Teaching Support Chatbot

Final Year Dissertation

Author: **Omar Issa Riyaz**
Supervisor: **Dr. Idris Skloul Ibrahim**

School of Mathematics and Computer Science

A thesis presented for the degree of
BSc Computer Science (Hons.)

Heriot-Watt University
April 15, 2024

Number of Pages: 40
Word Count: 11111

Declaration of Authorship

I, Omar Issa Riyaz, declare that this thesis titled, 'Teaching Support Chatbot' and the work presented in it is my own. I confirm that this work submitted for assessment is my own and is expressed in my own words. Any uses made within it of the works of other authors in any form (e.g. ideas, equations, figures, text, tables, programs) are properly acknowledged at any point of their use. A list of the references employed is included.

Signed: Omar Riyaz

Dated: 05/02/2023

Acknowledgement

My sincerest gratitude goes out to my supervisor, Dr. Idris Skloul Ibrahim, for all of his helpful advice, encouragement, and support during the writing of my dissertation. The direction and quality of this project have been greatly influenced by his knowledge, insightful feedback and persistent commitment. I would also like to express my gratitude to my family and friends for their encouragement, compassion, and understanding throughout this journey. Their unwavering support has served as an endless supply of inspiration. Lastly, I would want to thank all of the participants who so kindly gave of their time and insight, without which this dissertation would not have been possible.

Teaching Support Chatbot

Omar Issa Riyaz

Abstract

In the educational technology space, this dissertation sets out on an important journey to develop an innovative chatbot that can accurately respond to student questions based on the lecture slides provided to the students on Canvas. Built as a new solution, this chatbot aims to better engage students.

The ability of the chatbot to understand lecture slides in PDF format is the foundation of this project. This capability enables the chatbot to function as a personalised learning assistant for each student, which would help them understand the lecture topics and provide smart answers to their inquiries, as well as free up time for the professors teaching the course, as they would not need to answer these inquiries themselves.

By using the newest developments of Machine Learning (ML) and Natural Language Processing (NLP), this chatbot leverages Artificial Intelligence (AI) to help close the knowledge gap between traditional learning and the digital age of learning. The combination of technology and education has significant effects for the future of teaching, since it can simplify and improve the learning process for students and professors.

The chatbot will be put through a thorough testing process with real students of Heriot Watt University, in order to assess its usefulness, accuracy and practicality. The evaluation method will take into account important measures, such as user experience and inquiry resolution efficiency, with the goal of determining how well the chatbot can assist learning and how well its liked by the students.

To summarize, this dissertation is a small step towards the evolution of the field of digital learning. It dives into the integration of AI and education through the development and testing of a chatbot that can answer student queries based on lecture slides. By doing this, it paves a path towards a day when personalized learning systems will be the new normal in how students interact with their learning materials.

Table of Contents

List of Figures

List of Tables

.

Abbreviations

AI	**A**rtificial **I**ntelligence
CD	**C**ontinuous **D**eployment
CI	**C**ontinuous **I**ntegration
CUI	**C**onversational **U**ser **I**nterface
GDPR	**G**eneral **D**ata **P**rotection **R**egulation
GPT	**G**enerative **P**re-Trained **T**ransformers
GUI	**G**raphical **U**ser **I**nterface
LLM	**L**arge **L**anguage **M**odel
ML	**M**achine **L**earning
NLP	**N**atural **L**anguage **P**rocessing
NLU	**N**atural **L**anguage **U**nderstanding
PDF	**P**ortable **D**ocument **F**ormat
SDLC	**S**oftware **D**evelopment **L**ife **C**ycle
SUS	**S**ystem **U**sability **S**cale
UI	**U**ser **I**nterface

Chapter 1

Introduction

The first chapter of the dissertation provides a thorough summary of the work and acts as its introduction. Section 1.1 provides background information and introduces the project, while section 1.2 presents the problem statement. The motivation behind the project is discussed in section 1.3, followed by the aims and objectives outlined in sections 1.4 and 1.5, respectively. Finally, the chapter concludes with a summary in section 1.6.

1.1 Background

The combination of Artificial Intelligence (AI) and education has lots of potential to change how students can interact with their course materials. Primary results in [Kim et al., 2020] show that positive attitudes towards using an AI teaching assistant are positively correlated with the AI teaching assistant's perceived utility and perceived ease of communication, which in turn increases the intention to implement AI teaching assistants in education. This dissertation sets out on a journey toward the creation of a chatbot that aims to meet an important demand today's education space. The main goal of the chatbot is to use Natural Language Processing (NLP) and Machine Learning (ML) to accurately answer questions from students based on the lecture slides posted on Canvas. By doing this, it aims to both improve student engagement with learning material and reduce the workload for professors, freeing time up for them to focus on more important matters while the personalized learning chatbot responds to student issues.

According to [Garcia Brustenga, 2018], a chatbot, also known as a conversational robot, is a computer program that can communicate with humans in spoken language. Its primary purpose is to imitate the logic of human conversation. The dissertation's primary focus is on the chatbot's ability to understand lecture slides that are given in Portable Document Format (PDF), thus making it act as a personalized learning tool for every student. This helpful aide will make lecture slides easier to understand and provides thoughtful answers to any doubts, making learning more effective and stimulating. The combination of AI, education, and responsive chatbot skills holds potential for simplifying the learning experience in a time where education and technology are becoming more prevalent together.

The chatbot is an important step in bridging the knowledge gap between traditional education and the modern age of technology, since it makes use of the newest advancements in ML, Large Language Models (LLM) and NLP. In [Medeiros et al., 2023], it mentions that when LLMs are integrated, users can communicate with AI-based assistants using natural language. Chatbots can quickly assess the user's inquiry and provide clear and concise guidance for troubleshooting or relevant information.

Heriot Watt University students will participate in a thorough testing process to evaluate the effectiveness of this chatbot, by letting them experiment with the chatbot and answer questionnaires. In [Følstad and Brandtzaeg, 2020], they have shown that collecting user feedback via a questionnaire study based on the critical incident technique is both feasible and advantageous. In order to test the system's ability to enhance learning and its acceptance by students, important metrics like user experience, accuracy, query resolution efficiency, and general usefulness will be looked at. Using these metrics it can then be determined whether the chatbot could replace some of the already existing query resolution solutions already in place inside of Canvas.

This dissertation represents a step forward in the field of digital learning. It opens the door for a time when personalized learning assistance systems are the norm, changing how students engage with what they are learning. The project aims to strengthen teachers and students by overcoming the space between the traditional classroom and today's digital world.

1.2 Problem Statement

The problem at hand involves the need for a technical solution to overcome limitations in query resolution techniques, specifically when it comes to student inquiries about lecture slides. The current solution, as demonstrated by the Canvas 'discussions' page, might not be the best at processing and answering these questions. This is because students have to wait for an actual professor or teaching assistant to manually answer these questions, instead of having these queries resolved instantly. This means students do not use the discussions tab. This motivates research into a chatbot that answers student inquiries using NLP and ML, improving query resolution accuracy and speed to the extent that it could out perform the existing solution.

1.3 Motivation

The motivation for this dissertation topic comes from being aware of the difficulties in today's educational environment, such as the flaws in the current techniques for answering questions from students. Motivated by the aim of improving a student's learning experiences and relieving professors of their workload, the chatbot acts as a smart solution. This dissertation aims to bring in a new era of productivity and engagement by changing the way students engage with their university assignments and exams through the incorporation of AI into their learning tools.

1.4 Aims

The primary aim of this dissertation is to create, test, and evaluate a chatbot that utilizes ML and NLP technologies to precisely answer student inquiries based on lecture slides in PDF format. The aim is to create a personalized learning assistant that can enhance student engagement with their lecture slides, simplifies resolution of doubts, and lessen the burden on professors.

1.5 Objectives

Table 1.1 lists the goals that are expected to be completed by the project's conclusion and at this stage in order to help you understand the project's goals.

ID	Objective
OBJ-1	Create a chatbot that can effectively scan and understand PDF lecture slides in order to respond accurately to student inquiries.
OBJ-2	Implement a user friendly interface for the chatbot, ensuring accessibility and ease of interaction for students.
OBJ-3	Create an evaluation system for the chatbot that includes actual Heriot Watt University students in order to assess its accuracy, practicality, effectiveness, and user experience.

Table 1.1: Objectives of the Teaching Support Chatbot Dissertation,

1.6 Conclusion

In conclusion, this chapter lays the groundwork for studying and developing a teaching support chatbot with the goal of improving student learning and reducing professors' workloads. The chatbot aims to improve query resolution methods through using ML and NLP technologies to provide fast and precise answers to student queries based on lecture slides. This dissertation seeks to further the development of digital learning environments by exploring how AI solutions affect teachers, learners, and universities. All things considered, this chapter provides the groundwork for an investigation that closes the gap between traditional methods of teaching and modern technological advancements.

Relevant material from the technical literature is explored in the upcoming chapter to gain further insight into the background of the research. In order to place the research within the larger context of previous studies and advancements in the area, some background discussion is provided. In order to guide the approach to creating a teaching assistance chatbot, significant ideas are looked for, techniques, and technologies in the relevant literature. The advances and difficulties that have formed the present state of AI-driven educational tools are to be clarified through this exploration, as well as to demonstrate the relevance of earlier work to the goals for the project.

Chapter 2

Literature Review

This chapter presents a comprehensive analysis of the state of knowledge in the field at the moment, highlighting theories, methods, important concepts, and findings from earlier research. An overview is provided in Section 2.1, laying the groundwork for a more in-depth analysis in the parts that follow. Sections 2.2 through 2.5 examine the details of earlier studies, breaking down important ideas, approaches, and conclusions. The chapter ends in Section 2.6, where insights from the literature are summarized and concluded.

2.1 Introduction

To introduce and examine the research done on AI and its broader applicability within assisted teaching, an in-depth literature review is conducted in this chapter. The research being conducted is important for demonstrating the project's hypothesis and achieving its goals.

The introduction of chatbots powered by AI into universities is a change that offers better learning outcomes. All aspects of creating AI teaching assistants are examined in this literature review, from their development and technological design to the interactions between humans and chatbots and the difficulties encountered in real-world implementations.

2.2 Developing an AI Chatbot

The development of chatbots has become a step forward in the ever evolving field of education guided by technology, providing a connection between traditional teaching and new age online interactions. In addition to changing the world of education, these chatbots, which are made for conversing with users through human like interactions, also act as evidence of the benefits of combining education and AI.

Creating chatbots for education requires a combination of classroom concepts and modern technology. Basically, developing a useful chatbot for educational purposes is a process that starts with defining the function and goals of the chatbot. The foundation for the next stages is set by an understanding of the educational environment and the requirements of both professors and students.

2.2.1 Needs Analysis and Educational Context

The first step in creating a teaching chatbot is doing an in-depth requirements analysis. This stage involves understanding the specific needs of the learning setting it aims to assist with. The first step in changing the chatbot's functionality to meet these particular requirements is to identify the problems, difficulties, and gaps in the current educational systems. This research forms the foundation of the chatbot's development, offering real-time query resolution and personalized learning. As explained in [Bisser, 2021], the most crucial element that impacts the success of your chatbot is its overall design. As a developer, the goal is to create chatbots that people will use. As a result, you should consider how you may make using your chatbot more advantageous for users than using any other web application or other alternatives.

2.2.2 Technological Architecture and Natural Language Processing

NLP converts spoken words into data [Vinita et al., 2022]. A teaching chatbot's technical framework is an important aspect of its development. Putting in place strong NLP algorithms is key to making the chatbot capable of understanding and answering student inquiries. According to [Lazarus, 2010], people in a variety of areas of life, including business, sales, management, athletics, health, education, coaching, therapy, and all other human activities involving communication, are quickly beginning to recognize NLP as a valuable set of tools and solutions. In [Aggarwal et al., 2023], it states that NLP, ML, deep learning, reinforcement learning, and other technologies help chatbots become smarter. The chatbot's ability to understand the nuances of human language is improved through NLP, which also helps it have more engaging and contextually relevant conversations. The chatbot becomes an interactive teaching assistant when ML algorithms are integrated, giving it even more ability to change and adapt in response to user interactions.

2.2.3 User Interface and Experience Design

The User Interface (UI) and experience design are important components that influence the quality of teaching chatbots, apart from their technological features. Students and professors may easily involve the chatbot into their learning experiences because of its straightforward design and user-friendly interface. This means taking into account not just the chatbot's functionality but also the user's experience from the moment of start to the resolution of their query.

In [Smestad and Volden, 2019], they suggest using personality to enhance the chatbots user experience in an attempt to change how people view them and the possible value they can offer. The objective of this paper was to investigate how chatbot personalities affect user experience, with a special focus on practical quality, the hedonic quality, and attractiveness in relation to Chatbots A and B. The paper comes to the conclusion that chatbot user interfaces are influenced by personality. In the future, the paper will also look into the long-term effects of chatbot personalities, how to improve the personality framework for different user groups and domains, how to better suit measurement tools like AttrakDiff for Conversational User Interfaces (CUI), and how to create tools to evaluate the user experience as conversational agents evolve and become more like themselves.

2.2.4 Ethical Considerations and Continuous Improvement

In [Ali and Djalilian, 2023], it describes that as technology develops, scientists should collaborate to define precise moral standards while acknowledging the valid applications of chatbots. Ethical considerations are also critical in the field of teaching chatbots. Responsible consideration of privacy, data security, and inclusion are all part of the development process. [Ng et al., 2022] talks about how conversational chatbots use ML, therefore AI ethics apply to them as well. Integrating ethics into automated technology is a difficulty presented by AI. The chatbot is constantly improved to ensure that it remains adaptable and in line with the changing needs of its users through the use of improvement processes, which make use of user input and data analytics.

Creating teaching chatbots is really a dance between innovation in technology and teaching principles. The following sections of this literature review will explore important studies and concepts that have thrown a light on this process of development and impacted the field of AI-driven educational supports.

2.3 Impact of a virtual teaching assistant

AI has produced a disruptive wave in the field of present day education, changing the way professors perform their jobs and introducing in new players: AI teaching assistants. These machine learning and AI capabilities have allowed these digital assistants to cross traditional educational boundaries and have a lasting effect on the field of teaching and learning.

Understanding the complex effects of AI teaching assistants requires looking into several educational environments and the function these digital assistants perform. From Ghanaian higher education institutions to international viewpoints on the changing role of teachers, the literature review tells a story that goes beyond basic assistance and goes into the core of how students interact with learning resources with the help of AI chatbots.

2.3.1 Global Perspectives on AI Teaching Assistants

One aspect of the effects concerns a global study on the ways in which AI Teaching Assistants, specifically in Ghanaian higher education, impact students' educational experiences. The incorporation of research in this field reveals the ways that AI can be utilized to improve accessibility, encourage participation, close gaps in learning, and change the current state of education.

Since the students who interacted with the chatbot outperformed the students in the control group who communicated with the course teacher, the primary findings and justifications of this study support the usefulness of deploying chatbots in higher education [Essel et al., 2022]. This is particularly important for nations like Ghana, where there is a high student-teacher ratio and it might be difficult to provide pupils with quick feedback and responses.

2.3.2 Transformative Roles of Educators in the Digital Age

The second article involves considering how the use of AI, as shown by websites like as Chat GPT, is changing the nature of teaching in the current digital learning environments. The study aims at understanding how AI teaching assistants transform the essence of teaching and student-professor relationships in a time where digital interaction is dominant, in addition to assisting professors in their work. In [Network, 2023] it mentions that teachers that use Chat GPT not only demonstrate their own adaptability and resilience, but they are also preparing their students for the future. They are concurrently teaching students how to use technology ethically and effectively while keeping a close eye out for instances of cheating and plagiarism.

2.3.3 Impact on New Teachers Effectiveness

The study examines, with a specific focus, how chatbot-based classroom exercises impact the efficacy of beginner teachers. The study's results are positive despite its limitations, and they propose that the application of AI technology will provide insight into how to boost preservice teachers' teacher efficacy by inspiring them to support active learning experiences [Song et al., 2022]. By looking at studies in this area, more about how chatbots, or AI teaching assistants, help shape the confidence and educational methods of future teachers is learnt.

2.3.4 Smart Teaching Assistants for Specialized Learning

This part investigates the impact of using AI-powered chatbots as smart teaching assistants, expanding the viewpoint to first-year engineering students. The paper examines how chatbots are changing the face of education and how they affect students educational experiences. The paper discovered an increasing variety of chatbot uses in educational settings by reviewing pertinent literature. They carried out an exploratory survey to learn more about the needs and opinions of students on the use of chatbots. After that, they implemented a chatbot system called Alpha and evaluated its usability. The findings indicated that students found the new system to be incredibly user-friendly and intuitive. This feature demonstrates the adaptability and efficiency of AI teaching assistants in fulfilling the unique requirements of students in technical subjects by looking at how they work in specialized learning environments [Abdelhamid and Katz, 2020].

Uncovering the effects of AI Teaching Assistants on the learning environment is the main objective as these various aspects are covered. The goal of this literature review is to offer an in-depth understanding of the ways in which these digital teachers impact learning experiences, alter the roles that the teachers play, and support the ongoing development of education in the digital age.

2.4 Interacting with Chatbots

The use of chatbots as teaching assistants in the link between technology and education represents a new era in the way students and teachers interact with digital resources. Connecting with chatbots goes past conventional methods of teaching by providing an individualized and interactive learning experience. This section of the literature review delves into the various aspects associated with building and the impact of interactions between humans and chatbots in learning environments.

2.4.1 Systematic Review of Interactions with Educational Chatbots

The study begins with an in-depth discussion of research on user interactions with educational chatbots. Through an extensive review of the current literature, this part aims to identify themes and understandings about the efficiency and challenges related to user interaction with AI-powered teaching assistants. [Kuhail et al., 2023] states that understanding the nuances of user engagement is crucial to optimizing the educational impact of these digital teachers. This study described how several educational chatbot approaches empower learners across various domains. The study analyzed 36 educational chatbots proposed in the literature. To analyze the tools, the study assessed each chatbot within seven dimensions: educational field, platform, educational role, interaction style, design principles, empirical principles, and challenges as well as limitations.

2.4.2 Human–Chatbot Interaction Design

The way in which human-chatbot conversations are designed is essential for the effectiveness of teaching chatbots. This section summarizes the study regarding the elements of design that impact interaction quality. [Chaves and Gerosa, 2021] investigates the ideas that guide the development of purposeful and beneficial user-chatbot interactions by looking at choices made in the areas of flexibility, conversational styles, and interface design. The study highlighted that some traits—like moral agency and communicability—are strongly influenced by the domain, while others—like manners and damage control—show greater universal application. They also identified domains where social qualities were primarily studied. By analyzing the relationships between these traits, they developed 22 hypotheses from their literature review, highlighting how complex it is to create chatbots that behave in socially acceptable ways. The premises and social features address important aspects of human–human communication, even though it is acknowledged that they may not be exhaustive.

The objective as these elements are gone through is to uncover insights that guide the design principles for most effective engagement, as well as to learn the mechanics of interacting with teaching chatbots. The goal of the literature review is to present a thorough understanding of the ways in which the specifics of human-chatbot interactions enhance the educational process as a whole, creating an environment in which learning is conversational, personalized, and not just instructional.

2.5 Problems that come with an AI learning tool

The use of AI technologies for learning has opened up previously unheard-of possibilities in education in the current era of technological innovation. This new event is not without its difficulties though. The section on the problems surrounding AI learning tools goes into a detailed analysis of the drawbacks, restrictions, and ethical challenges that come with implementing these modern technologies in educational environments.

2.5.1 AI-Chatbots for Teaching Geography

Focusing in on a particular area, the study examines the issues surrounding the use of AI chatbots in the teaching of geography. [Scheider et al., 2023] looks at the specific challenges presented by the use of AI learning tools in a particular academic discipline in order to shed light on potential downsides, differences in subject requirements, and the challenges involved in using technology for teaching in a variety of domains. The analysis of ChatGPT's results in Science and Geography tasks shows that, although it can accurately respond to questions in a wide range of ability categories and educational levels, it has limitations. Problems with clarity, completeness, and conciseness point to difficulties in customizing solutions to satisfy complex academic standards. These results point to the difficulties in providing thorough and customized support in educational environments, indicating that further development is required to maximize the chatbot's total educational value.

In [Tamayo, 2020] it mentions that even if robots could be programmed to teach a curriculum, it wouldn't make them successful teachers. It becomes apparent as these articles have been covered that there are quite a few obstacles to overcome when incorporating AI learning technologies in a learning environment. The purpose of the literature review is to offer a critical analysis of these issues, recognizing that although AI offers exciting possibilities, there are also ethical issues, technical constraints, and possible obstacles to its successful application. Teachers, developers and researchers may create solutions to these problems by understanding the issues surrounding AI learning tools. This will help to promote the ethical and well-informed incorporation of cutting-edge technologies into education.

2.6 Conclusion

The literature study offers a thorough examination of the advancement, significance, and difficulties faced by AI teaching assistants. It emphasizes the significance of creating user-friendly chatbots that are suited to educational environments by combining findings from a variety of studies. A table of all the references and their descriptions are given in tables D.1 and D.2 in the appendix. All things considered, the literature review provides insightful information about the applications of AI-driven educational technologies.

The requirements gathering process, which will be covered in detail in the upcoming chapter, will involve deciding on specific requirements and goals for the creation of the chatbot. In order to convert the findings from the literature research into real-world requirements that will direct the chatbot system's design and evaluation, this stage is important.

Chapter 3

System Requirements

This chapter provides a thorough analysis of the project's system requirements, including the functional and non-functional requirements necessary for the system's creation and operation. An overview of the system requirements is given in Section 3.1, which also sets the stage for a more in-depth examination in the parts that follow. The functional and non-functional requirements are covered in detail in Sections 3.2 and 3.3, respectively. These sections clarify the essential standards, characteristics, and limitations that are essential to the project's success. The results of the system requirements analysis are compiled and presented in Section 3.4, which serves as the chapter's conclusion.

3.1 Introduction

This chapter focuses on the the requirements gathering section, which creates the groundwork for the project's successful delivery. It acts as an in-depth manual for identifying, documenting, and prioritizing both functional and non-functional requirements. This will ensure that the project's objectives are clearly laid out and, eventually, lead to a sound project completion.

To create a chatbot that is both effective and helpful for both the fields of technology and education, a full set of requirements needs to be defined. The two main categories of these requirements are functional and non-functional. Inaccurate requirement specification can cause delays and cost overruns according to [Shah and Patel, 2016].

Functional requirements describe the specific features that a system must have in order to meet the users needs. These requirements give developers an outline for creating, implementing, and testing elements that support the system's purpose by defining the intended behavior, interactions, and functions of the system.

Non-functional requirements include features other than functions, such as usability, security, performance, and scalability. These factors define the system's quality standards, ensuring that it satisfies requirements for effectiveness, dependability, and user experience. Non-functional requirements are essential to a system's overall success and performance.

The MoSCoW method is used to prioritize the requirements, ranking them from "must-have," "should-have," "could-have," and "won't-have" in order to determine which feature is most important to the project's success and which would be possible for a later iteration. This prioritization technique is highly well-liked since it is straightforward, reliable, and may give the reader the information they need fast [Kravchenko et al., 2022].

3.2 Functional Requirements

Functional requirements focus on specific features and capabilities that the chatbot needs for it meet its main goals. These include the fundamental features of the chatbot, like understanding questions from students and providing accurate responses. The list of functional requirements are given below in table 3.1.

ID	Requirement	Priority
FR1	The chatbot shall allow the sharing of files like as PDFs.	Must
FR2	The chatbot should be able to read and understand lecture slides provided in PDF format.	Must
FR3	The chatbot should accurately understand and interpret questions from students.	Must
FR4	The chatbot should provide relevant answers to student queries.	Must
FR5	The chatbot should authenticate and verify the identity of users, to make sure that only students of Heriot Watt can access the chatbot.	Could
FR6	The chatbot should facilitate interactive learning by providing additional explanations and examples.	Should
FR7	Maintain context within conversations to provide relevant responses.	Must
FR8	Enable the system to recommend articles, papers, and relevant material aligned with user preferences and chat history	Could
FR9	It must support a user-friendly interface for students to interact with the chatbot effectively.	Should
FR10	The chatbot should be capable of handling a high volume of concurrent users without significant performance degradation.	Should
FR11	The chatbot should be compatible with various devices, browsers, and operating systems.	Should
FR12	Allow users to access and review past interactions and conversations.	Should
FR13	Provide onboarding assistance and support for new users to familiarize them with the chatbot's capabilities and features.	Should
FR14	The chatbot must contain a history of the chats and previous PDFs uploaded.	Must

Table 3.1: Functional Requirements for the Teaching Assistant Chatbot.

3.3 Non-Functional Requirements

Non-functional requirements describe the qualities, standards of performance, and restrictions that the chatbot must meet. The requirements cover things like scalability, accuracy, user-friendliness, reaction speed, and data privacy. This section looks into all of the requirements that will guide the development and testing of the chatbot. The list of non-functional requirements are given below in table 3.2.

ID	Requirement	Priority
NFR1	The chatbot should respond to user queries quickly.	Must
NFR2	The chatbot should be available 24/7, with almost no downtime for maintenance.	Must
NFR3	The chatbot should have a high accuracy rate in providing accurate answers to questions (above 90 percent).	Must
NFR4	The chatbot's User Interface (UI) should be simple to use needing no prior experience for students to use successfully.	Should
NFR5	The chatbot needs comply with privacy laws and ensure the privacy of user data and conversations.	Must
NFR6	The system should be scalable to manage more users and newer features for the future.	Could
NFR7	The system should allow for PDF's of upto 1MB in file size to be uploaded	Should

Table 3.2: Non-functional Requirements for the Teaching Assistant Chatbot.

3.4 Conclusion

This chapter's breakdown of functional and non-functional requirements lays a strong basis for the project's next stages. A solution that can be created now that meets the needs and expectations of its users by precisely specifying the characteristics and functions that the teaching assistant chatbot must have. The focus on prioritization guarantees that the most important features are taken care of first, which increases the project's effectiveness and productivity.

The following chapter, which covers the research methodology, the project will go over the steps taken to carry out research, collect information via online surveys and chatbot trials, and put the chosen Agile SCRUM software development life cycle model into practice. The project's goal is to provide an effective and noteworthy solution by implementing this methodology.

Chapter 4

Research Methodology

This chapter explores the research methods used in this study in great detail, providing insight into the models, approaches, and conclusions that were picked. An introduction is provided in Section 4.1, which also sets the stage for a thorough analysis of the technique used. After that, Section 4.2 explores the model that was chosen for the project and explains its importance and suitability. In Section 4.3, the emphasis switches to comparing the SCRUM Agile methodology with alternative approaches in an effort to throw light on the advantages and disadvantages of each. The chapter is concluded in Section 4.4, which summarizes the major conclusions and findings from the analysis of research methodology.

4.1 Introduction

Software engineers use research methodologies to build and develop projects in an efficient and orderly manner. Software engineers have to select an approach for their project that best fits the requirements they have, even though there are many approaches available for developing software. For instance, with the waterfall approach, every stage must be finished before going on to the next. This aspect might guarantee that the functional requirements are implemented successfully, but it might not make the original date. The Agile SCRUM The Software Development Lifecycle (SDLC) model will be used to construct this study. Despite the fact that this is an individual project and it is software development teams that commonly use the Agile model, the study will benefit from the regular analysis and improvement that this approach will provide.

4.2 Selected Model for the project

The Agile SDLC is one of the most commonly used techniques for developing and delivering dependable software. Agile SCRUM approach combines the SCRUM framework with the agile philosophy [Peek, 2023]. Agile helps to meet the needs of present-day projects and is in line with the current stage of development as it handles each project differently than the more conventional waterfall and prototype methods. Also, it seemed that the most appropriate methods to accomplish the project's aims and objectives were quantitative ones, such as a web-based questionnaire and a chatbot experiment.

4.3 SCRUM Agile vs. other Methodologies

Even though SCRUM is mainly used by software engineering teams to develop software in sprints, it still has useful aspects that make it a better option than standard methods like waterfall development. With SCRUM, the work may be split into realistic sprints that allow me to effectively plan my time and dedicate myself to each task. The smallest unit of a scrum is called a sprint, during which usually a small team finishes the given task. It runs between one and three weeks. A sprint backlog defines what has to be done during that sprint. All of the requirements for the current sprint that need to be worked on are listed in the sprint backlog. User stories are lists of requirements created by the product owner and contributed to the product backlog. It is separated into sprint planning, which includes methods for completing a sprint, and sprint backlogs. Agile approaches, as opposed to more standard techniques, offer flexible task distribution and prioritization in combination with effective project management.

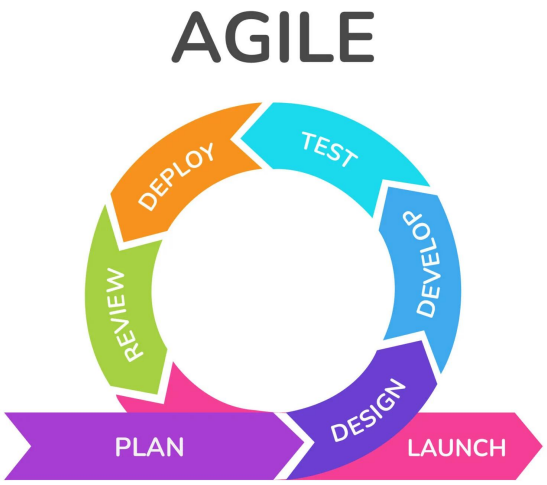

Figure 4.1: Agile Methodology steps from [Amoros, 2022].

4.4 Conclusion

To sum up, this chapter gives an organized way to collect data for the creation of the AI teaching assistant chatbot and to carry out confirmatory research. Throughout its completion, the project will maintain flexibility, adaptability, and continual enhancement due to the adoption of the Agile SCRUM SDLC model. The dynamic nature of AI projects and the requirement for quick prototyping and revision are matched with the Agile methodology's focus on continuous improvement and frequent consumer input.

Following that, the next chapter will explore the project management and risks associated with the chatbot's development. This covers talking about the project schedule, version control systems, risk analysis, and professional, legal and ethical issues.

Chapter 5

Project Management and Risk Analysis

This chapter outlines the project management method that will be used. The timeline for completing the project is described in Section 5.1. An overview of the version control used is given in section 5.2. The potential risks and mitigation techniques are described in section 5.3. The professional, legal, ethical and social issues are covered in section 5.4, and finally the chapter is concluded in section 5.5.

5.1 Project Schedule

Figure A.1 in the appendix shows the Gantt Chart which contains the plan for both semester 1 and 2. For deliverable 1, the workload was split into 8 different stages. The initial stages consisted of choosing a proposal, meeting with the project supervisor, working on the project system and initial research on the topic. The next 7 stages were split into week by week basis. Week 6 is when the project planning, the ethics form, abstract, introduction, aims and objectives were finished. Week 7 requirements analysis and paper reading was finished. During weeks 8 and 9 the design, project management and evaluation strategy were completed. Week 10 the literature review was completed and the next two weeks were used to catch up and add final touches. For deliverable 2, the project schedule has been broken down into seven two-week long sprints. Beginning on January 15th, Sprint 1 will be devoted to selecting and developing the questions for the online questionnaire. Starting the development of the chatbot that will be used in the experiment is another goal of the first sprint. The chatbot is set to be developed and tested along side the requirements in the second sprint. Sprint 3 aims to find participants and start conducting the chatbot experiment and questionnaire on them. In sprint 4, the results of the chatbot experiment and the online questionnaire will be evaluated and documented. Next, sprint 5 focus on updating the dissertation document, and to catch up on any potentially missed work from the previous sprints. Sprint 6 is dedicated to finishing up the final draft for the document and to add final touches to it. Sprint 7, in order to account for potential errors that may occur during the study, has been left empty with no assigned tasks. And at last, on April 15th of 2023, the dissertation will be handed in. The table containing these sprints and the tasks meant to be completed during the sprints are in table 5.1

Sprint	Task
Sprint 1	Gather questions, create online questionnaire, start development on the chatbot and start the D2 document.
Sprint 2	Finish writing the design and implementation of the chatbot and finish development and testing of functionality of the chatbot.
Sprint 3	Find participants and conduct the experiment and questionnaire on them.
Sprint 4	Analyze and document the results of the experiment and questionnaire and add data analysis to the document.
Sprint 5	Update the document and catch up on any missed out work from any of the previous sprints
Sprint 6	Demo my system, write about limitations and future work, finish the D2 document draft and submit the draft.
Sprint 7	Catch up on any missed out work and submit the deliverable and the developed system.

Table 5.1: Sprints and Tasks for Deliverable 2.

5.2 Version Control

For the project life cycle to be clear, seamless, and easy to understand, comprehensive software documentation is necessary. GitHub will be used to implement documentation, and there will be a repository for detailed procedures, guidelines, and end-user documentation there. Fig. 5.1 shows the GitHub Repository and its structure. The project supervisor will have access to this repository, enabling easy communication and information exchange. Tracking project progress and managing code changes all depend on effective version control. The main version control system, GitHub, will be used to host code solutions in a single repository. Git workflows will be used to pull in the most recent code changes, make changes, and publish updates to the main repository. The implementation of Continuous Integration (CI) procedures will automate testing against code modifications, guaranteeing the stability and integrity of the code. Workflows for Continuous Deployment (CD), will make it easier to push updates to production environments, allowing for quicker iteration and the release of new features. The GitHub link: https://github.com/omarriyaz/Teaching_Assistant_Chatbot.

5.3 Risk Analysis

For every risk, a mitigation strategy has been created so that, in case a risk appears, appropriate measures can be taken. The purpose of the risk analysis is to show the chance of the project not being completed on schedule and to reduce the chances that each risk will occur. The risks that might occur during the course of the project will be listed in the risk analysis table, given in table 5.2. Higher-priority risks will be given priority if they occur simultaneously with other risks. Every sprint will conclude with an examination of the project's present state, during which risks that are expected to happen will be noted and addressed with appropriate mitigation strategies.

ID	Description	Impact	Likelihood	Mitigation
R-1	Insufficient time allocated to project	High	Low	The sprints have been carefully created and planned out, including free time in weeks 5 and 7 to catch up on any missed work.
R-2	Loss of work	High	Medium	All of the work is regularly backed up and uploaded online.
R-3	Changes in deadline	Low	Low	Unlikely to happen as the deadlines are already set, and if they are to change it would be within the same week, which is fine as the project will finish before that week.
R-4	Requirements not met	High	Low	Enough time was allocated during the sprints to develop the chatbot and meet all requirements.
R-5	Changes must be made to the requirements late in the development stage	Low	Low	With careful preparation and planning, the chatbot design reduces the risk of last-minute requirement changes. This ensures the development process's resilience.
R-6	Not enough participants	High	Medium	Time has been allocated to find enough participants from Heriot Watt University.
R-7	Not enough Data from questionnaires	Low	Low	The questionnaire will be sent out to a large diverse group of students from Heriot Watt University.
R-8	Delay due to illness	High	Medium	Enough time was allocated during the sprints to make up for lost days due to illness or other problems.
R-9	Unavailability of the Canvas Severs	High	Low	All documents related to the deliverable will be downloaded and stored onto a local device.
R-10	Data Leak	Low	Low	Data from the questionnaire will be stored onto the University's OneDrive.
R-11	Acts of God	High	Low	Acts of God cannot be avoided but is not a concern due to their very low possibilities.

Table 5.2: The Risk Analysis Table.

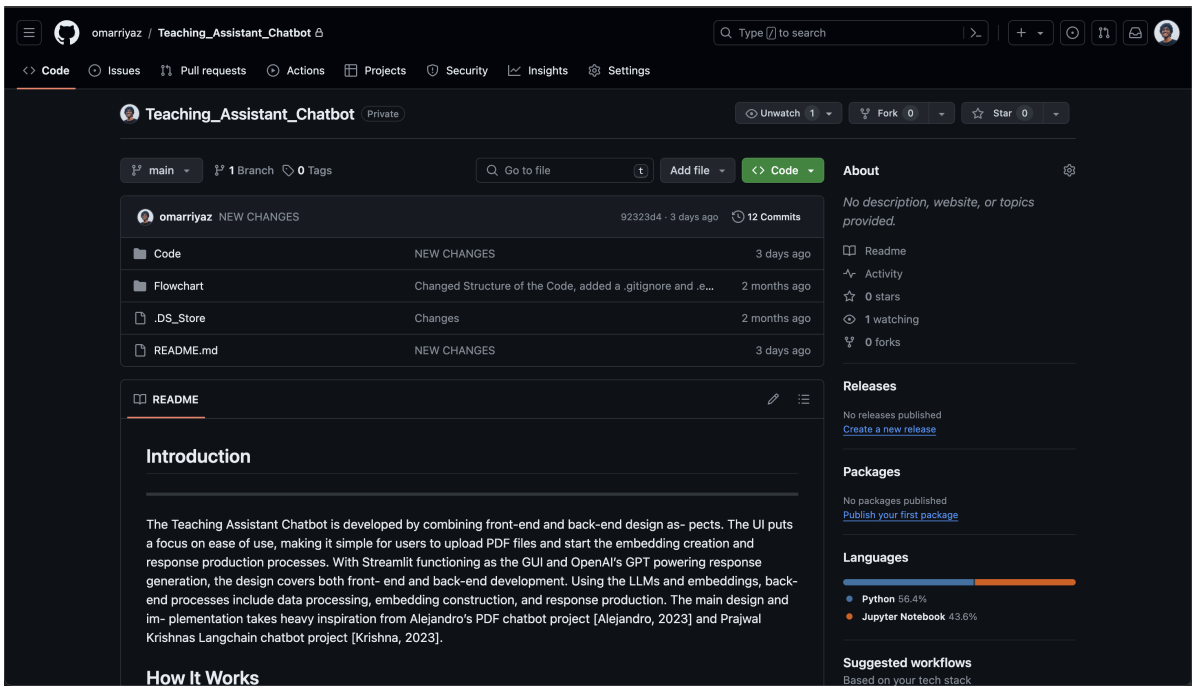

Figure 5.1: The GitHub Repository.

5.4 Professional, Legal, Ethical and Social Issues

Throughout this project, the codes of conduct established by the British Computer Society will be honored and adhered to. The Heriot-Watt University ETHICS board has given the project ethical approval. There will be enough time for participants to receive notifications after completing the chat box experiment and the online questionnaire. All obtained sensitive data will only be stored on the University's single Drive system and will not be distributed to any other parties.

5.4.1 Professional Issues

Working together with the students of Heriot Watt University, particularly when it comes to tasks such as using an chatbot, emphasizes how important trust is. The system's language tone is crucial. It needs to be reassuring in order to create the right level of trust while also avoiding misunderstanding, maintaining clarity, and instilling confidence. Furthermore, it is crucial to follow the British Computer Society's standards of actions. Additionally, given the wide range of technological experiences that University students have, the system needs to give emphasis to usability for students with different technical skill levels.

5.4.2 Legal Issues

Respecting certain legal requirements when managing data is essential to ensuring proper and fair use. Following laws like the Data Protection Act is crucial since it establishes guidelines for the use of data and expands on the GDPR. These rules must be properly followed while

using data to make sure it is accurate, legitimate, and limited to one use. Any use of data must be in line with the system's intended purpose and avoid from accessing unrelated information in order to comply with legal requirements and protect data privacy.

5.4.3 Ethical Issues

Fairness, equity, and openness must be upheld by the project in its operations due to ethical reasons. The Heriot-Watt University ETHICS board has granted ethical approval in this regard, guaranteeing that the project will uphold strict ethical standards during its implementation. Furthermore, the University's single Drive system will be used to safely store sensitive data gathered for the project, with strict safeguards in place to guard against illegal access and release to outside parties.

5.4.4 Social Issues

In order to protect the university students, social considerations must be addressed in order to provide a supportive environment. It is crucial that the system's design refrain from asking for private or sensitive information that may compromise people's rights. For instance, the system must not ask for information relating to protected characteristics. The Equality Act of 2010 forbids direct or indirect discrimination on the basis of protected characteristics, such as age, race, faith, sex, or other distinguishable qualities. Such acts would be in violation of this law.

5.5 Conclusion

In the end, this chapter provides a thorough overview of the project management framework, risk assessment, and professional, legal, ethical, and social factors to consider. It ensures an organized approach to project execution by outlining the deadlines in the project schedule. The integration of Git workflows and CI/CD procedures strengthens the project's code stability and integrity, while the use of GitHub as the main version control platform highlights the project's dedication to clear documentation and efficient code management. The risk analysis segment anticipates problems that can occur during project implementation by identifying potential risks and mitigation solutions. The project's adherence to established standards and guidelines is further emphasized by the discussion of professional, legal, ethical, and social issues. Through the incorporation of these elements into the project management framework, the chapter guarantees the project's seamless advancement and emphasizes its dedication to moral behavior and social accountability.

The next chapter explores the intricacies of design, which was a crucial component in creating the Teaching Support Chatbot. Design is more than just aesthetics, it includes the user interface, user experience, and comprehensive plan for the chatbot's operation.

Chapter 6

Design

This chapter dives deeply into the chatbot's design, covering everything from its architecture to user interface considerations. An overview and introduction are given in Section 6.1, which also serves as a foundation for the in-depth examination that follows. The design of the chatbot is thoroughly reviewed in Section 6.2, covering important elements such its architecture, functionality, and user interaction flow. The chapter concludes with Section 6.3, which summarizes the lessons learned from the design process and provides final thoughts on the ramifications and design principles of the chatbot.

6.1 Introduction

This chapter looks at the techniques and tools to build the chatbot. The goal of this chatbot is to improve assisted teaching by understanding lecture slides as PDFs and providing customized answers to inquiries from students. In order to achieve the project's goals and take a significant step toward proving the hypothesis, the design stage must be gone through.

6.2 Design of the Chatbot

The chatbot is developed by combining front-end and back-end design aspects. The UI puts a focus on ease of use, making it simple for users to upload PDF files and start the embedding creation and response production processes. With Streamlit functioning as the Graphical User Interface (GUI) and OpenAI's Generative Pre-trained Transformer (GPT) powering response generation, the design covers both front-end and back-end development. Using the LLMs and embeddings, back-end processes include data processing, embedding construction, and response production. The main design and implementation takes heavy inspiration from Alejandro's PDF chatbot project [Alejandro, 2023] and Prajwal Krishnas Langchain chatbot project [Krishna, 2023].

6.2.1 Embeddings

Embeddings allow us to organise any text based on its semantic meaning [Maameri, 2023]. Mathematical representations of items or data in a reduced dimension are called embeddings. In NLP, words, sentences, or documents are frequently represented as vectors of real numbers in a continuous vector space using embeddings. Embeddings are used to record similarities and semantic linkages between various pieces based on how they are used in context. For the chatbot, embeddings are calculated for text segments that are taken out of the PDF document. These embeddings allow for quick processing and semantic search by providing a concise and meaningful representation of the textual material. Using the knowledge base that was created during the PDF processing stage, the chatbot makes use of these embeddings to understand and respond to student inquiries.

6.2.2 Software Development Architecture

Similar to well-known systems such as ChatGPT, the features replicate user interfaces and allow for the uploading of PDFs, the creation of queries, and the gathering of responses. The system utilizes GPT 3.5 turbo on the back end to get data from submitted PDFs and produce accurate responses. The stages that make up the implementation process are as follows: uploading PDF files, breaking them up into smaller pieces, and computing embeddings to create a knowledge base. The system then searches this knowledge base for relevant answers in response to student inquiries. The design includes the programming environment, using Visual Studio Code coding with specific packages and a virtual environment. Streamlit is a key tool for Python graphical user interface creation. PyPDF2 is used for PDF reading, the file uploader component, and basic Streamlit functionality. The recursive character text splitter from LangChain is used to meet the need for splitting and the computation of embeddings for these chunks.

6.2.3 State Diagrams and Prototype

Given below in Figures 6.1 and 6.2 are diagrams that show the architecture design of the chatbot, explaining the design behind how the chatbot will accept PDF files and convert into a format where it can understand the content inside of the PDF, in order to be able to answer questions based on it. A low fidelity prototype of the chatbot's UI design is shown in Figure B.1. in the appendix. The prototype presents the main features and layout components of the desired user interface in a simplified visual representation. The prototype has a large "Upload PDFs" button that allows users to upload PDF files with lecture slides or additional relevant course materials. Below this button is a section that shows previously submitted PDFs and related chat logs, making it simple for users to refer to and navigate. When a PDF is uploaded, the chatbot starts a processing phase in which it creates an overview of the document's contents. This summary gives readers a brief rundown of the most important details and highlights the main ideas discussed in the paper.

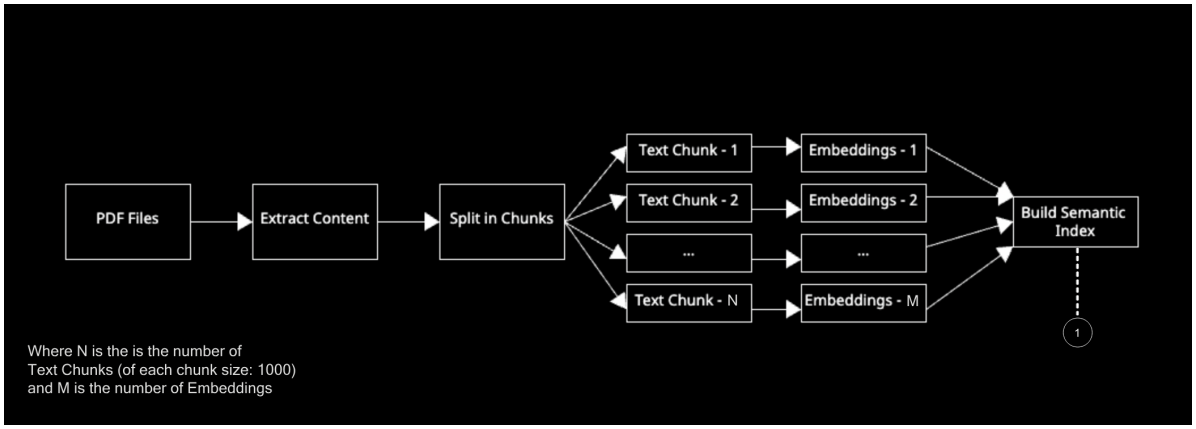

Figure 6.1: State Diagram 1 for the architecture design of the chatbot, continued in Fig. 6.2.

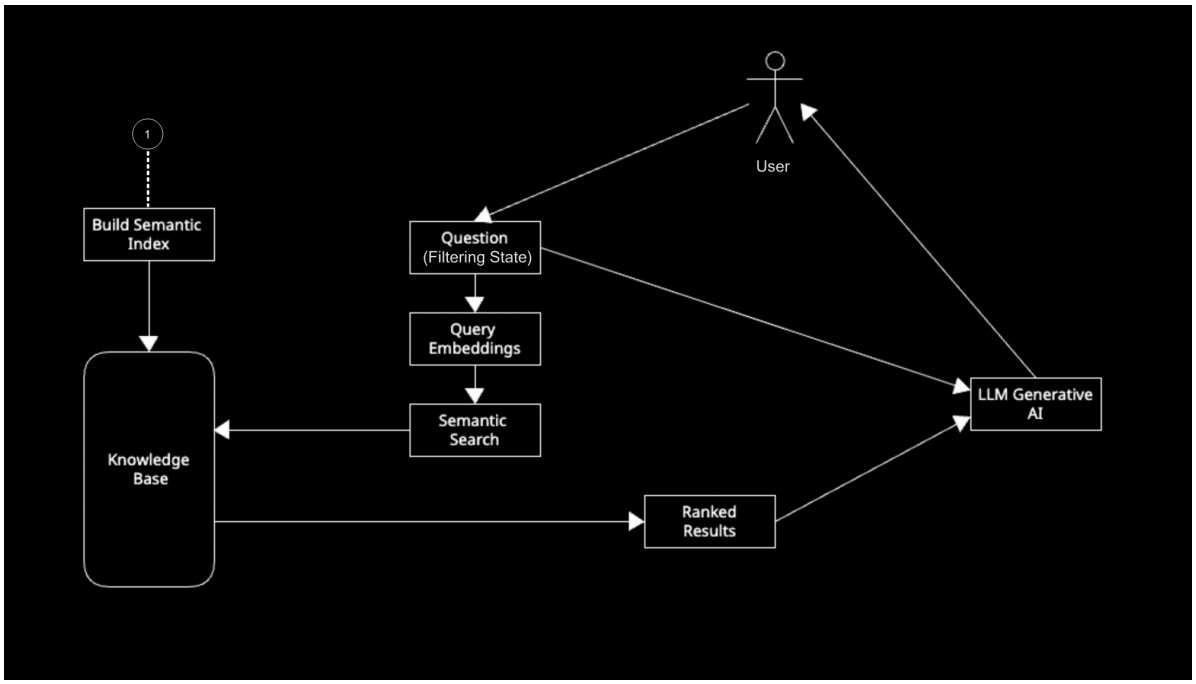

Figure 6.2: State Diagram 2 for the architecture design of the chatbot.

6.2.4 Chatbot Flowchart

Given below in figure 6.3 is the flow chart for the chatbot. The user starts off by opening the Chatbot web page, they then can upload PDF(s) to on the sidebar. They can also open up the settings by clicking on the settings button. Once the PDF(s) is uploaded, it checks whether they are the right size (200MB per file). If they are not, an error message pops up. Once the PDF(s) have been upload, the user clicks on the 'Send to Chatbot' button, a spinner that says processing pops up and then after the PDF is embedded a success/error message is shown depending on the status of embedding the PDF. If it runs into any errors, the user can try pressing the button again. Once successfully uploaded, the PDF is added to the LLMs Knowledge Base. The user can now start the chat, the chatbot is on an infinite loop waiting for the user input. Once a question is asked, the chatbot checks if the answer is available in the Knowledge Base. If yes, it provides the answer, if no, it responds appropriately. The user can then clear the chat or end the session. The software architecture, which is similar to well-known systems like ChatGPT, makes it easy for users to upload PDFs, process content, and generate responses. The architecture and operation of the chatbot are clearly explained using design diagrams, which also include a prototype user interface and state diagrams. A detailed flowchart outlines the user's path from PDF upload to chatbot interaction. In the upcoming chapter, testing and user assessment will be evaluated. Specifically, the chatbot's usability and performance will be evaluated in order to improve its functionality and guarantee that it works well in educational settings.

6.3 Conclusion

In summary, the project's design and implementation phase is a critical step towards the goal of enhancing aided learning using AI. By carefully combining front-end and back-end design considerations, a chatbot has been created that can answer questions from students and understand course slides in PDF format. Using embeddings, it has been made possible for the chatbot to analyze queries quickly and perform semantic searches, which will help it better understand and respond to questions from students.

The implementation of these design elements will be examined in the next chapter, describing the procedures that were followed to turn the chatbot's idea into a working product.

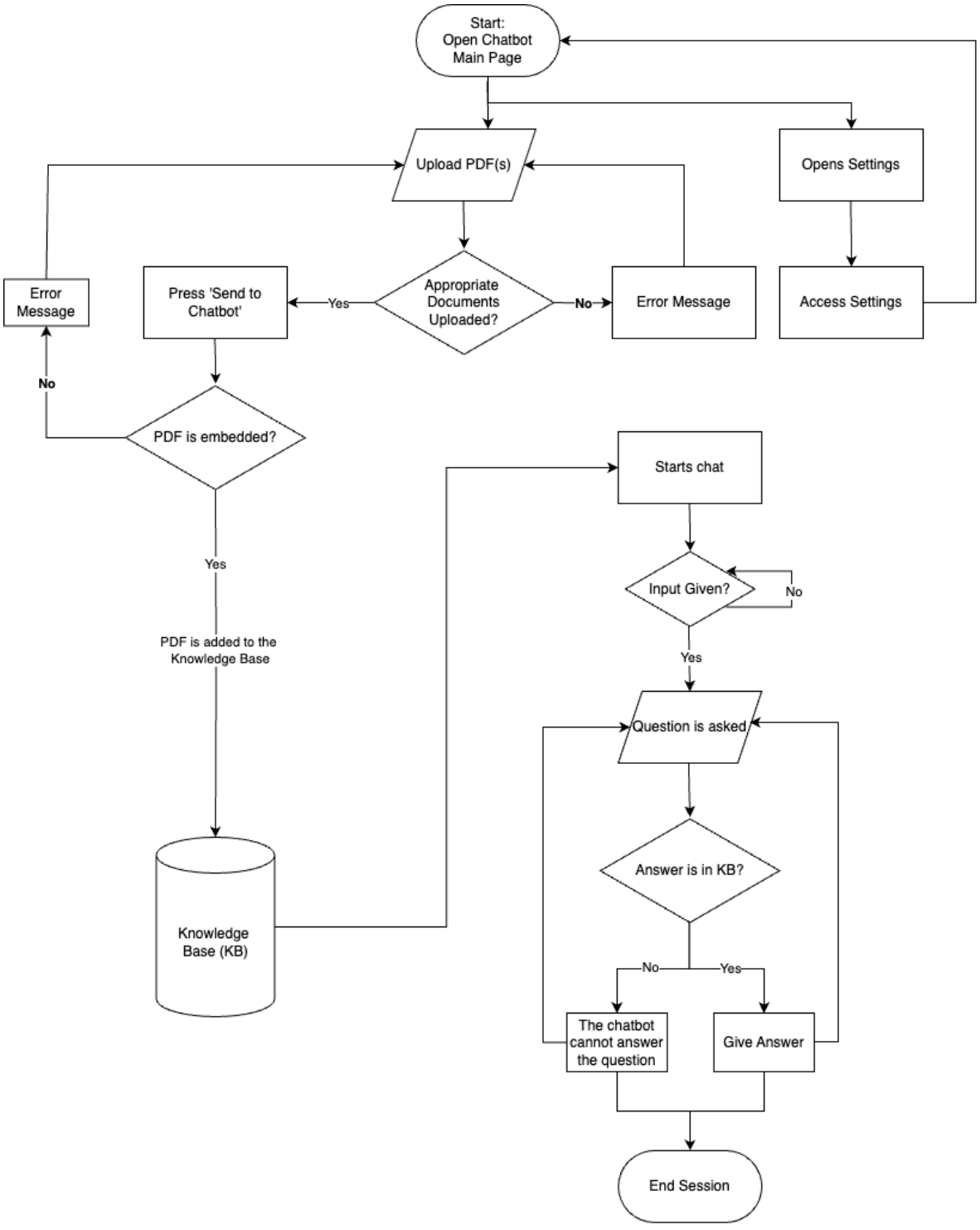

Figure 6.3: The Chatbot flowchart created in draw.io.

Chapter 7

Implementation

This chapter explores how the design concepts presented in the previous chapters are really implemented in practice. It offers insights into the coding techniques, development process, and technological details involved in implementing the AI-driven teaching assistant chatbot. Section 7.1 of this chapter provides an overview of the implementation stage. Section 7.2 provides specifics on the implementation. Then finally 7.3 concludes the chapter.

7.1 Introduction

A key stage in making the Teaching Assistant Chatbot possible is the project implementation phase, which involves employing a variety of tools and approaches in practical settings to achieve the intended functionality. For the implementation of the chatbot, python was used as programming language, the Streamlit python library was used as a GUI for the chatbot. Streamlit is an open-source Python library with strong functionality and ease of use that facilitates the development of interactive web applications for data science and machine learning [Dayanithi, 2023]. OpenAI's GPT 3.5 Turbo was used as Natural Language Understanding (NLU) and to respond to the queries made by the students.

7.2 Project Implementation

The technical implementation of the Teaching Assistant Chatbot is the center of this section. By utilizing Python modules like PyPDF2, Streamlit, langchain, and OpenAI, an advanced system was built that can analyze PDF documents and process user queries. The section examines the usage of these libraries for managing conversational flow, handling user input, extracting and embedding PDFs, and utilizing natural language understanding skills.

7.2.1 Python Libraries

For the Teaching Assistant Chatbot to function, it needs to be implemented using certain important libraries. First of all, Streamlit is the primary tool used to construct the user interface and manage user interactions. Using Streamlit, user-friendly interfaces can be quickly and easily designed and implemented, by leveraging Python to create interactive online apps. The

dotenv package is also used to safely retrieve environment variable-stored API keys, guaranteeing that private data is shielded from unwanted access. Text extraction from user-uploaded PDF documents depends heavily on PyPDF2. The chatbot can process the content of PDF files and extract relevant data to provide responses by utilizing PyPDF2. Additionally, the langchain library offers vectorization, text splitting, and embeddings, among other advanced text processing features. With the use of this library, the chatbot can now read and assess the textual content of PDF documents, making it easier to understand meanings and retrieve relevant information. Last but not least, the OpenAI chat model library provides a cutting edge models for natural language processing, giving the chatbot the language comprehension skills required to produce intelligent answers to user inquiries.

7.2.2 Extracting and Embedding PDFs

The Teaching Assistant Chatbot's ability to work depends on the extraction and embedding of PDFs. The PyPDF2 library is used by the implementation to extract text from submitted PDF documents. The chatbot can analyze PDF files thanks to this library, extracting textual data that may be processed further for analysis. Following extraction, the text is separated using langchain's CharacterTextSplitter. In order to optimize the text for embedding and further analysis, this stage breaks it up into smaller, easier to understand parts. After that, these text chunks are transformed into embeddings using the instructor-xl embedding, which is a text embedding model with instruction fine-tuning that, given the task instruction alone, can produce text embeddings customized for any task (e.g., classification, retrieval, clustering, text evaluation, etc.) and domain (e.g., science, finance, etc.). This embedding model enables the chatbot to interpret the text's semantic content and provide pertinent results in response to user inquiries. Through the process of extracting and embedding PDFs, the chatbot is able to understand and analyze the content of uploaded documents, which enables it to provide intelligent answers to user queries. Fig. 7.1 shows the python code for Extracting and Embedding PDFs.

```
14
15  # Function to extract text from PDFs
16  def get_pdf_text(pdf_docs):
17      text = ""
18      for pdf in pdf_docs:
19          pdf_reader = PdfReader(pdf) # Using the PdfReader class from PyPDF2 library
20          for page in pdf_reader.pages:
21              text += page.extract_text()
22      return text
23
24  # Function to split text into chunks
25  def get_text_chunks(text):
26      text_splitter = CharacterTextSplitter( # Using the CharacterTextSplitter class from langchain library
27          separator="\n", # Single line break
28          chunk_size=1000, # Chunk size of 1000 characters
29          chunk_overlap=200, # Chunk overlap of 200 characters
30          length_function=len # Length
31      )
32      chunks = text_splitter.split_text(text)
33      return chunks
34
```

Figure 7.1: The python functions for Extracting and Embedding PDFs.

7.2.3 Handling User Input

The Teaching Assistant Chatbot's ability to process user input effectively is essential to its operation. Streamlit's text input widget is used to record user input, giving users a simple interface through which to communicate with the chatbot. The implementation uses the handle user input function to process user questions and queries and initiates the chatbot to provide a response. In order to facilitate smooth communication between the user and the system, this function acts as a bridge between the user interface and the chatbot's conversational flow. Through proficiently managing user input, the chatbot can comprehend customer inquiries and furnish pertinent answers, augmenting the application's overall user experience and usefulness. Fig. 7.2 shows the python code for the handle user input function.

```
# Function to handle user input
def handle_user_input(user_question):
    response = st.session_state.conversation({'question': user_question}) # Get response from the conversation chain using the question asked by the user
    st.session_state.chat_history = response['chat_history'] # Update the chat history stored in session state with the conversation

    # Display the chat history
    for i, message in enumerate(reversed(st.session_state.chat_history)):
        if i % 2 == 0:
            # If the index is even, it means it's a Chatbot's message
            st.markdown(f'<div style="background-color: #2b313e; color: white; padding: 10px; border-radius: 10px; margin-bottom: 20px;">Chatbot: {message.content}</div>', unsafe
        else:
            # If the index is odd, it means it's a User's message
            st.markdown(f'<div style="background-color: #475063; color: white; padding: 10px; border-radius: 10px; margin-bottom: 20px;">You: {message.content}</div>', unsafe_all

```

Figure 7.2: The python code for the 'handle user input' function

7.2.4 Conversational Flow

Using langchain's ConversationalRetrievalChain class, a complex conversational retrieval chain controls the Teaching Assistant Chatbot's conversational flow. This chain combines a retriever component to obtain pertinent responses based on user queries and document embeddings with a potent language model, namely the GPT 3.5 Turbo Chat Model. AI and LLMs in particular, such GPT 3.5 Turbo and GPT-4, have demonstrated exceptional competence in understanding, producing, and modifying human language [Mallio et al., 2024]. The language model makes use of cutting-edge natural language processing methods to comprehend user questions' semantics and produce logical answers. To obtain pertinent data from the corpus of PDF documents supplied by users, the retriever component also makes advantage of document embeddings. By integrating these elements, the chatbot can converse meaningfully with users, comprehend their questions, and respond with precision and contextual relevance, improving the application's overall usability and user experience. Fig. 7.3 shows the conversational flow python code.

```python

# Function to create vectorstore
def get_vectorstore(text_chunks):
    embeddings = HuggingFaceInstructEmbeddings(model_name="hkunlp/instructor-xl") # Using the instructor-xl embeddings
    vectorstore = FAISS.from_texts(texts=text_chunks, embedding=embeddings)
    return vectorstore

# Function to create conversation chain
def get_conversation_chain(vectorstore):
    llm = ChatOpenAI() # GPT 3.5 Turbo Chat Model
    memory = ConversationBufferMemory(memory_key='chat_history', return_messages=True)
    conversation_chain = ConversationalRetrievalChain.from_llm(
        llm=llm,
        retriever=vectorstore.as_retriever(),
        memory=memory
    )
    return conversation_chain

```

Figure 7.3: The python code for the conversational flow.

7.2.5 Natural Language Understanding

When provided an instruction, the OpenAI API can carry out NLP tasks like classification or natural language production. One of the most potent language models on the market right now, the Generative Pretrained Transformer 3.5 (GPT-3.5), has an interface provided by OpenAI [Cochran et al., 2023]. Open AI's language models are integrated into the Teaching Assistant Chatbot to help with natural language understanding, which is a fundamental component of its functionality. This cutting edge model provides the chatbot with the ability to understand the semantics of user inquiries and produce logical responses based on contextual data and language comprehension skills. The chatbot can comprehend user input, get relevant information, and produce responses that are both linguistically and contextually correct by utilizing this language model. The chatbot's capacity to comprehend natural language improves its conversational skills by enabling it to hold meaningful conversations with users and offer insightful support based on the information found in uploaded PDF documents.

7.3 Conclusion

In conclusion, the Teaching Assistant Chatbot implementation demonstrates an adaptable integration of frameworks and libraries, offering a user-friendly interface for users to engage with educational material. The chatbot improves learning by providing precise responses through strong natural language processing methods and efficient user input management. Its functionality will be further enhanced by future improvements, which should result in even more educational support and engagement. Screen shots of the entire system are provided in figures E.1, E.2, E.3 and E.4 in the appendix.

The performance and usability of the chatbot will be assessed and tested in the upcoming chapter. This assessment will highlight opportunities for development and offer insights into how well the chatbot meets its intended goals. The goal is to improve the chatbot's capabilities for greater educational support and to further polish it through extensive testing and analysis.

Chapter 8

Testing and Evaluation

In order to give readers an in-depth understanding of the procedures involved in determining the functionality and efficacy of the Teaching Assistant Chatbot, this chapter explores testing and evaluation approaches in great detail. Section 8.1 covers the testing methods and results, section 8.2 covers the evaluation side of the chatbot and finally section 8.3 concludes the chapter

8.1 Testing

This section explores the specifics of testing methodologies, including integration, and unit testing, stressing their significance in confirming the chatbot's operation. This chapter aims to clarify the critical role that testing plays in the chatbot development life cycle by going over testing procedures in depth and laying the groundwork for the evaluation phase that follows.

8.1.1 Introduction

In order to ensure the operation and dependability of the Teaching Assistant Chatbot, testing is an essential aspect of software development. The testing stage of the chatbot's creation is the subject of this section. Each step of the testing process, from integration testing to unit testing, works to improve user satisfaction and the chatbot's quality. The goal is to provide a dependable and efficient chatbot that enhances student learning through thorough assessment.

8.1.2 Functional Testing

Functional testing verifies that newly added functionality to software follow to the requirements specification, it frequently does this by connecting to a user story. Functional testing starts when development tasks are being finished. Documentation related to development reviews for each task will contain confirmation of this as well as the outcomes of any relevant test cases. Functional testing will also include performance testing, which should be conducted in settings that simulate real-world situations in order to assess the impact of any changes made to the responsiveness or stability of the chatbot.

Functional Testing involves:

- Unit Testing : A unit test is a type of software test that focuses on components of a software product. The purpose is to ensure that each unit of software code works as expected. Within the source code of an application, a unit can be any function, method, module, object, or other type of entity [Dizdar, 2023]. This involves testing individual chatbot components to make sure they work properly when used alone, such as particular conversational features or input processing capabilities.

- Integration Testing : Software modules are logically combined and tested as a group in an approach known as integration testing [Hamilton, 2023]. This makes sure the various components of the chatbot, such as the response generation and natural language processing modules, integrate well to deliver cohesive functionality.

8.1.3 Unit Testing Results

Unit testing was used as part of the project to confirm that certain functions in the program worked as intended. The behavior of code units was assessed by creating unit tests, which had allowed the code generated to be verified.

For example, unit tests were put in place for the functions get_pdf_text and get_text_chunks, which are essential parts of the application's document processing module. The get text chunks function divides a supplied text into smaller chunks according to established criteria, whereas the get_pdf_text function retrieves text from PDF documents.

A sample PDF document with known text was provided for the get_pdf_text test, which checked to see if the function successfully extracted the text. Similarly, a sample text was supplied for the get_text_chunks test, which checked if the function appropriately separated it into chunks based on the given parameters.

These unit tests developed an organized approach to verifying how important functions within the program behaved. Fig. 8.1 shows the completed unit testing result.

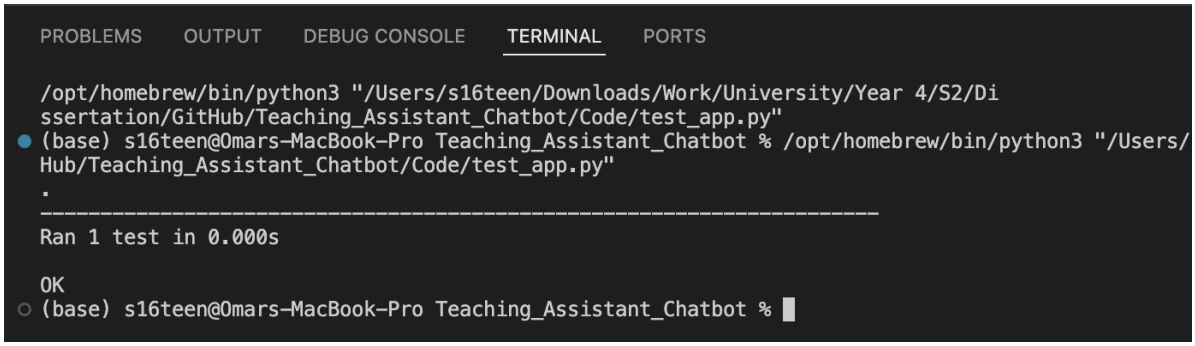

Figure 8.1: Unit testing results for some functions in the chatbot code.

8.1.4 Integration Testing

Ensuring the resilience and dependability of the application requires the integration of error management methods, like the one found in the Teaching Assistant Chatbot. This error-handling code is a vital part of integration testing, which is an important stage in the software development process. The application's durability to unanticipated failures is improved by implementing specific error handling for probable problems, such the unavailability of the OpenAI API. Integration testing greatly enhances the Teaching Assistant Chatbot's overall quality and user experience by methodically testing and improving error handling techniques.

```
# Error handling
except Exception as e:
    st.error(f"Error occurred: {str(e)}") # Error message
    # Check if the error is related to OpenAI API
    if "ConnectionError" in str(e):
        st.error("OpenAI API is not accessible at the moment. Please try again later.")
```

Figure 8.2: Integration testing code for the OpenAI API.

8.1.5 Functional and testing Traceability matrix

A testing tool called a traceability matrix records all user requirements together with the specifics of the test cases that are mapped to each need [Bakshi, 2021].The following table in 8.1 showcases the functional Requirements Traceability Matrix corresponding to the system requirements outlined in Table 3.1 after implementation and testing.

Req. ID	Test ID	Test Case	Status
FR1	TC01	Streamlits PDF Upload function works as intended.	Pass
FR2	TC02	PyPDF PDF reader can read the PDF Uploaded.	Pass
FR3	TC03	A LLM is connected to the chatbot.	Pass
FR4	TC04	A powerful LLM is used to provide relevant answers.	Pass
FR5	TC05	Chatbot can authenticate to only allow Heriot Watt Students	Fail
FR6	TC06	GPT 3.5 can provide additional explanations and examples	Pass
FR7	TC07	The chatbot has a conversational flow and history	Pass
FR8	TC08	GPT can provide recommend articles, papers, and relevant material	Pass
FR9	TC09	Creating a user friendly interface with Streamlit	Pass
FR10	TC10	Deploy the app using Streamlit	Fail
FR11	TC11	The chatbot is compatible with all browsers and devices.	Pass
FR12	TC12	The chatbot shows its history	Pass
FR13	TC13	Provide onboarding screen	Fail
FR14	TC02	Provide history for each PDF	Fail

Table 8.1: The Requirements Traceability Matrix

8.1.6 Traceability Matrix Results

The first requirement, FR1, is verified by TC01, which evaluates how well Streamlit's PDF upload capability works. This shows that the chatbot has successfully allowed the sharing of documents such as PDFs with a passing status. TC02 validates FR2, ensuring that the PyPDF library can efficiently read the PDFs uploaded by users, therefore verifying the chatbot's comprehension of lecture slides that are supplied in PDF format.

TC03 validates FR3 by confirming that an LLM is correctly connected to the chatbot and that it is integrated with an appropriate LLM, this LLM being GPT 3.5 Turbo. TC04 then confirms FR4, suggesting that the linked LLM can successfully respond to inquiries from students.

Not all requirements, though, are validated. One example of a feature that is not properly implemented is FR5, which deals with the chatbot's authentication of Heriot Watt students. This feature fails validation in TC05. The reason for this is issues with Streamlit authentication feature. This requirement was a low priority one, thus the requirement was ignored. FR13 also fails validation in TC13, the reason for this is that no on-boarding screen as such was created, but the main page provides instructions for the user on how to use the chatbot.

Other requirements, meanwhile, successfully complete validation. The test cases corresponding to FR6, FR7, FR8, FR9, FR11, and FR12 confirm that the chatbot supports interactive learning, preserves conversational flow and history, suggests relevant material, has a simple user interface, works across multiple platforms, and presents conversation history as intended.

Lastly, FR10 and FR14 have room for improvement, as evidenced by their failed test cases (TC10 and TC02, respectively), which point to the necessity of implementing PDF history provision and optimizing for heavy user loads. The reason for TC10 failing is due to some issues with connecting streamlit with the Heriot Watt GitHub organization. Again this requirement is not of high priority, thus it was also ignored. The reason for TC14 is different though, originally the plan was to make the chatbot in a way that the student can only talk to one PDF per chat and then view all his individual chats in a history. The functionality of the finished product is different so this requirement does not apply. The system now allows the student to talk to multiple PDFs at the same time, so there is no need for a PDF history anymore.

8.2 Evaluation

The evaluation process is examined in detail in this section, along with the important metrics, methods of research, and data analysis techniques used to determine the chatbot's influence on the learning environment. The evaluation approaches used are thoroughly examined in the following sections, which also offer insights into the benefits, drawbacks, and implications of the chatbot's performance as a teaching assistant. This chapter seeks to offer insightful information about the chatbot's usefulness and efficacy in improving the educational experiences for users.

8.2.1 Introduction

An evaluation strategy is a methodical approach of evaluating a project's effectiveness, functionality, and performance. To make sure the chatbot satisfies user expectations and predetermined standards, it is essential to establish an evaluation approach during the chatbot creation process.

8.2.2 Project Evaluation Strategy

Two different testing approaches will be used throughout the development process to determine the chatbot's effectiveness. Functional testing and research questions. First, a thorough assessment of the chatbot's performance will be conducted to determine its functionality and capabilities. And secondly, real users will also be given online surveys to complete in order to assess the chatbot's usefulness and effectiveness in real-world situations. Together, these two testing strategies will help provide an exhaustive assessment of the quality and usability of the chatbot.

8.2.3 How Data will be collected

This research section of the project places a heavy focus on gathering data from consenting participants in order to validate the project's key ideas. To gather relevant information for the research, two quantitative methods will be employed:

1. A chatbot experiment.
2. An online questionnaire.

8.2.4 The experiment

A user study was performed with participants from the target user population, Heriot Watt students enrolled in relevant courses, for the chatbot experiment. The purpose of the experiment was to assess the Teaching Assistant Chatbot's usefulness and efficacy in meeting students educational needs. The participants first received a participant ID, then signed a consent form and a pre-experiment form to obtain some demographic information about them. These forms are shown in Figures C.1 and C.2 respectively in the appendix. They then were asked to upload a PDF relevant to their course onto the chatbot, ask it some questions and then fill out a post-experiment questionnaire, which is shown in fig. C.3 and C.4 in the appendix in which they ranked the chatbot on various metrics.

8.2.5 Pre-Experiment Questionnaire Results

The purpose of the pre-experiment questionnaire was to collect background data on the participants use of chatbots as well as demographic information. Information on the participants' age, gender, current study program, and whether or not they had used chatbots before was requested. The results showed that showed that 67 percent of participants were men and the remaining were women. The age ranges of the participants fell under above 18 and under 30. 42 percent of participants were computer science students, and all 12 have at least once used a chatbot, with 10 of them regularly using them. Figures 8.3 to 8.6 below show visualizations on these findings.

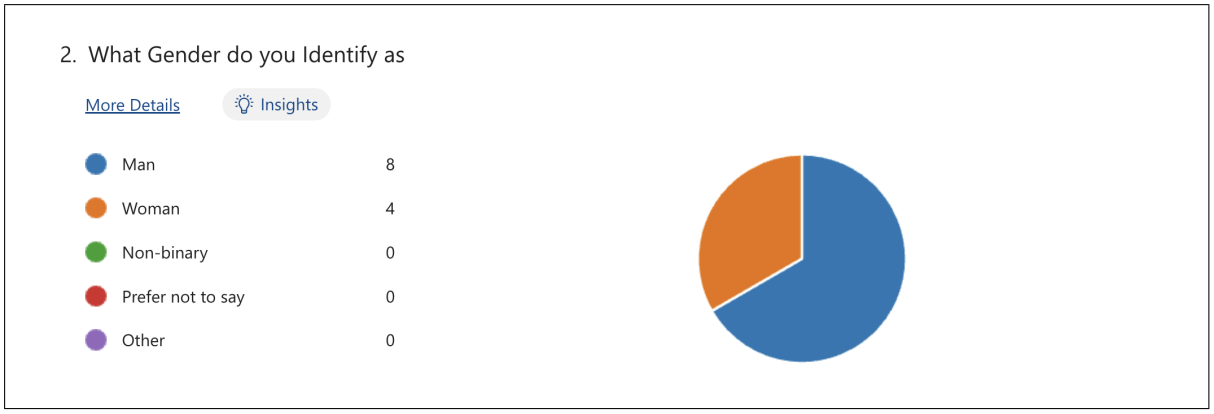

Figure 8.3: Pie chart showing the distribution of genders of the participants.

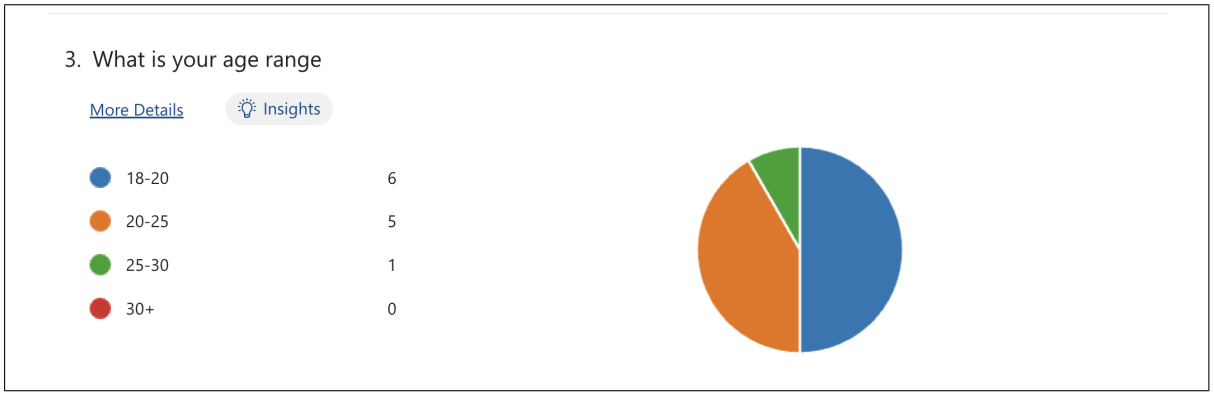

Figure 8.4: Pie chart showing the distribution of Age of the participants.

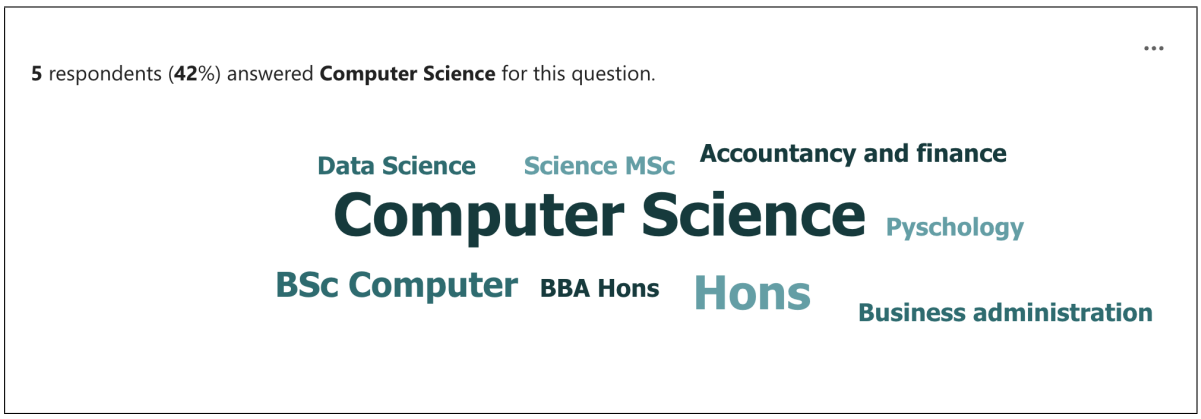

Figure 8.5: Visualization showing the distribution of courses taken by the participants.

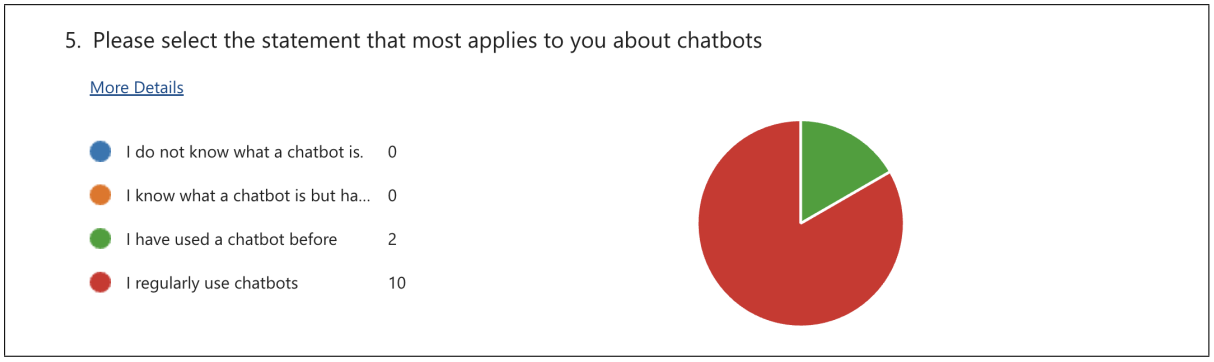

Figure 8.6: Pie chart on how familiar with chatbots the participants were.

This data provide insights into a participant demographic that is primarily male and consists primarily of people between the ages of 18 and 30. The fact that a sizable fraction of participants were computer science majors is important and suggests that they may have some familiarity with technological tools and systems. In addition, the importance and possible effect of the Teaching Assistant Chatbot within this user group are highlighted by the fact that all participants had prior experience with chatbots, with the majority using them on a regular basis. These revelations offer insightful background information that is helpful in assessing the experiment's later outcomes and understanding the viewpoints and interactions of the participants with the chatbot.

8.2.6 Post-Experiment Questionnaire

During the post-experiment evaluation, the Teaching Assistant Chatbot's effectiveness will be evaluated with the widely used System Usability Scale (SUS) alongside with a model similar to the Godspeed Questionnaire model [Bartneck, 2008]. This thorough assessment approach seeks to provide a comprehensive understanding of the chatbot's functioning and usability by delving into a number of areas, such as correctness, practicality, efficacy, and user experience. The average rating, out of 5 were the following: 4.67 for accuracy, usability and helpfulness and a 4 for responsiveness. Figures 8.7 to 8.10 show visualizations on the questionnaire results about perceived accuracy, responsiveness, usability and helpfulness.

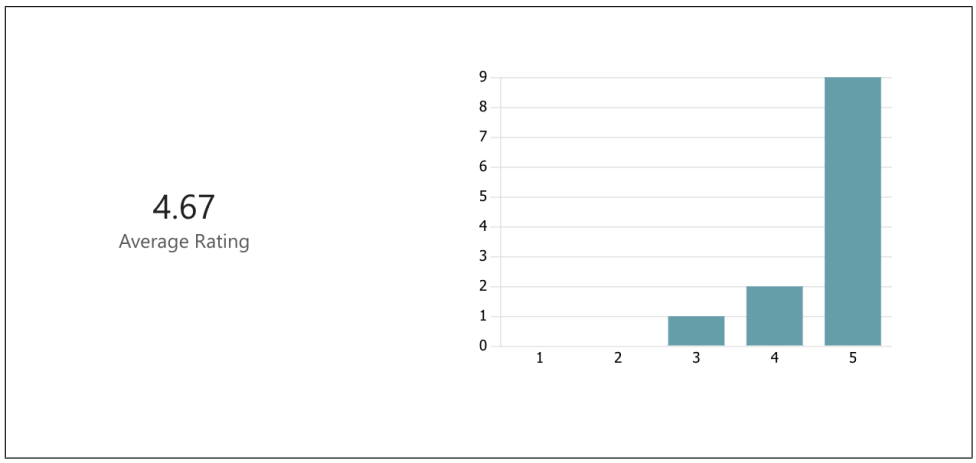

Figure 8.7: Bar chart on how accurate the participants found the chatbots

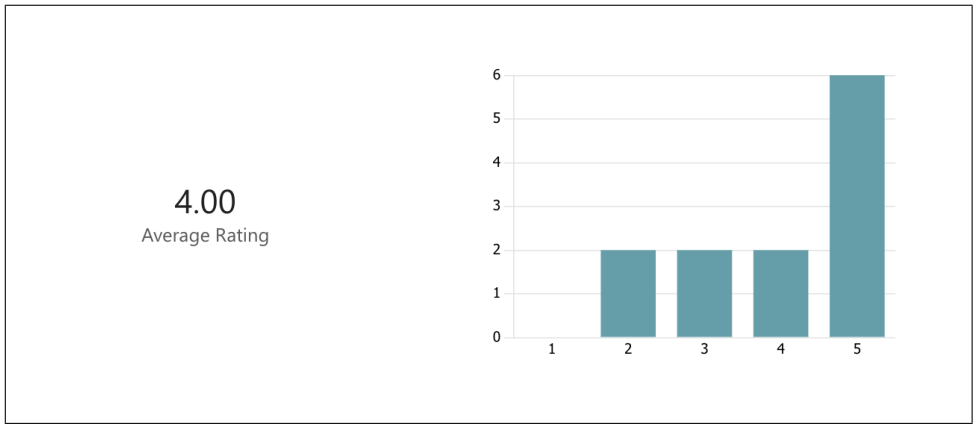

Figure 8.8: Bar chart on how responsive the participants found the chatbots

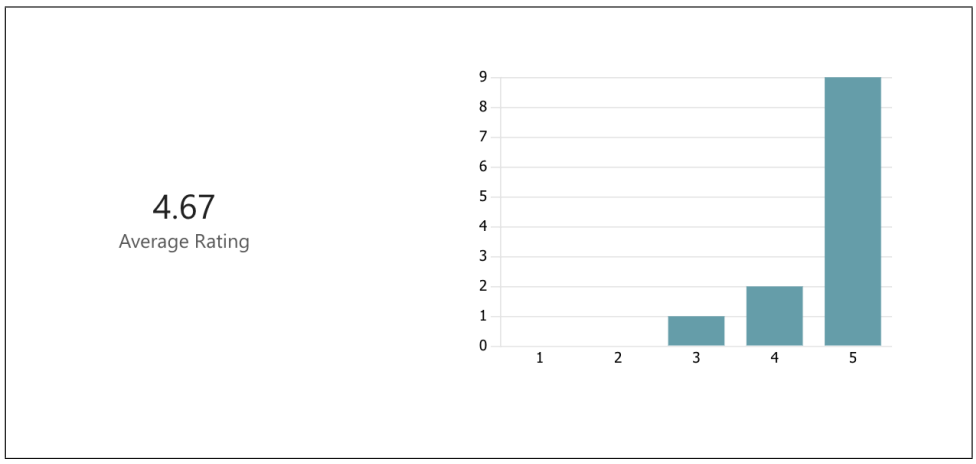

Figure 8.9: Bar chart on how usable the participants found the chatbots

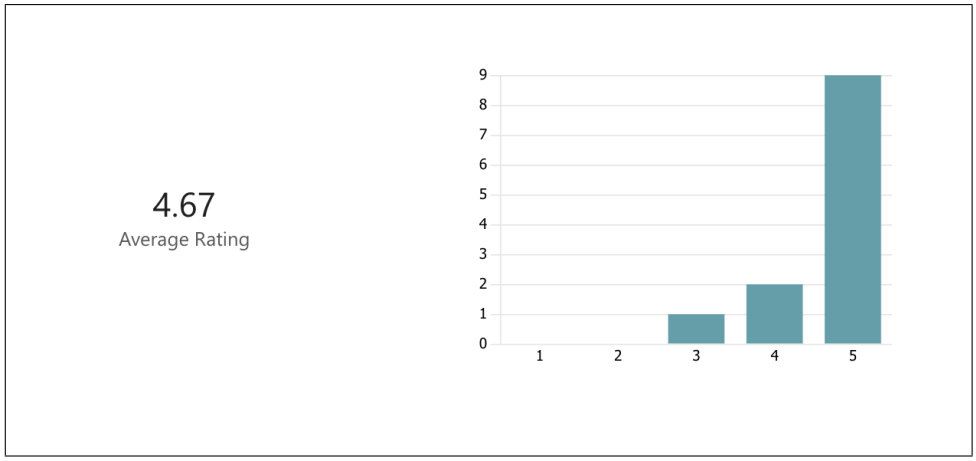

Figure 8.10: Bar chart on how helpful the participants found the chatbots

It's clear from the post-experiment questionnaire answers that the Teaching Assistant Chatbot scored highly on a number of important metrics, demonstrating its efficacy in giving users accurate and helpful support. Although the chatbot performs well overall, its promptness in replying to user queries might use some work, as seen by its somewhat lower responsiveness rating of 4. The reasoning and potential fix for this low responsiveness rating is covered the in the next chapter where limitations and future work are described.

The findings of the SUS questionnaire, which are shown in Figure 8.11, show that every participant gave a good response to questions about the chatbot's usability. More specifically, when asked about the convenience, helpfulness, usefulness, and ease of use of the chatbot, participants mainly chose answers that fell between the categories of "somewhat agree", "agree" and "strongly agree." All of the participants in this collective response seem to have agreed on one thing, about how user-friendly the system was, they all said that the chatbot was simple to use.

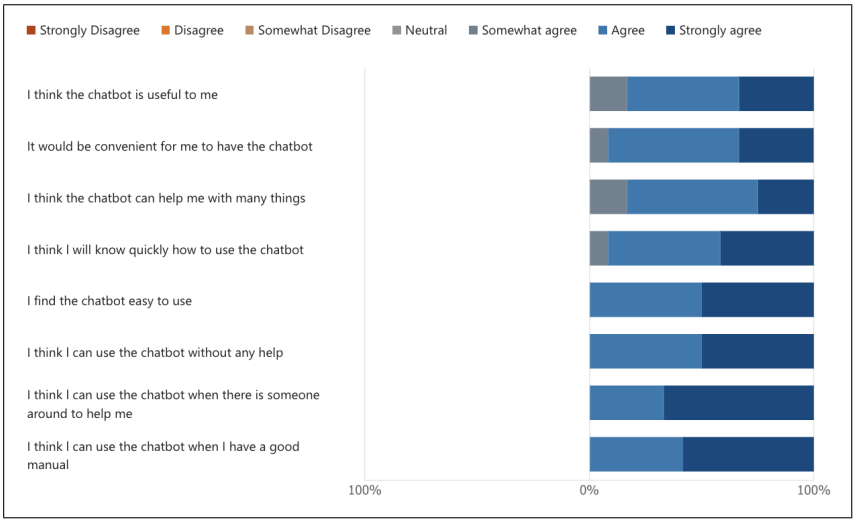

Figure 8.11: The post-experiment SUS questionnaire results

Overall, the good results of the SUS questionnaire highlight how well the chatbot has performed in providing users with an easy-to-use and practical way to support them in their studies. This encouraging response points to the chatbot's potential acceptability and adoption within its intended user base, providing a strong basis for its ongoing use and future growth in educational environments.

8.3 Conclusion

To sum up, the evaluation chapter offers insightful information about the functionality and perception of the Teaching Assistant Chatbot among users. A thorough evaluation of the chatbot's efficacy was carried out using a combination of pre-experiment questionnaires, post-experiment assessments utilizing models such as the SUS, and customized metrics for accuracy, usability, helpfulness, and responsiveness. The participants reaction was overwhelmingly positive, as seen by the high evaluations. Even though there were a few small areas, like responsiveness, that needed work, overall the chatbot showed that it may be a useful tool for supporting learning tasks.

Finally, the next and last chapter will explore possible directions for further research and address the limitations that occurred throughout this study. This will ultimately contribute to the chatbot's continuous success in assisting educational activities.

Chapter 9

Conclusion

In this chapter, section 9.1 provides a summary of the main conclusions and insights discovered during the study process while reflecting on the journey traveled in this dissertation. In Section 9.2, the study's limits are analyzed, along with how they might affect the study's conclusions. In order to expand on the study's findings and further the field, Section 9.3 concludes by outlining potential directions for future research.

9.1 Critical Reflection

In conclusion, the primary goals of this dissertation, creating and assessing the Teaching Assistant Chatbot, have been effectively met. The goal of the project was to develop an AI-powered chatbot that could comprehend lecture slides in PDF format and provide personalized answers to queries, helping students with their academic assignments. The chatbot was created to successfully carry out its intended function by putting front-end and back-end design concerns into practice. A variety of tools and technologies were integrated throughout implementation, such as OpenAI's GPT for response generation, Python for back-end development, and Streamlit for the user interface. The evaluation phase provided more evidence of the chatbot's effectiveness, as participants expressed satisfaction with its accuracy, usability, and helpfulness.

Technical procedures were adhered throughout the development process, including handling user input, extracting and embedding text from PDFs, and maintaining a conversational flow. To guarantee effective task execution and optimize the development process, libraries like PyPDF2, Streamlit, and langchain were employed. The chatbot's usability and functioning were greatly influenced by these technological methods, which also laid the groundwork for its successful deployment.

The project placed a strong emphasis on the user experience, and pre- and post-experiment questionnaires were used to assess how accurate, user-friendly, helpful, and responsive the chatbot was. The findings showed that users gave the chatbot good marks in these categories, considering it to be user-friendly and beneficial for their educational experience. The chatbot's design and implementation were informed by pre-experiment questionnaire insights, and post-experiment assessments provided insightful feedback for future improvements.

9.2 Limitations

Despite the project's successes, a number of limitations need to be addressed. The chatbot's capability was restricted due to time and resource constraints, with its primary focus being on comprehending PDF course slides and providing relevant answers. Some other limitations include using a better embedding model; instructor-xl, a free model, was used; however, the processing time of PDFs would be significantly reduced by using OpenAI's embedding model. The speed and correctness of the generated replies would also be improved by using GPT-4 or any other LLM model that might be developed in the future in place of the one that is now being used. Finally, using a website created with HTML, CSS, and JavaScript would raise the user experience ratings because Streamlit was unable to perform some intended functions.

9.3 Future Work

In order to increase the chatbot's reach and accessibility, future development and extensions could focus on improving its functionality and integrating it with already-existing learning systems. Innovative features like personalized recommendations and natural language understanding could improve learning even more by offering each student individualized support based on their unique requirements and learning preferences. In addition, the integration of machine learning methodologies for examining user interactions and feedback may improve the chatbot's ability to enhance its responses in real-time and adjust to evolving user preferences.

Additionally, investigating how to incorporate multimedia content, like audio and video lectures, into the chatbot's functionalities might result in a more thorough educational experience. To provide more varied and dynamic responses, this might involve implementing algorithms to extract relevant information from multimedia sources and adding it to the chatbot's knowledge base. Furthermore, by using emotion analysis to assess user opinions and emotions during conversations, the chatbot may be able to respond to users with greater empathy and support, which would increase user happiness and engagement.

To evaluate the chatbot's long-term effects on student learning results and engagement, long-term research could be carried out. Through the ongoing monitoring of students' academic achievements, happiness levels, and usage habits, researchers could learn a great deal about the effectiveness of the chatbot and identify areas in need of development. Further information about user preferences, requirements, and issues might be obtained through user research and surveys with a bigger and more varied sample of students, which could help with the design and development of the chatbot's future versions.

Bibliography

[Abdelhamid and Katz, 2020] Abdelhamid, S. and Katz, A. (2020). Using chatbots as smart teaching assistants for first-year engineering students. In *Association for Engineering Education - Engineering Library Division Papers*, Atlanta. American Society for Engineering Education-ASEE. 7, 53

[Aggarwal et al., 2023] Aggarwal, S., Mehra, S., and Mitra, P. (2023). Multi-purpose nlp chatbot : Design, methodology conclusion. *arXiv.org*. 5, 52

[Alejandro, 2023] Alejandro (2023). Chat with multiple pdfs. https://github.com/alejandro-ao/ask-multiple-pdfs. 20

[Ali and Djalilian, 2023] Ali, M. J. and Djalilian, A. (2023). Readership awareness series – paper 4: Chatbots and chatgpt - ethical considerations in scientific publications. *The ocular surface*, 28:153–154. 6, 52

[Amoros, 2022] Amoros, J. L. (2022). The Agile Development Process for Mobile Apps — Krasamo — krasamo.com. https://www.krasamo.com/agile-development-process/. [Accessed 28-11-2023]. vi, 14

[Bakshi, 2021] Bakshi, A. (2021). Requirements traceability matrix in an Agile process – Webomates — webomates.com. https://www.webomates.com/blog/software-requirement-metrics/9-reasons-why-requirements-traceability-is-important-in-agile/. [Accessed 29-11-2023]. 31

[Bartneck, 2008] Bartneck, C. (2008). The Godspeed Questionnaire Series - Christoph Bartneck, Ph.D. — bartneck.de. https://www.bartneck.de/2008/03/11/the-godspeed-questionnaire-series/. [Accessed 13-04-2024]. 36

[Bisser, 2021] Bisser, S. (2021). *Microsoft conversational ai platform for developers : end-to-end chatbot development from planning to deployment / Stephan Bisser*. 1st ed. 2021. edition. 5, 52

[Chaves and Gerosa, 2021] Chaves, A. P. and Gerosa, M. A. (2021). How should my chatbot interact? a survey on social characteristics in human-chatbot interaction design. *International journal of human-computer interaction*, 37(8):729–758. 8, 53

[Cochran et al., 2023] Cochran, K., Cohn, C., Rouet, J. F., and Hastings, P. (2023). Improving automated evaluation of student text responses using gpt-3.5 for text data augmentation. In Wang, N., Rebolledo-Mendez, G., Matsuda, N., Santos, O. C., and Dimitrova, V., editors, *Artificial Intelligence in Education*, pages 217–228, Cham. Springer Nature Switzerland. 28

[Dayanithi, 2023] Dayanithi (2023). Streamlit in 2 Mins — medium.com. https://medium.com/data-and-beyond/streamlit-d357935b9c. [Accessed 10-04-2024]. 25

[Dizdar, 2023] Dizdar, A. (2023). Unit Testing: Definition, Examples, and Critical Best Practices — brightsec.com. https://brightsec.com/blog/unit-testing/. [Accessed 28-11-2023]. 30

[Essel et al., 2022] Essel, H. B., Vlachopoulos, D., Tachie-Menson, A., Johnson, E. E., and Baah, P. K. (2022). The impact of a virtual teaching assistant (chatbot) on students' learning in ghanaian higher education. *International Journal of Educational Technology in Higher Education*, 19(1):1–19. 6, 53

[Følstad and Brandtzaeg, 2020] Følstad, A. and Brandtzaeg, P. B. (2020). Users' experiences with chatbots: findings from a questionnaire study. *Quality and user experience*, 5(1). 2

[Garcia Brustenga, 2018] Garcia Brustenga, F.-A. . M.-C. (2018). Briefing paper: chatbots in education. 1

[Hamilton, 2023] Hamilton, T. (2023). Integration Testing: What is, Types with Example — guru99.com. https://www.guru99.com/integration-testing.html. [Accessed 28-11-2023]. 30

[Kim et al., 2020] Kim, J., Merrill, K., Xu, K., and Sellnow, D. D. (2020). My teacher is a machine: Understanding students' perceptions of ai teaching assistants in online education. *International journal of human-computer interaction*, 36(20):1902–1911. 1

[Kravchenko et al., 2022] Kravchenko, T., Bogdanova, T., and Shevgunov, T. (2022). Ranking requirements using moscow methodology in practice. In Silhavy, R., editor, *Cybernetics Perspectives in Systems*, pages 188–199, Cham. Springer International Publishing. 11

[Krishna, 2023] Krishna, P. (2023). Langchain pdf app. https://github.com/praj2408/Langchain-PDF-App-GUI. 20

[Kuhail et al., 2023] Kuhail, M. A., Alturki, N., Alramlawi, S., and Alhejori, K. (2023). Interacting with educational chatbots: A systematic review. *Education and information technologies*, 28(1):973–1018. 8, 53

[Lazarus, 2010] Lazarus, J. (2010). *Successful NLP*. Crimson Publishing, Richmond. 5, 52

[Maameri, 2023] Maameri, S. (2023). Building a Multi-document Reader and Chatbot With LangChain and ChatGPT — betterprogramming.pub. https://betterprogramming.pub/

building-a-multi-document-reader-and-chatbot-with-langchain-and-chatgpt-d1864d47e339. [Accessed 29-11-2023]. 21

[Mallio et al., 2024] Mallio, C. A., Bernetti, C., Sertorio, A. C., and Zobel, B. B. (2024). Chatgpt in radiology structured reporting: analysis of chatgpt-3.5 turbo and gpt-4 in reducing word count and recalling findings. *Quantitative imaging in medicine and surgery*, 14(2):2096–2102. 27

[Medeiros et al., 2023] Medeiros, T., Medeiros, M., Azevedo, M., Silva, M., Silva, I., and Costa, D. G. (2023). Analysis of language-model-powered chatbots for query resolution in pdf-based automotive manuals. *Vehicles*, 5(4):1384–1399. 2

[Network, 2023] Network, d. (2023). How chat gpt is transforming the role of teachers in today's world of digital learning? 7, 53

[Ng et al., 2022] Ng, J., Haller, E., and Murray, A. (2022). The ethical chatbot: A viable solution to socio-legal issues. *Alternative law journal*, 47(4):308–313. 6, 52

[Peek, 2023] Peek, S. (2023). What Is Agile Scrum Methodology? - businessnewsdaily.com — businessnewsdaily.com. https://www.businessnewsdaily.com/4987-what-is-agile-scrum-methodology.html. [Accessed 28-11-2023]. 13

[Scheider et al., 2023] Scheider, S., Bartholomeus, H., and Verstegen, J. (2023). Chatgpt is not a pocket calculator – problems of ai-chatbots for teaching geography. *arXiv.org*. 9, 53

[Shah and Patel, 2016] Shah, T. and Patel, S. (2016). A novel approach for specifying functional and non-functional requirements using rds (requirement description schema). *Procedia Computer Science*, 79:852–860. Proceedings of International Conference on Communication, Computing and Virtualization (ICCCV) 2016. 10

[Smestad and Volden, 2019] Smestad, T. L. and Volden, F. (2019). Chatbot personalities matters: Improving the user experience of chatbot interfaces. In *Internet Science*, pages 170–181, Cham. Springer International Publishing. 5, 52

[Song et al., 2022] Song, D., Oh, E. Y., and Hong, H. (2022). The impact of teaching simulation using student chatbots with different attitudes on preservice teachers' efficacy. 25(3):46–59. 7, 53

[Tamayo, 2020] Tamayo, P. A. (2020). Should we welcome robot teachers? - Ethics and Information Technology — link.springer.com. https://link.springer.com/article/10.1007/s10676-016-9387-z. [Accessed 29-11-2023]. 9, 53

[Vinita et al., 2022] Vinita, K. P., Sonali, S., Anshita, S., and Mohammad, S. S. (2022). Medical chatbot using ai and nlp. *i-Manager's Journal on Software Engineering*, 16(3):46. 5, 52

Appendix A - Gantt Charts

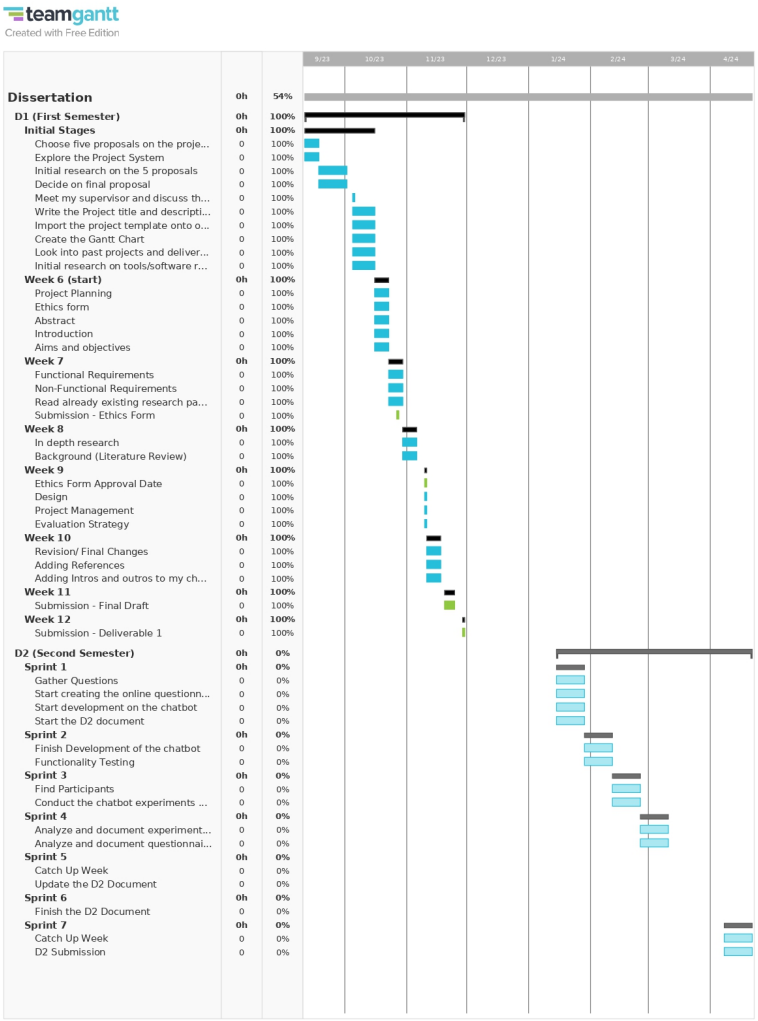

Figure A.1: Gantt Chart for Deliverables 1 and 2

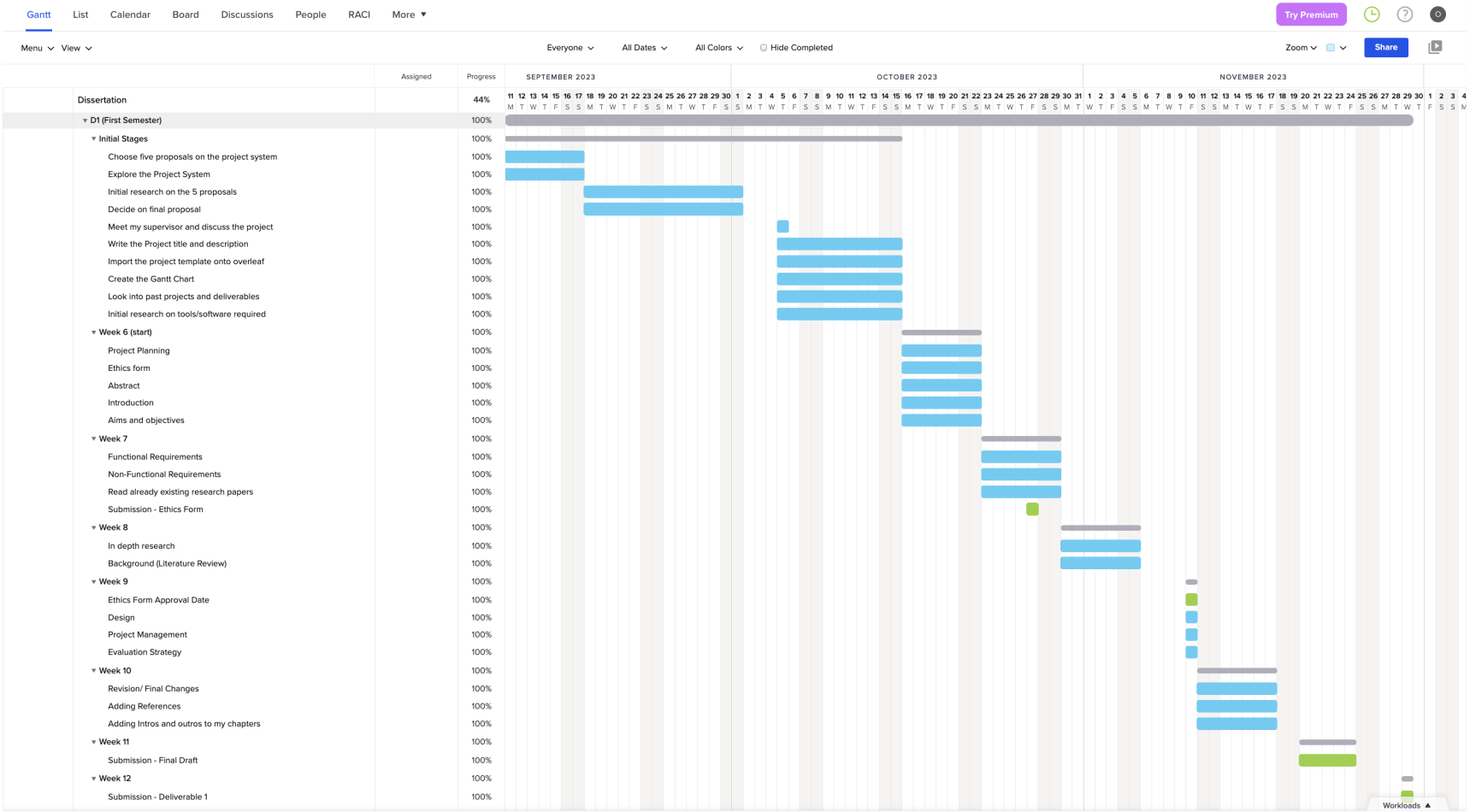

Figure A.2: Gantt Chart for Deliverable 1

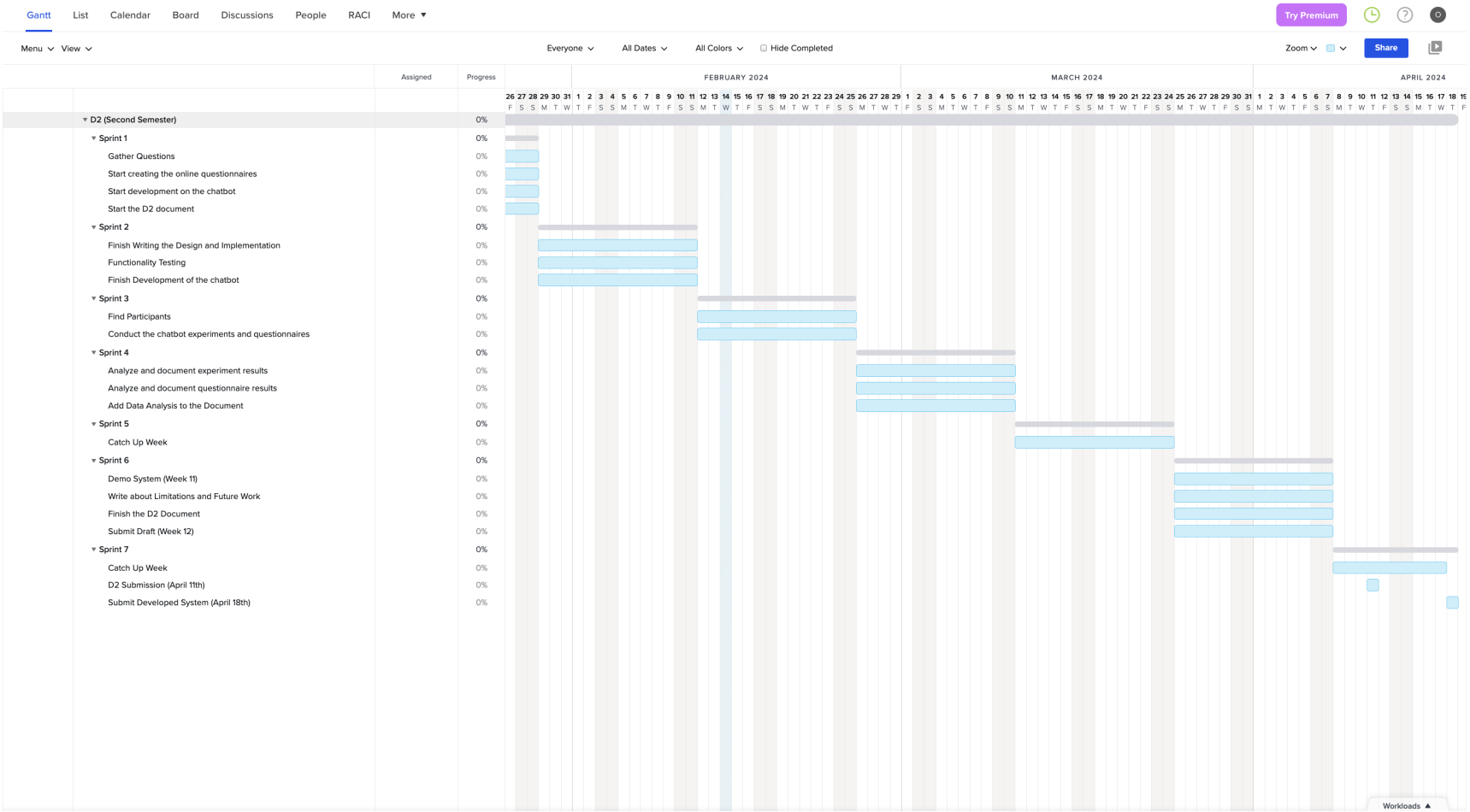

Figure A.2: Gantt Chart for Deliverable 1

Appendix B - Prototype

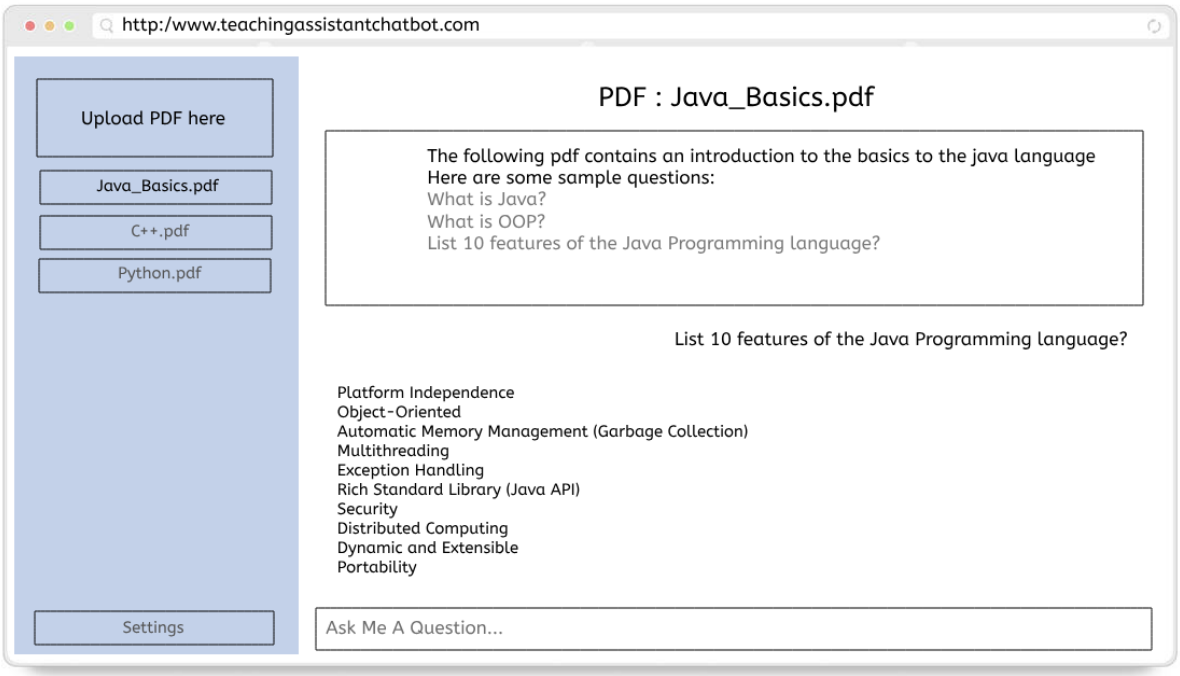

Figure B.1: The Chatbot Prototype created in Photoshop

.

Appendix C - Microsoft Forms

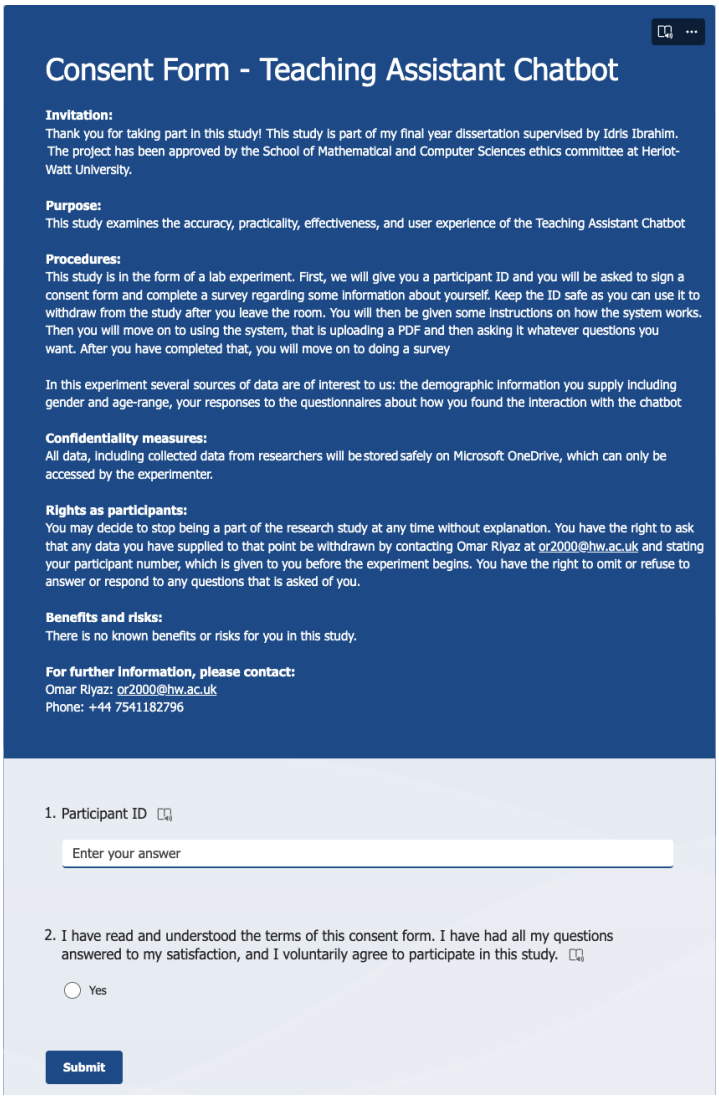

Figure C.1: The consent form created in Microsoft Forms

.

Figure C.2: The pre-experiment form created in Microsoft Forms

.

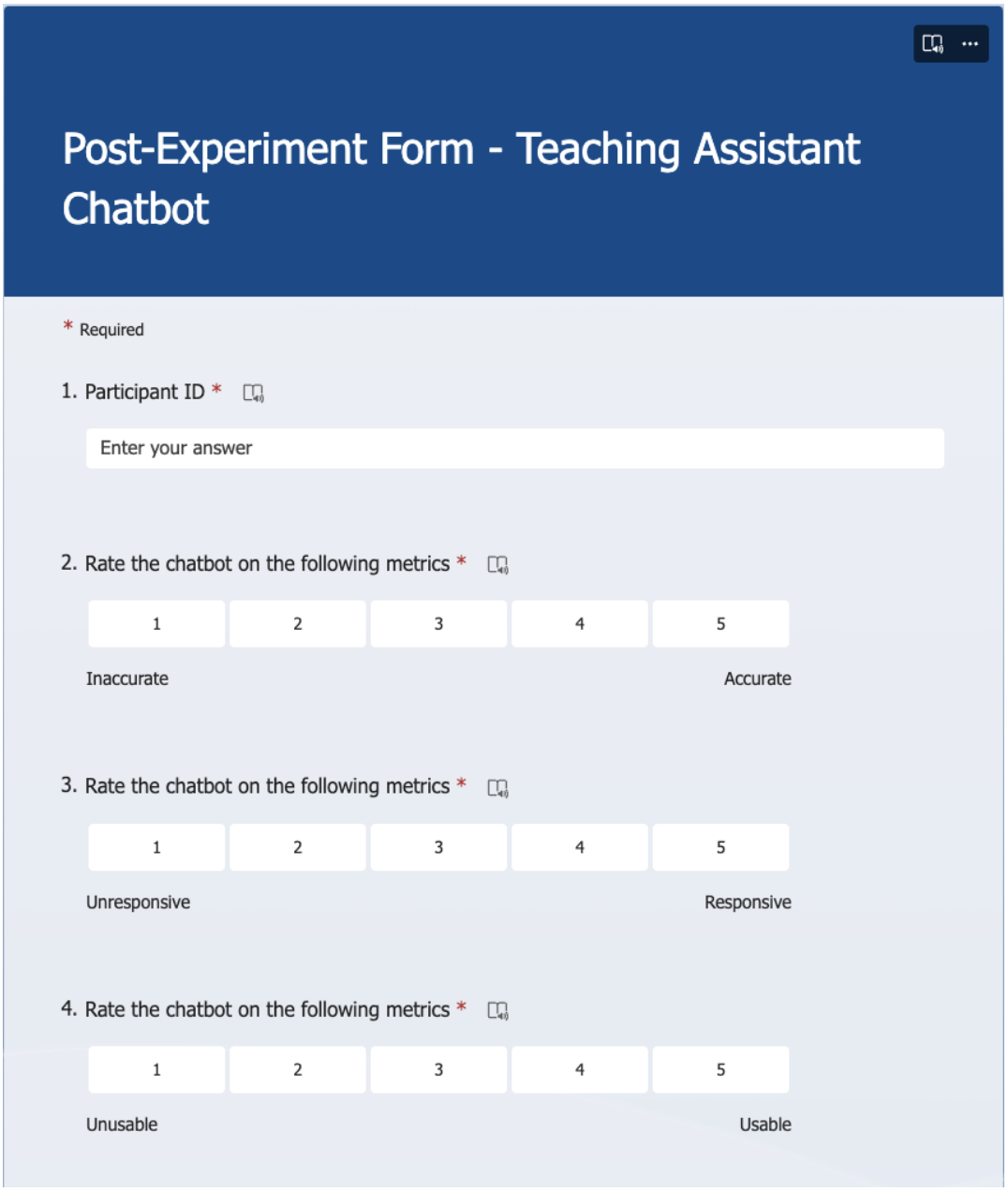

Figure C.3: The post-experiment form created in Microsoft Forms (1)

.

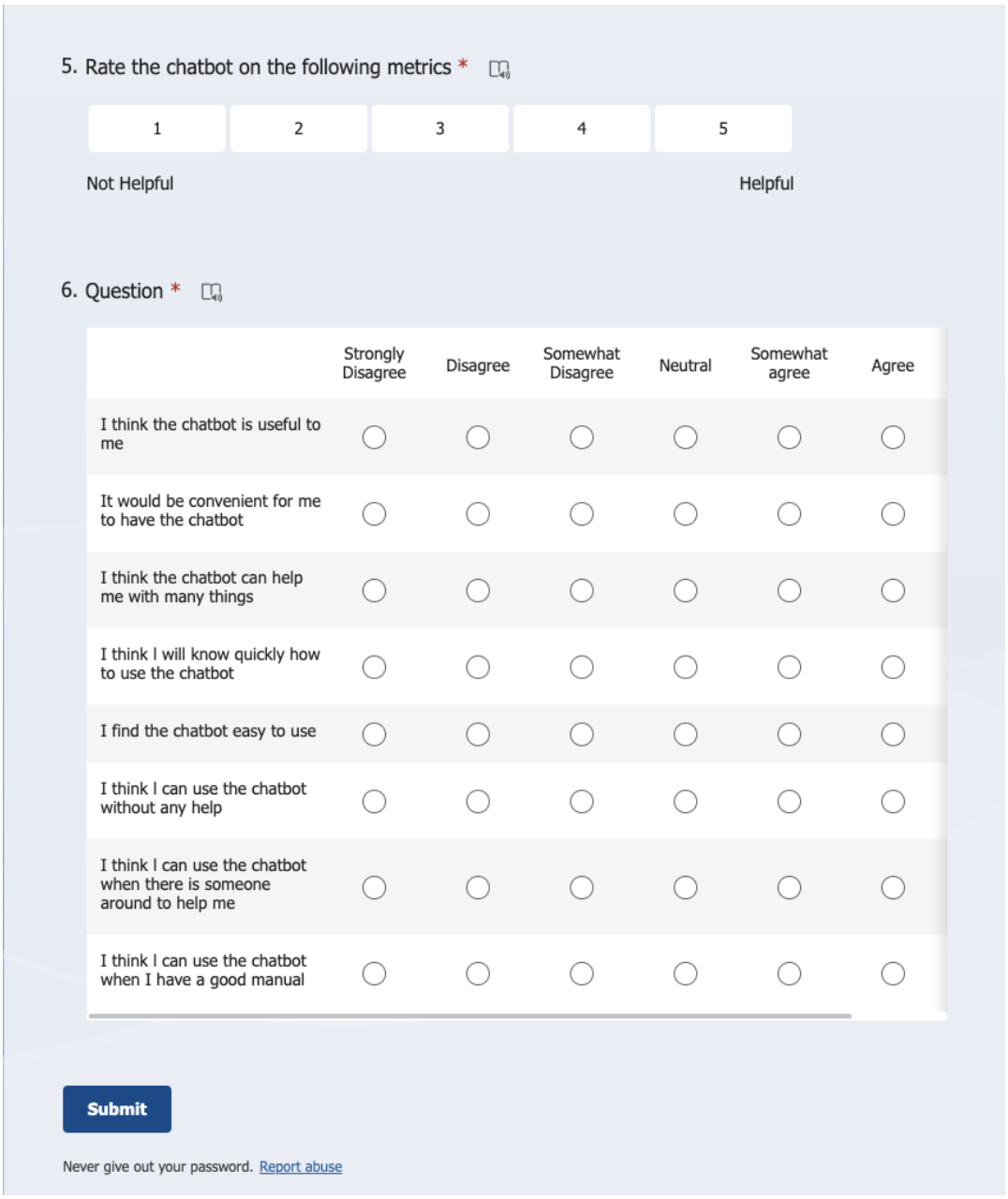

Figure C.4: The post-experiment form created in Microsoft Forms (2)

.

Appendix D - Literature Review References Table

ID	Reference	Description
1	[Bisser, 2021]	The paper provides an organized approach to using Microsoft Conversational AI to create chatbots. It offers useful insights and reusable code samples for successful development from planning to deployment.
2	[Vinita et al., 2022]	The paper examines the advantages and uses of chatbots, emphasizing their value in a range of industries, such as marketing, healthcare, and education.
3	[Lazarus, 2010]	The book is about the usefulness of NLP, written for people with no experience in NLP and want to dive in a little deeper into the topic.
4	[Aggarwal et al., 2023]	The paper offers a comprehensive analysis of chatbot technology, highlighting its natural language processing, sentiment analysis, reinforcement learning techniques, and flexibility.
5	[Smestad and Volden, 2019]	This study examines how user experience is affected by chatbot-user personality alignment, emphasizing the benefits of doing so while taking user group and context into account.
6	[Ali and Djalilian, 2023]	The usage of AI chatbots, such as OpenAI's ChatGPT in the creation of scholarly publications is covered in this paper.
7	[Ng et al., 2022]	The paper explores the diverse applications of chatbots, especially in handling socio-legal issues. It highlights how they have the ability to uphold moral principles in the field of AI while bridging social divides and having a positive impact.

Table D.1: Literature Review references table (1)

ID	Reference	Description
8	[Essel et al., 2022]	The study highlights the use of zero-coding techniques in constructing virtual teaching assistants, showing that students' academic performance was improved as compared to traditional teacher engagement.
9	[Network, 2023]	This study discusses how Chat GPT is changing education by influencing teaching tactics and providing thoughtful responses. Teachers are adopting it, reorganizing classroom participation, and utilizing it as a tool for teaching support.
10	[Song et al., 2022]	This paper shows that teaching simulation using student chatbots with different attitudes impacted preservice teachers' efficacy. Participants who taught an ordinary chatbot significantly increased their teacher efficacy levels.
11	[Abdelhamid and Katz, 2020]	The use of chatbots as teaching aides for first-year engineering students is examined in this study. Their mobile application combines online services, speech recognition, a cloud-based database, and AI into a chatbot to help students and offer real-time support.
12	[Kuhail et al., 2023]	The impact of educational chatbots on teaching and learning is revealed by a comprehensive review of thirty-six articles in the field in this paper.
13	[Chaves and Gerosa, 2021]	This study shows that in order to minimize user frustration and discontent during human-chatbot interactions, chatbots should display social qualities that are contextually appropriate.
14	[Scheider et al., 2023]	This study explores the significance of giving chatbots social characteristics that align with user expectations in order to improve their interacting experience.
15	[Tamayo, 2020]	The paper outlines situations in which robots could act as instructors, friends, or entities that evoke caring while also examining the ethical ramifications of introducing them into classrooms.

Table D.2: Literature Review references table (2)

Appendix E - Finished Website

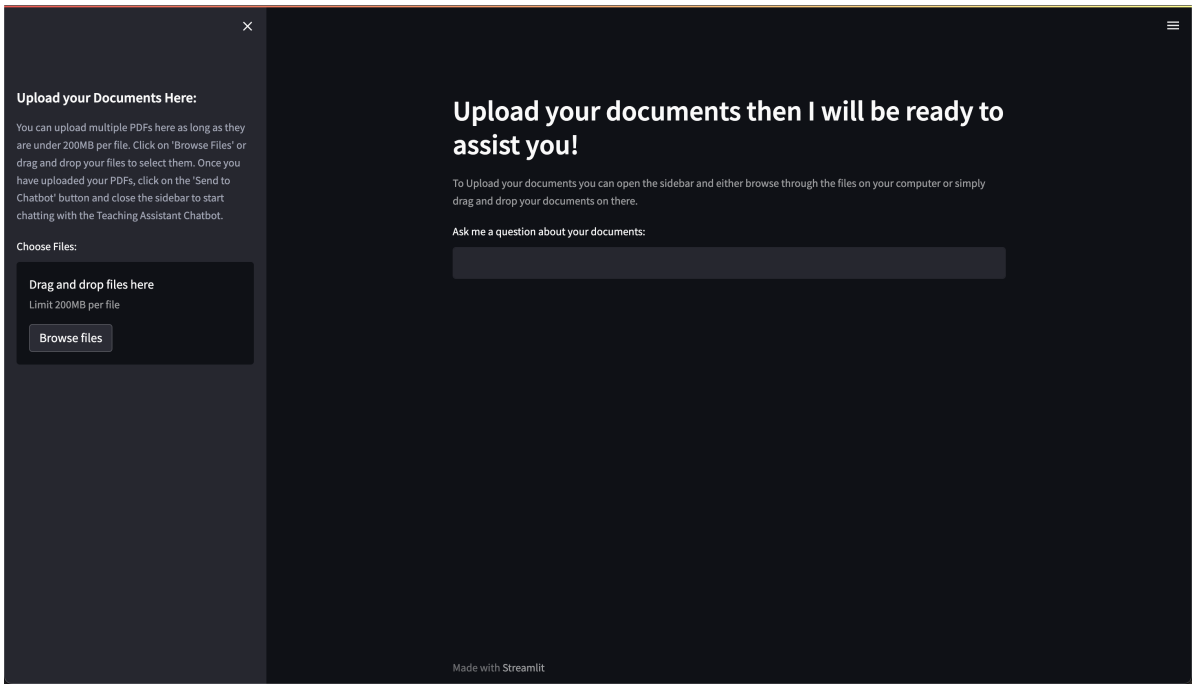

Figure E.1: The start page of the Teaching Assistant Chatbot

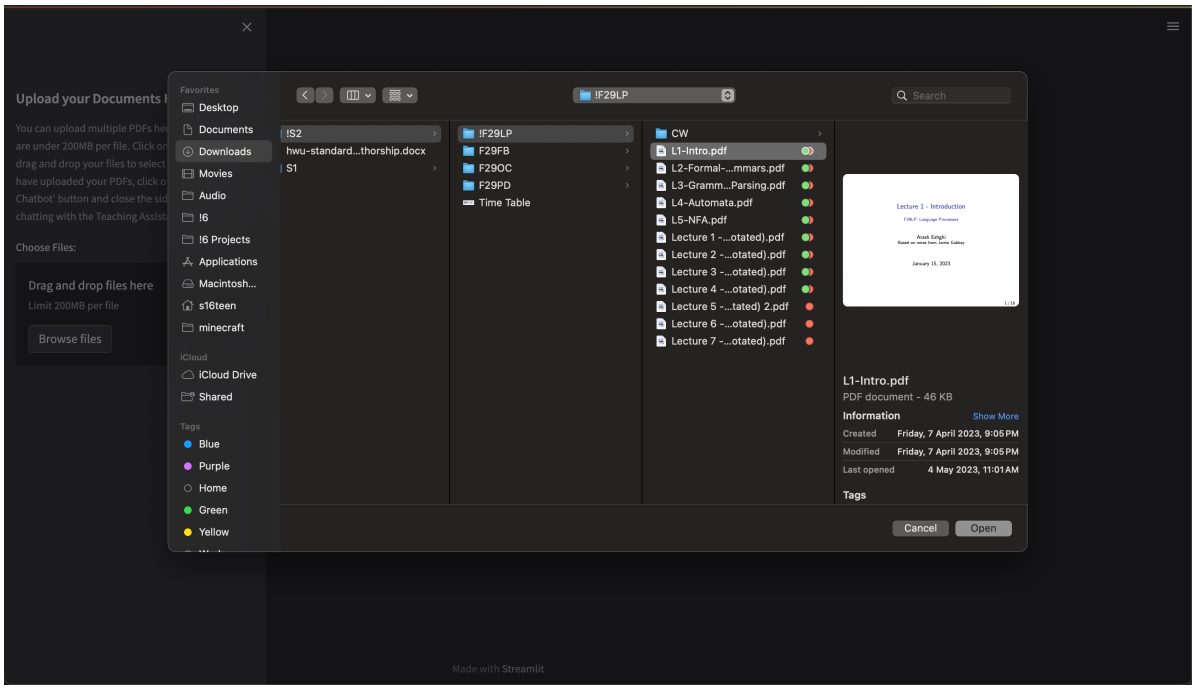

Figure E.2: Selecting PDF(s) to upload to the Teaching Assistant Chatbot

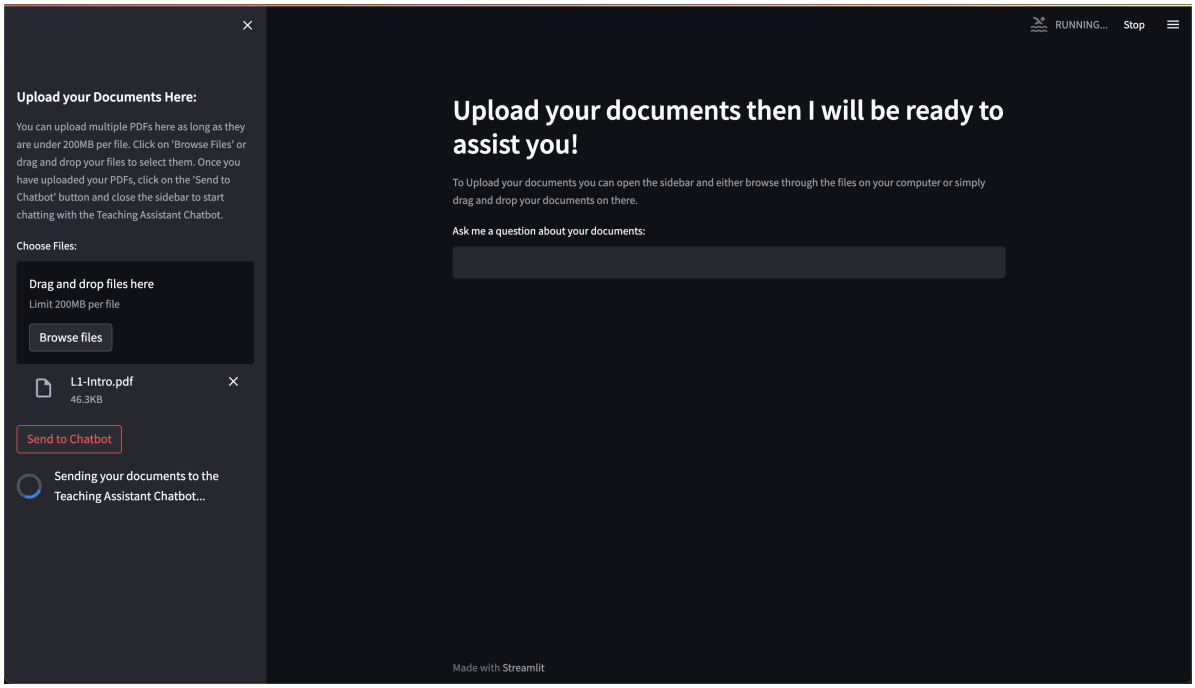

Figure E.3: The PDF being extracted and embedded in the Teaching Assistant Chatbot

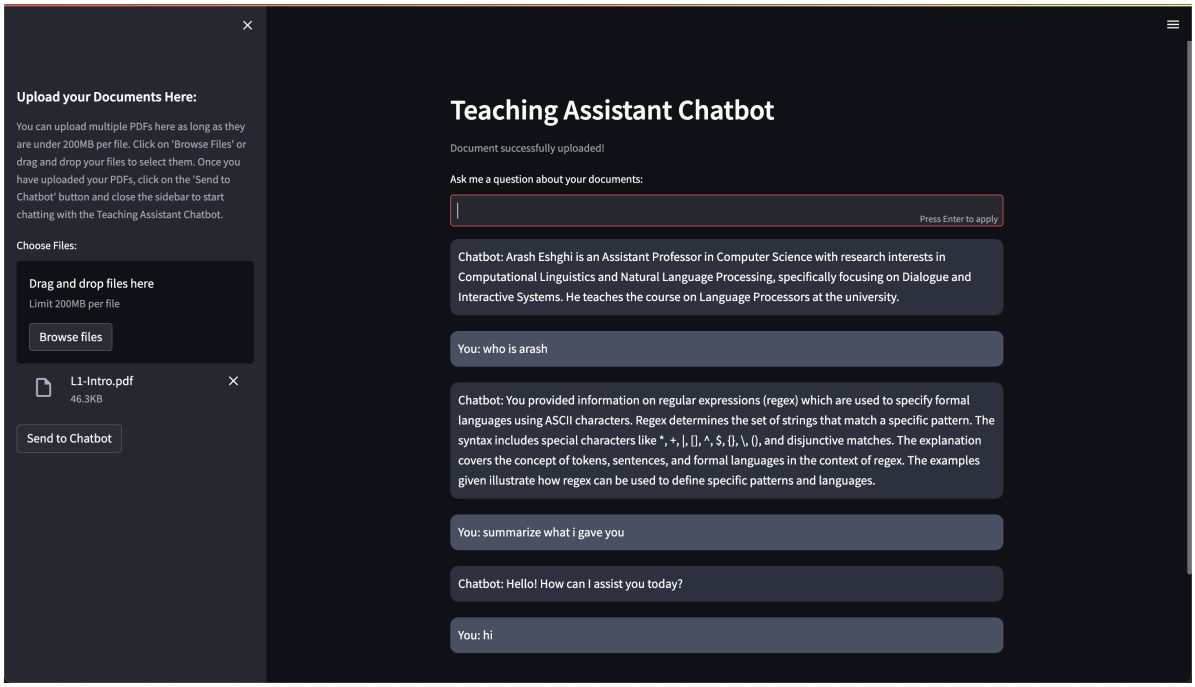

Figure E.4: The sample conversation of the Teaching Assistant Chatbot

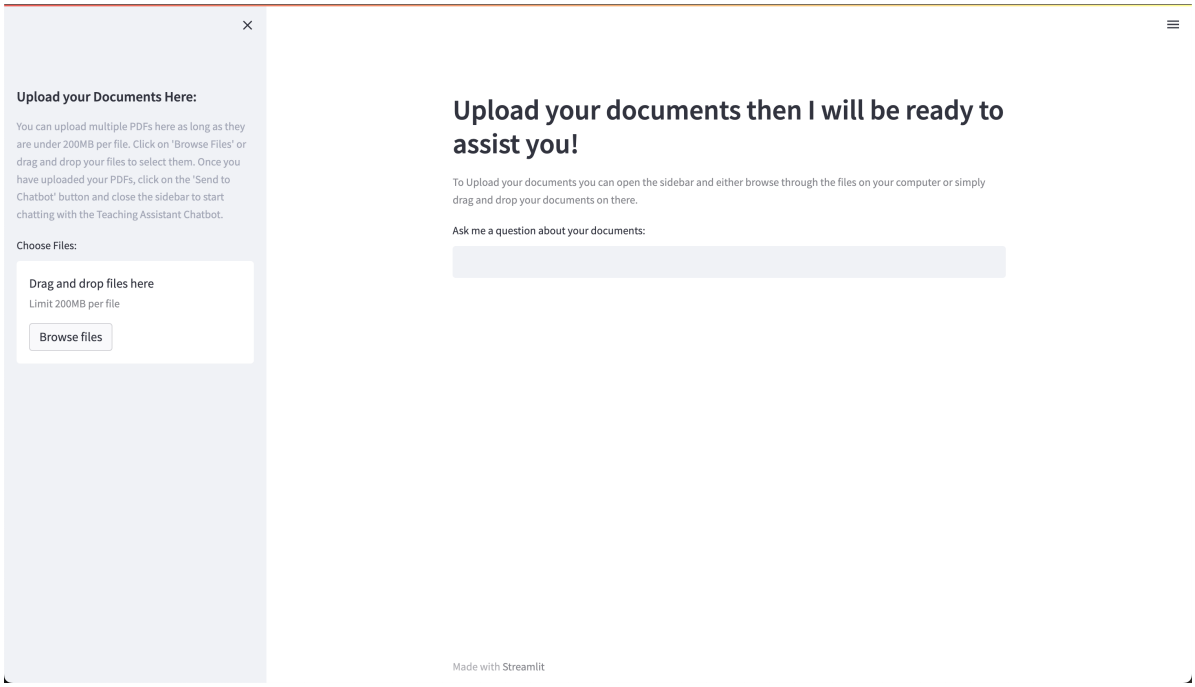

Figure E.5: The light mode of the Teaching Assistant Chatbot as changed in the settings