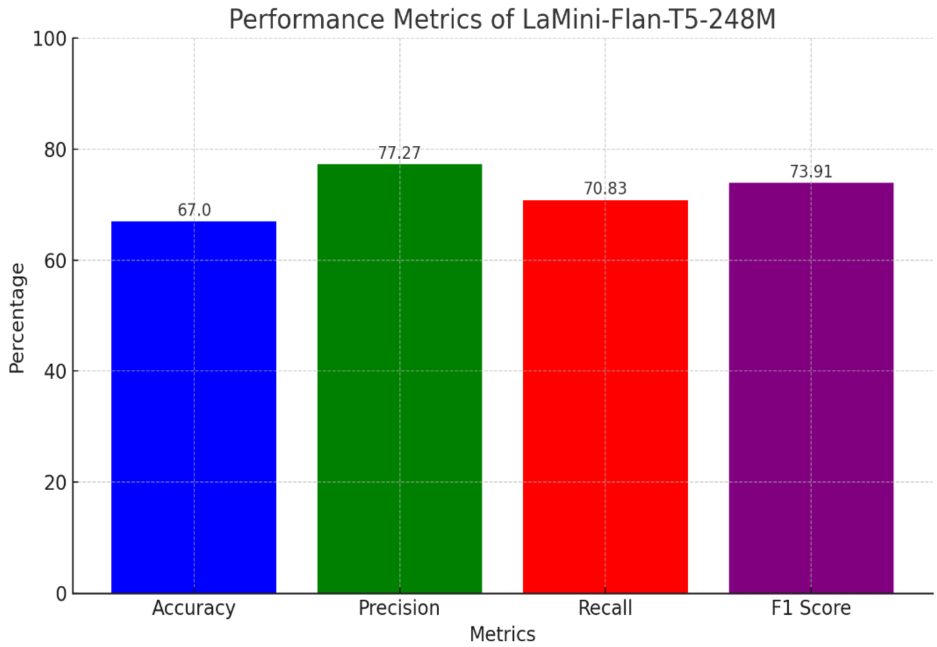


Figure 6.15: Performance Metrics Bar Chart.

6.4.2 Confusion Metrics

In this results section, I discuss an assessment of the LaMini-Flan-T5-248M model through the confusion matrix. This matrix offers a credibility of the designed model on a validation set consisting of 750 QA pairs obtained from total 7500 QA pairs. The dataset was split into 90%:10% for the training data with 6750 pairs, and the validation with 750 pairs. In fact, confusion matrix is quite helpful in seeing the distribution of the classifier's results in relative to the TP, FN, FP, TN. Here are the calculated values based on the provided data:

- True Positives (TP): 350.57 (approximately 351)
- True Negatives (TN): 151.93 (approximately 152)
- False Positives (FP): 103.12 (approximately 103)
- False Negatives (FN): 144.38 (approximately 144)

The Table 6.4 illustrates the model's results for the validation set as well as the LaMini-Flan-T5-248M on the test dataset and presents the outcomes of the model predictions. It is divided into four key components: All of these make use of the four main metrics; true positives (TP), true negatives (TN) false positives (FP), and false negatives (FN). The TPP, which is 351, stand for the true positive that is the actual positive in the considered data. There are truly negative cases with a number of 152, which symbolizes the number of cases that were accurately diagnosed as negative. The model also generated 103 false positive (FP), this means the model predicted positive when the instance is actually negative. Similarly, false negative (FN) were 144, which reveals the instances that the model itself misjudged as negative all though, they were positive. It unveils the two significant aspects of the model's performance by assessing its ability to identify true cases as well as demonstrating where it is less effecting, miscategorising some instances.

Table 6.4: Confusion Matrix.

	Predicted Positive	**Predicted Negative**
Actual Positive	TP = 351	FN = 144
Actual Negative	FP = 103	TN = 152

The (Fig. 6.15) shows the heatmap gives an overview of the confusion matrix for the Tutoring Agent when it was used to predict the validation set. A heatmap is obtained regarding the output of the model which is further divided into four quadrants. The true positive (TP) was represented on the top-left quadrant indicating that 351 samples were classified right in the positive category. This part of the graph represents the darkest coloration, which demonstrates that the model was most correct in predicting the positives. The last quadrant is (FN), 144 were misclassified as negative outcomes when in actual it had positive outcomes. This lighter area reveals the model's issue in correctly classifying specific positive cases. The bottom left shows the false positive which is mistakenly classified as positive with 103 instances. This lighter shade area proposes the misclassifications of negative values as positive ones. Last, the lowest right quadrant is false negatives (FN), which is at 152 indicating that they were correctly classified as negative. This section, which is a bit lighter, demonstrates the correctly predicted negative cases. On the whole, the heatmap is informative in showing that the model performs well in true positives and at the same time, show the points that it wrongly classified instances, hence helpful in the evaluation of general classification.

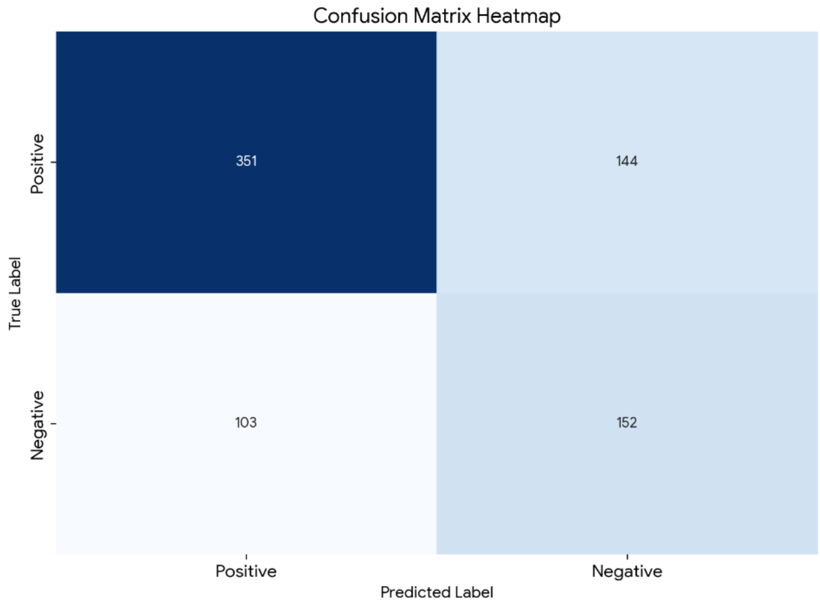


Figure 6.16: Confusion Metrics Heatmap.

In summary our results are mostly consistent with previous discoveries specifically, a recall rate of 70.83% means that our model was not able to discern all the relevant positive instances and can still be better at identifying them during tests in future. While this is a bit less than some other top-tier models out there (which typically display recall rates more around the 80-85% mark), it means that poor performance on any one dataset can be unmasked faster. The reason may be due to the specifics of our dataset and having limited computational resources. This thesis provides some highlights especially applying LaMini-Flan-T5-248M model developing Intelligent Tutoring Agent. This stands in sharp contrast to most token classification-studies, which rather focus on broad NLP models and wider-range applications. Our confusion matrix provided comprehensive analysis of the relative strengths and weaknesses by class, informing how optimization development can be approached going forward. In addition, the user feedback we brought up through surveys also adds a different image to our evaluation. Overall, the results of this study are in line with prior work looking at relational judgment using AI and suggest both opportunities to further improve performance through increased data size or manipulation techniques [42]. The results provide a comprehensive evaluation of the generative AI tutoring agent, focusing on its visual representation, user feedback, and performance metrics. The agent features an emotional avatar designed to enhance user interaction by displaying appropriate emotional responses based on user inputs. User feedback, collected through a detailed survey, highlighted the agent's strengths and areas for improvement. Overall, users found the agent effective in providing accurate and relevant responses, easy to use, and responsive [43].

7 Conclusion

This thesis presents a comprehensive investigation into the development of an intelligent tutoring agent, a sophisticated educational tool designed to revolutionize personalized learning experiences. The foundation of this innovative tool lies in the integration of cutting-edge artificial intelligence (AI) technology, specifically generative language models, exemplified by LaMini-Flan-T5-248M. This model enables the agent to comprehend and respond to student inquiries in a natural and intuitive manner, akin to a personal tutor who can adapt to individual learning needs.

The agent's backend, meticulously constructed using Python and the Flask framework, ensures a seamless and efficient exchange of information between the frontend interface and the underlying AI model. This robust infrastructure not only facilitates smooth communication but also enables the system to handle a large volume of student interactions simultaneously. The inclusion of a database further enhances the agent's capabilities by storing user interactions and model responses, allowing for data persistence and detailed performance analysis.

A distinguishing feature of this intelligent tutoring agent is its emotional avatar, designed to foster a more engaging and supportive learning environment. By utilizing natural language processing (NLP), the avatar can interpret the emotional tone of student input and respond accordingly, creating a more empathetic and personalized interaction. This innovative approach aims to enhance student engagement and motivation, ultimately leading to improved learning outcomes.

The evaluation of the agent's performance, based on both user feedback and comprehensive metrics, paints a promising picture of its potential impact. While there remains room for refinement in terms of the model's recall rate, the overwhelmingly positive feedback from users highlights its effectiveness, user-friendliness, and responsiveness. This resounding endorsement from students underscores the transformative potential of generative AI in education, offering a glimpse into a future where personalized, adaptive learning is readily accessible.

The development process of the agent involved a meticulous integration of software development and machine learning expertise. The selection and fine-tuning of the LaMini-Flan-T5-248M model, facilitated by the Hugging Face Transformers library, laid the groundwork for the agent's sophisticated language processing capabilities. Additionally, the creation of an intuitive graphical user interface (GUI) using HTML, SCSS, Bootstrap, TypeScript, and Angular ensured a seamless and user-friendly interaction experience.

Incorporating an emotion classification model further enhanced the agent's ability to understand and respond to students' emotional states, fostering a more supportive and personalized learning environment. Rigorous testing procedures, including unit testing, integration testing, and performance testing, were implemented to guarantee the agent's reliability, efficiency, and overall quality.

The implications of this research extend beyond the development of a single tutoring agent. By demonstrating the feasibility and effectiveness of generative AI in education, this thesis paves the way for a new era of intelligent tutoring systems. These systems have the potential to address the limitations of traditional teaching methods, such as the inability to provide personalized attention to each student. Furthermore, by offering adaptive learning experiences, real-time feedback, and emotional support, these systems can cater to diverse learning styles and enhance student engagement and motivation.

In conclusion, this thesis represents a significant contribution to the field of educational technology. The development of the intelligent tutoring agent, with its sophisticated AI capabilities, user-friendly interface, and personalized approach, demonstrates the transformative potential of generative AI in education. As this technology continues to evolve, we can anticipate a future where AI-powered educational tools become an integral part of the learning landscape, empowering students and educators alike. By fostering a more personalized, engaging, and effective learning experience, these tools have the potential to unlock the full potential of every learner, regardless of their background or abilities. The journey towards this future begins with the innovative research presented in this thesis, laying the groundwork for a new era of educational excellence.

References

[1] M. Abuazizeh, K. Yordanova and T. Kirste, “Affect-Aware Conversational Agent for Intelligent Tutoring of Students in Nursing Subjects,” *International Journal of Artificial Intelligence in Education,* pp. 147-171, 2023.

[2] N. Goksel and A. Bozkurt, “Artificial Intelligence in Education,” 2019, pp. 224-236.

[3] R. Baker and G. Siemens, “Educational Data Mining and Learning Analytics,” Cambridge University Press, 2014, pp. 253-272.

[4] R. Mohanan, C. Stringfellow and D. Gupta, “An emotionally intelligent tutoring system,” 2017.

[5] K. VanLEHN, “The Relative Effectiveness of Human Tutoring, Intelligent Tutoring Systems, and Other Tutoring Systems,” *Educational Psychologist,* vol. 46, no. 4, pp. 197-221, 10 2011.

[6] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever and D. Amodei, “Language Models are Few-Shot Learners,” 5 2020.

[7] O. Tapalova and N. Zhiyenbayeva, “Artificial Intelligence in Education: AIEd for Personalised Learning Pathways,” *Electronic Journal of e-Learning,* vol. 20, no. 5, pp. 639-653, 12 2022.

[8] N. Matsuda, W. W. Cohen and K. R. Koedinger, “Teaching the Teacher: Tutoring SimStudent Leads to More Effective Cognitive Tutor Authoring,” *International Journal of Artificial Intelligence in Education,* vol. 25, no. 1, pp. 1-34, 3 2015.

[9] A. E. S. Marde Helbig, Women in Computational Intelligence: Key Advances and Perspectives on Emerging Topics, A. E. Smith, Ed., Cham: Springer International Publishing, 2022.

[10] R. Binns, M. Van Kleek, M. Veale, U. Lyngs, J. Zhao and N. Shadbolt, “'It's Reducing a Human Being to a Percentage',” New York, NY, USA, 2018.

[11] N. D'Ippolito, N. Rodríguez and S. Sardina, “Fully Observable Non-deterministic Planning as Assumption-Based Reactive Synthesis,” *Journal of Artificial Intelligence Research,* vol. 61, pp. 593-621, 3 2018.

[12] A. W. C. Z. M. A.-M. A. F. A. Minghao Wu, “LaMini-Flan-T5-248M Model,” April 2023. [Online]. Available: https://huggingface.co/MBZUAI/LaMini-Flan-T5-248M.

[13] G. Jezic, J. Chen-Burger, M. Kusek, R. Sperka, R. J. Howlett and L. C. Jain, Eds., Agents and Multi-agent Systems: Technologies and Applications 2023, vol. 354, Singapore: Springer Nature Singapore, 2023.

[14] M. Murtaza, Y. Ahmed, J. A. Shamsi, F. Sherwani and M. Usman, “AI-Based Personalized E-Learning Systems: Issues, Challenges, and Solutions,” 2022. [Online].

[15] L.-K. Lee, Y.-C. Fung, Y.-W. Pun, K.-K. Wong, M. T.-Y. Yu and N.-I. Wu, “Using a Multiplatform Chatbot as an Online Tutor in a University Course,” 8 2020. [Online].

[16] M. Koivisto, “Tutoring Postgraduate Students with an AI-Based Chatbot,” 3 2023. [Online].

[17] R. Singh, “MACHINE LEARNING CHATBOT FOR EDUCATION SEARCH PURPOSE IN DUBLIN,” 2020.

[18] M. Wölfel, M. B. Shirzad, A. Reich and K. Anderer, “Knowledge-Based and Generative-AI-Driven Pedagogical Conversational Agents: A Comparative Study of Grice’s Cooperative Principles and Trust,” *Big Data and Cognitive Computing,* vol. 8, no. 1, p. 2, 12 2023.

[19] R. Makharia, Y. C. Kim, S. Bin Jo, M. A. Kim, A. Jain, P. Agarwal, A. Srivastava, A. V. Agarwal and P. Agarwal, “AI Tutor Enhanced with Prompt Engineering and Deep Knowledge Tracing,” 3 2024. [Online].

[20] S. Gupta, R. R. Dharamshi and V. Kakde, “An Impactful and Revolutionized Educational Ecosystem using Generative AI to Assist and Assess the Teaching and Learning benefits, Fostering the Post-Pandemic Requirements,” 2 2024. [Online].

[21] F. B. Emdad, B. Ravuri, L. Ayinde and M. I. Rahman, “"ChatGPT, a Friend or Foe for Education?" Analyzing the User’s Perspectives on The Latest AI Chatbot Via Reddit,” 3 2024. [Online].

[22] U. U. Shegupta, G. V. Gowthaman Malarvizhi, M. Jose, T. S. Kumar and W. Hardt, “INTERACTIVE TUTORING AGENT GENERATION FOR STUDENT ASSISTANCE IN HIGHER EDUCATION,” 2022.

[23] U. U. Shegupta, G. V. Gowthaman Malarvizhi, M. Jose, T. S. Kumar and W. Hardt, “INTERACTIVE TUTORING AGENT GENERATION FOR STUDENT ASSISTANCE IN HIGHER EDUCATION,” 7 2022. [Online].

[24] N. Khediri, M. Ben Ammar and M. Kherallah, “A Real-time Multimodal Intelligent Tutoring Emotion Recognition System (MITERS),” 12 2023. [Online].

[25] N. Singh, V. K. Gunjan and M. M. Nasralla, “A Parametrized Comparative Analysis of Performance Between Proposed Adaptive and Personalized Tutoring System “Seis Tutor” With Existing Online Tutoring System,” 2022. [Online].

[26] N. Petherbridge, “RiveScript,” [Online]. Available: https://www.rivescript.com/. [Accessed 16 07 2024].

[27] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer and V. Stoyanov, “RoBERTa: A Robustly Optimized BERT Pretraining Approach,” 7 2019.

[28] OpenAI, “Better language models and their implications,” 2023. [Online]. Available: https://openai.com/blog/better-language-models/. [Accessed 03 07 2024].

[29] Meta, “Llama2 - Hugging Face,” 2024. [Online]. Available: https://huggingface.co/meta-llama/llama-2-7b.

[30] L. Team, “LaMini-Flan-T5-248M - Hugging Face,” 2024. [Online]. Available: https://huggingface.co/LaMini-Flan-T5-248M. [Accessed 2024].

[31] B. Ajibola, “Dialogue Flow Generation for Two Comparable Models of a Tutoring Bot,” Unpublished manuscript, Automotive Software Engineering, Technische Universität Chemnitz, Chemnitz, Germany, 2024.

[32] T. D. Science, “Data Augmentation Techniques for NLP,” 2024. [Online]. Available: https://towardsdatascience.com/data-augmentation-techniques-for-nlp-56cf575174e4. [Accessed 2024].

[33] S. D. Obi, “Development of a Frontend for Agents in a Virtual Tutoring System,” Unpublished thesis, Automotive Software Engineering, Technische Universität Chemnitz, Chemnitz, Germany, 2023.

[34] J. Hartmann, “Emotion English DistilRoBERTa Base Model,” 2024. [Online]. Available: https://huggingface.co/j-hartmann/emotion-english-distilroberta-base, https://github.com/j-hartmann/emotion-english-distilroberta-base. [Accessed 07 07 2024].

[35] C. S. Akula, “ Integration of Learning Analytics into the ARC-Tutoring Workbench,” Unpublished manuscript, Chemnitz, 2024.

[36] Y. Windiatmoko, A. F. Hidayatullah, D. H. Fudholi and R. Rahmadi, “Mi-Botway: a Deep Learning-based Intelligent University Enquiries Chatbot,” 6 2021. [Online].

[37] M. Team, “Mixtral: A Versatile Model for AI Applications,” 2024. [Online]. Available: https://example.com/mixtral. [Accessed 06 July 2024].

[38] W. Chango, R. Cerezo, M. Sanchez-Santillan, R. Azevedo and C. Romero, “Improving prediction of students’ performance in intelligent tutoring systems using attribute selection and ensembles of different multimodal data sources,” *Journal of Computing in Higher Education,* vol. 33, no. 3, pp. 614-634, 12 2021.

[39] Google, “Survey on Intelligent Tutoring Systems for Technical University Students,” Google Form, 2024. [Online]. Available: https://docs.google.com/forms/d/e/1FAIpQLSc03lJUZ4r1ocy3uudxHacz-L42WhCUvC_wqD0dmjdOxnH3OA/viewform. [Accessed 2024 07 01].

[40] N. Sandu and E. Gide, “Adoption of AI-Chatbots to Enhance Student Learning Experience in Higher Education in India,” 2019.

[41] M. Vakili, M. Ghamsari and M. Rezaei, “Performance Analysis and Comparison of Machine and Deep Learning Algorithms for IoT Data Classification,” 2020.

[42] M. Muntean and F.-D. Militaru, “Metrics for Evaluating Classification Algorithms,” 2023, pp. 307-317.

[43] G. Naidu, T. Zuva and E. M. Sibanda, “A Review of Evaluation Metrics in Machine Learning Algorithms,” 2023, pp. 15-25.

Appendix A: AI Tutoring Agent User Survey Analysis

A.1 Introduction

This appendix describes the analysis of the user's questionnaire designed to measure the effectiveness of an AI-based tutoring agent. These aspects include user satisfaction, efficiency, reliability as well as overall user experience of the facility as measured by an administrated survey. Therefore, the objective of this study is to draw insight on the positives of the tutoring agent from the responses and deduce on aspects that requires changes to facilitate the tutoring agent to perform better in the capacity of education.

A.2 Survey Design and Methodology

To gather the data, the survey was planned thoroughly in Google Forms and the participants were those who used the tutoring agent in the past one month. It comprised of factual questions in the form of quizzes and rating questions Likert scale to avoid having biased feedback. Thus, the target group was reached through an online survey that reached people of different ages and with different levels of education. The response time was assessed to decide how the tutoring agent was capable of instantly responding to the user's queries, which is significant for the effectiveness of the tutoring or teaching–learning process. The quality of responses matters when it comes to the credibility and relevance of the tutoring agent, and this was established to determine how correct and related users found the responses of the agent. All the answers and the given information were evaluated through this criterion, to measure the extent to which the tutoring agent was able to identify the main concerns of the users and provide answers that are pertinent to them. Elaborateness: the extent to which the tutoring agent can comprehend the users' emotional state and provide empathetic, appropriate responses. This aspect is especially relevant to educational tools, as users' motivation influences their general engagement with the application. The aspect of learnability, which included the interface design, the intensity of the user interface and input forms, helps to determine how self-explanatory the web application was. Also, one question was designed to be open-ended, and this was done since the study also aimed at getting the users' qualitative data; the users were asked what they thought would offer improvement or add value to the tutoring agent. This is beneficial for future development or improvements as it includes the clients' and users' feedback and suggestions.

A.3 Survey Questions

These questions were developed to focus on key areas of the tutoring agent abilities. The main attention was paid to the experience of users, the reaction rate, accuracy, relevance and emotional appeal as well as convenience. Also, an open-ended question was posed to offer ideas for future revisions of the questionnaires.

Table A.1: Survey Questions and Response Summaries.

Question Text	Response Type	Key Insights
How would you rate the overall user experience?	scale (1-5)	Average rating: 3.6.
How would you rate the response speed?	scale (1-5)	Most rate 4 or 5, indicating high satisfaction.
How easy was it to navigate the web application?	scale (1-5)	Ease of navigation scored high with most ratings at 4 or 5.
How accurate were the responses provided?	scale (1-5)	50% gave the highest accuracy rating.
Did the tutoring agent provide relevant answers?	Multiple choice	40% said always, 30% most of the time.
How would you rate the emotional relevance?	scale (1-5)	Mixed responses, suggesting variability.
How would you rate the ease of use of the input form?	scale (1-5)	Generally high ease of use.
What features would you like to see added?	Open-ended	60% indicated no need for additional features.

A.4 Graphical Representations and Analysis

Each figure corresponds to specific survey questions, providing visual insights into the collected data. These graphical representations are critical for understanding the distribution of responses and identifying trends.

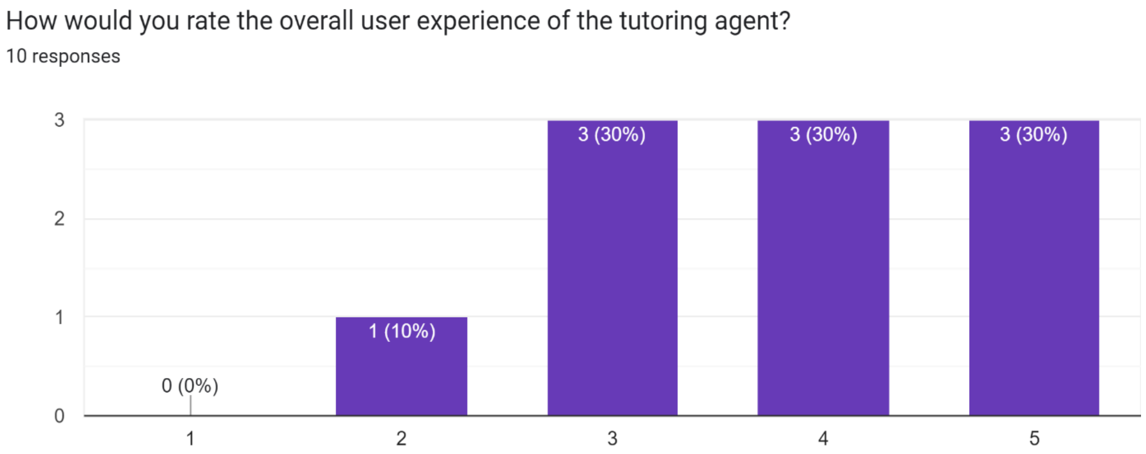


Figure A.1: Overall User Experience.

- **Description:** This bar chart shows the distribution of user experience ratings.
- **Insights:** In total, interacting with this site's overall user experience received an average rating of 3. 6 which is fairly good, as 60% of the users are expressing their fairly good satisfaction by rating it 4 or 5. It demonstrates a relative positive attitude towards the services, but it is not very satisfactory and can be increased.

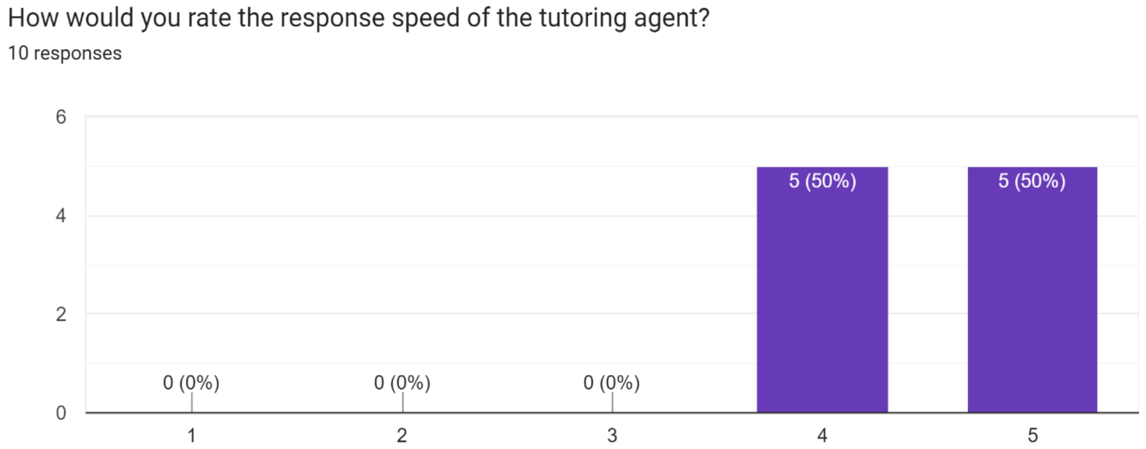


Figure A.2: Response Speed of the Tutoring Agent.

- **Description:** This bar chart illustrates how users rated the response speed of the tutoring agent.
- **Insights:** All respondents rated the response speed as either 4 or 5, demonstrating high satisfaction with the agent's efficiency. This is a strong positive indicator of the system's performance in real-time interactions.

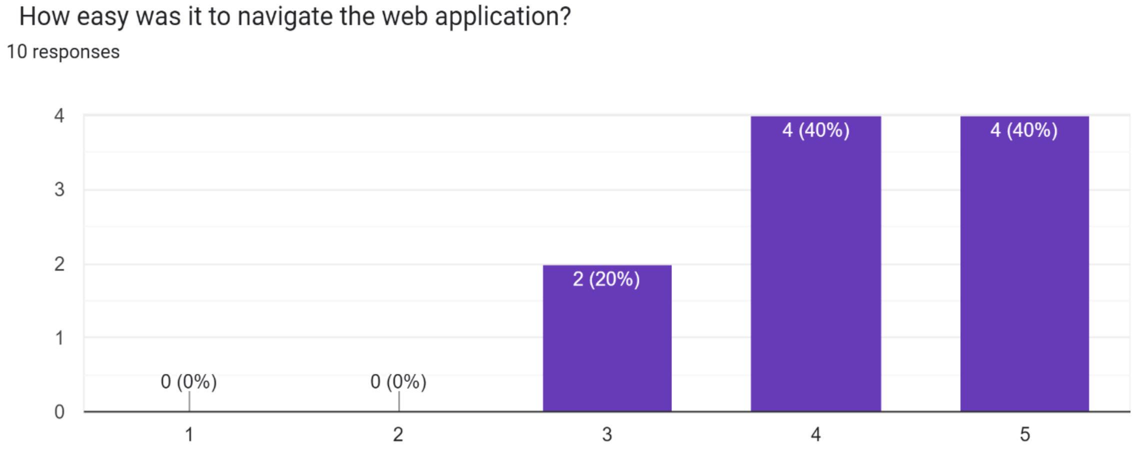


Figure A.3: Ease of Navigating the Web Application.

- **Description:** This bar chart reflects users' ratings on how easy it was to navigate the web application.
- **Insights:** With 80% of users rating the navigation ease as 4 or 5, the web interface is considered user-friendly. This suggests that the design and layout of the application are effective in providing a smooth user experience.

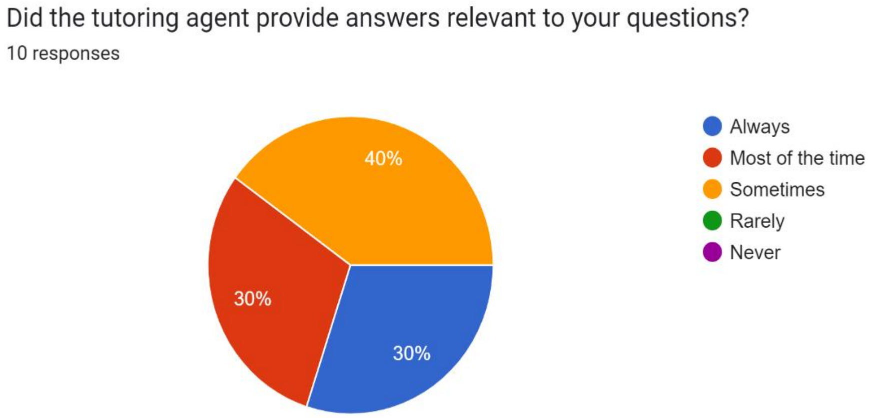


Figure A.5: Relevance of Answers Provided by the Tutoring Agent.

- **Description:** This pie chart shows the perceived relevance of the tutoring agent's answers.
- **Insights:** 40% of respondents found the answers always relevant, and 30% found them relevant most of the time. This high relevance rating suggests that the tutoring agent generally provides useful and pertinent information.

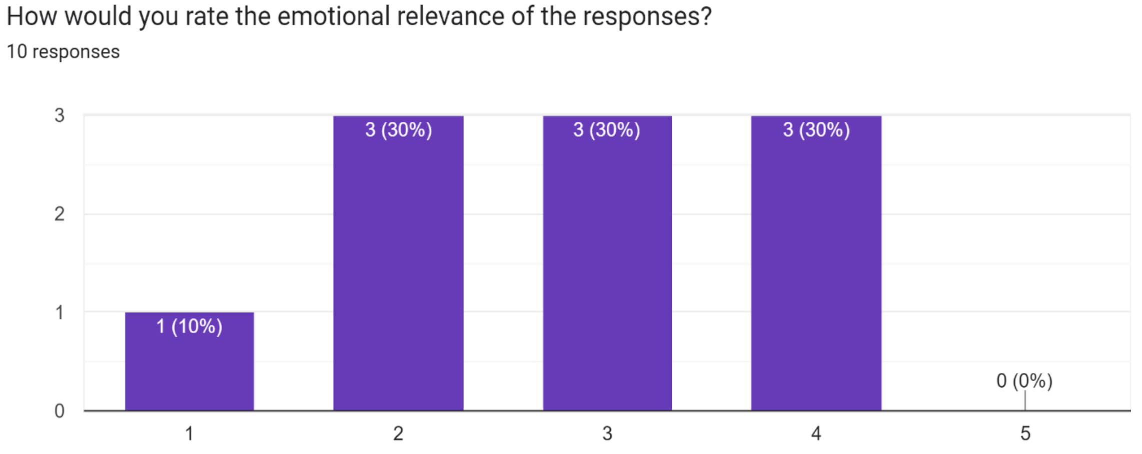


Figure A.6: Emotional Relevance of the Responses.

- **Description:** This bar chart captures user ratings of the emotional relevance of the responses.
- **Insights:** Responses to emotional relevance were mixed, with ratings spread across 2 to 4. This indicates variability in the agent's ability to connect emotionally with users, highlighting an area for development.

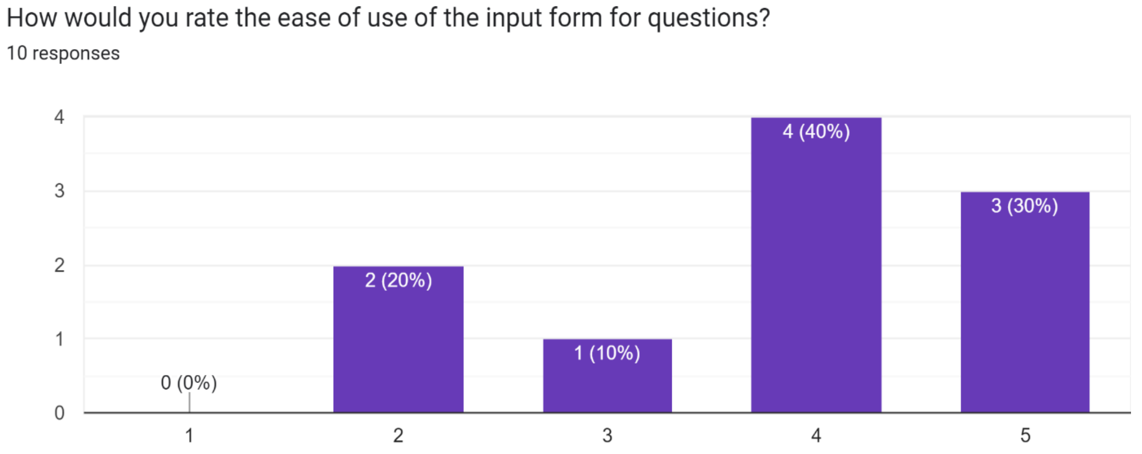


Figure A.7: Ease of Use of the Input Form for Questions.

- **Description:** This bar chart depicts how users rated the ease of using the input form for submitting questions.
- **Insights:** The majority of users found the input form easy to use, with 70% giving it a rating of 4 or 5. This suggests that the form is intuitive and accessible, facilitating user interaction.

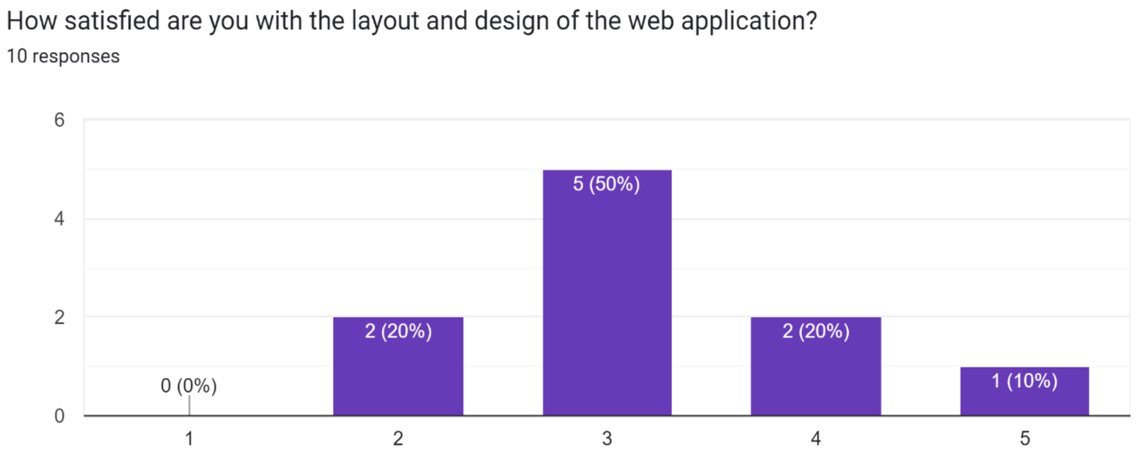


Figure A.8: Satisfaction with the Layout and Design of the Web Application.

- **Description:** This bar chart shows user satisfaction with the web application's design.
- **Insights:** Satisfaction ratings were moderate, with 50% giving a neutral rating of 3. This indicates potential for design improvements to enhance visual appeal and user engagement.

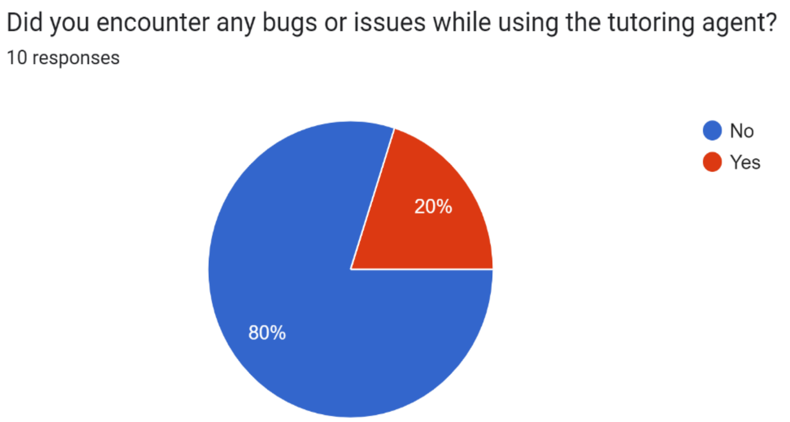


Figure A.9: Encounter with Bugs or Issues.

- **Description:** This pie chart illustrates the percentage of users who encountered bugs or issues.
- **Insights:** 80% of users did not encounter any bugs, while 20% did. This suggests a generally stable system, though addressing the reported issues could improve reliability.

Each graphical representation provides specific insights that are crucial for understanding user interactions with the tutoring agent. The detailed analyses help identify areas of strength, such as response speed and ease of use, as well as areas needing improvement, like emotional relevance and design satisfaction.

A.5 Detailed Analysis

The detailed analysis of survey data reveals several key findings:

- **Usage Frequency:** The tutoring agent is not used frequently by the majority of respondents, indicating a need for strategies to increase engagement.
- **User Experience:** Overall, users have a positive experience with the agent, but there is room for improvement to reach higher satisfaction levels.
- **Response Speed:** High satisfaction with response speed suggests that the agent performs efficiently in real-time interactions.
- **Navigation and Usability:** The ease of navigation and input form usability indicates that the web application is well-designed, though minor enhancements could further improve user experience.
- **Answer Relevance:** While most users find the responses relevant, there is still a significant percentage that experiences less consistent relevance, suggesting the need for refining the agent's response algorithms.
- **Emotional Connection:** The variability in emotional relevance ratings highlights an area where the agent could be improved to provide more empathetic and contextually aware responses.
- **Design Satisfaction:** The moderate satisfaction with the web application’s design suggests that visual and functional design improvements could enhance overall user satisfaction.
- **Technical Stability:** Although the majority did not encounter bugs, the 20% who did indicate the necessity for ongoing technical maintenance and troubleshooting.

TECHNISCHE UNIVERSITÄT CHEMNITZ

Zentrales Prüfungsamt
Selbstständigkeitserklärung

Name:	Ismail	**Bitte beachten:** 1. Bitte binden Sie dieses Blatt am Ende Ihrer Arbeit ein.
Vorname:	Shahid	
geb. am:	06.12.1991	
Matr.-Nr.:	579322	

Selbstständigkeitserklärung*

Ich erkläre gegenüber der Technischen Universität Chemnitz, dass ich die vorliegende Masterarbeit selbstständig und ohne Benutzung anderer als der angegebenen Quellen und Hilfsmittel angefertigt habe.

Die vorliegende Arbeit ist frei von Plagiaten. Alle Ausführungen, die wörtlich oder inhaltlich aus anderen Schriften entnommen sind, habe ich als solche kenntlich gemacht.

Diese Arbeit wurde in gleicher oder ähnlicher Form noch nicht als Prüfungsleistung eingereicht und ist auch noch nicht veröffentlicht.

Datum: 26.07.2024 Unterschrift: ....

* Statement of Authorship

I hereby certify to the Technische Universität Chemnitz that this thesis is all my own work and uses no external material other than that acknowledged in the text.

This work contains no plagiarism and all sentences or passages directly quoted from other people's work or including content derived from such work have been specifically credited to the authors and sources.

This paper has neither been submitted in the same or a similar form to any other examiner nor for the award of any other degree, nor has it previously been published.

Chemnitzer Informatik-Berichte

In der Reihe der Chemnitzer Informatik-Berichte sind folgende Berichte erschienen:

CSR-22-01	Henrik Zant, Reda Harradi, Wolfram Hardt, Expert System-based Embedded Software Module and Ruleset for Adaptive Flight Missions, September 2022, Chemnitz
CSR-23-01	Stephan Lede, René Schmidt, Wolfram Hardt, Analyse des Ressourcenverbrauchs von Deep Learning Methoden zur Einschlagslokalisierung auf eingebetteten Systemen, Januar 2023, Chemnitz
CSR-23-02	André Böhle, René Schmidt, Wolfram Hardt, Schnittstelle zur Datenakquise von Daten des Lernmanagementsystems unter Berücksichtigung bestehender Datenschutzrichtlinien, Januar 2023, Chemnitz
CSR-23-03	Falk Zaumseil, Sabrina, Bräuer, Thomas L. Milani, Guido Brunnett, Gender Dissimilarities in Body Gait Kinematics at Different Speeds, März 2023, Chemnitz
CSR-23-04	Tom Uhlmann, Sabrina Bräuer, Falk Zaumseil, Guido Brunnett, A Novel Inexpensive Camera-based Photoelectric Barrier System for Accurate Flying Sprint Time Measurement, März 2023, Chemnitz
CSR-23-05	Samer Salamah, Guido Brunnett, Sabrina Bräuer, Tom Uhlmann, Oliver Rehren, Katharina Jahn, Thomas L. Milani, Güunter Daniel Rey, NaturalWalk: An Anatomy-based Synthesizer for Human Walking Motions, März 2023, Chemnitz
CSR-24-01	Seyhmus Akaslan, Ariane Heller, Wolfram Hardt, Hardware-Supported Test Environment Analysis for CAN Message Communication, Juni 2024, Chemnitz
CSR-24-02	S. M. Rizwanur Rahman, Wolfram Hardt, Image Classification for Drone Propeller Inspection using Deep Learning, August 2024, Chemnitz
CSR-24-03	Sebastian Pettke, Wolfram Hardt, Ariane Heller, Comparison of maximum weight clique algorithms, August 2024, Chemnitz
CSR-24-04	Md Shoriful Islam, Ummay Ubaida Shegupta, Wolfram Hardt, Design and Development of a Predictive Learning Analytics System, August 2024, Chemnitz
CSR-24-05	Sopuluchukwu Divine Obi, Ummay Ubaida Shegupta, Wolfram Hardt, Development of a Frontend for Agents in a Virtual Tutoring System, August 2024, Chemnitz

Chemnitzer Informatik-Berichte

CSR-24-06 Saddaf Afrin Khan, Ummay Ubaida Shegupta, Wolfram Hardt, Design and Development of a Diagnostic Learning Analytics System, August 2024, Chemnitz

CSR-24-07 Túlio Gomes Pereira, Wolfram Hardt, Ariane Heller, Development of a Material Classification Model for Multispectral LiDAR Data, August 2024, Chemnitz

CSR-24-08 Sumanth Anugandula, Ummay Ubaida Shegupta, Wolfram Hardt, Design and Development of a Virtual Agent for Interactive Learning Scenarios, September 2024, Chemnitz

CSR-25-01 Md. Ali Awlad, Hasan Saadi Jaber Aljzaere, Wolfram Hardt, AUTOSAR Software Component for Atomic Straight Driving Patterns, März 2025, Chemnitz

CSR-25-02 Billava Vasantha Monisha, Hasan Saadi Jaber Aljzaere, Wolfram Hardt, Automotive Software Component for QT Based Car Status Visualization, März 2025, Chemnitz

CSR-25-03 Zahra Khadivi, Batbayar Battseren, Wolfram Hardt, Acoustic-Based MAV Propeller Inspection, Mai 2025, Chemnitz

CSR-25-04 Tripti Kumari Shukla, Ummay Ubaida Shegupta, Wolfram Hardt, Time Management Tool Development to Support Self-regulated Learning, August 2025, Chemnitz

CSR-25-05 Ambu Babu, Ummay Ubaida Shegupta, Wolfram Hardt, Development of a Retrieval Model based Backend of a Tutoring Agent, August 2025, Chemnitz

CSR-25-06 Shahid Ismail, Ummay Ubaida Shegupta, Wolfram Hardt, Development of a Generative Model based Backend of Tutoring Agent, August 2025, Chemnitz

Chemnitzer Informatik-Berichte

ISSN 0947-5125

Herausgeber: Fakultät für Informatik, TU Chemnitz
Straße der Nationen 62, D-09111 Chemnitz