KFUEIT Smart Assistant

Session (Fall 2019- Spring 2023)

Bachelor of Studies in Software Engineering

Submitted by

Muhammad Qasim	SWEN-19111094
Muhammad Mubshir Asif	SWEN-19111169
Aman Ullah Khan	SWEN-19111152
Muhammad Dawood Mushtaq	SWEN-19111093

Supervised By

Engr. Hussain Mahmood Sargana

DECLARATION

It is hereby declared that this project, neither as a whole nor as a part has been copied out from any source. It is further declared that I developed this software and this report entirely on the basis of my personal efforts made under the sincere guidance of my project supervisor. If any part of this software is proved to be copied or found to be a report of some other, I standby the consequences. No portion of the work presented in this report has been submitted in support of any application from any other degree of qualification of any other university or institute of learning. I further declare that this software and all associated documents, reports and records are submitted as partial requirement for the degree of BSSE (Bachelor of Studies in Software Engineering). I therefore understand and transfer copyrights for these materials to Khwaja Fareed University of Engineering and Technology Rahim Yar Khan.

Reg No :	SWEN19111094	Reg No:	SWEN19111169
Name :	Muhammad Qasim	Name :	Muhammad Mubshir Asif
Signature:		Signature:	
Date :		Date :	
Reg No :	SWEN19111093	Reg No:	SWEN19111152
Name :	Muhammad Dawood Mushtaq	Name :	Aman Ullah khan
Signature:		Signature:	
Date :		Date :	

APPROVAL FOR SUBMISSION

This is certifying that we have read the project report titled "**KFUEIT Smart Assistant**" submitted by following Students of BSSE Khwaja Fareed University of Engineering and Technology.

It is our judgment that this project report is of sufficient standard to warrant its acceptance by the Institute of Computer and Software Engineering in Khwaja Fareed University of Engineering and Technology Rahim Yar Khan.

Approved by:

Signature: _____

Supervisor: Engr. Hussain Mahmood Sargana

Date : _____

Meeting Log

FYP-I Meeting Log

Institute of Computer & Software Engineering

Khwaja Fareed University of Engineering and Information Technology, Rahim Yar Khan

Project Name: KFUEIT Smart Assistant

Supervisor: Engr. Hussain Mahmood Sargana

Meeting No.	Date	Outcome	Signature Of Supervisor
1	24/10/2022	This is the first meeting about the idea of the project.	
2	4/11/2022	The meeting is about basic requirements of a project like a database, tools etc.	
3	21/11/2022	Draft of documentation is presented.	
4	5/12/2022	This meeting is about updates in the documentation.	
5	19/12/2022	This meeting is about diagrams of the project.	
6	9/01/2023	In this meeting, final documentation is reviewed.	

Meeting Log

FYP-2 Meeting log

Institute of Computer and Software Engineering

Khawaja Fareed University of Engineering and Information Technology

Rahim Yar Khan

Project Name: KFUEIT Smart Assistance

Supervisor: Engr. Hussain Mahmood Sargana

Meeting No.	Date	Outcome	Signature of Supervisor
1	22 / 1 / 2023	This is the first meeting about the implementation of project.	
2	3 / 2 / 2023	In this meeting Design of chatbot was discussed.	
3	18 / 2 / 2023	In this meeting Design of website shown.	
4	1 / 3 / 2023	In this meeting first phase of coding was shown.	
5	20 / 3 / 2023	In this meeting second phase of chatbot was shown.	
6	4 / 4 / 2023	In this meeting third phase of chatbot was shown.	
7	3 / 5 / 2023	In this meeting final phase of chatbot was shown.	

ACKNOWLEDGEMENT

We would like to express our special thanks and gratitude to our teachers as well our supervisor who gave us the golden opportunity to do this wonderful project on the topic **KFUEIT Smart Assistant**, which also helped us in doing a lot of research and we came to know about so many new things. Also, our first thanks also go to **Khwaja Fareed University OF Engineering and Information Technology** for designing such a worthy syllabus and making us to do this project. Secondly, we are very thankful to the faculty of the ICSE without whom our project would have been impossible. Especially our heartfelt thanks go to **Engr. Hussain Mahmood Sargana**, for our project supervisor who constantly supervised us throughout the project time period.

Dedication

We dedicate our dissertation work to our family, respected teachers and friends. A special feeling of gratitude to our loving parents.

TABLE OF CONTENTS

List of Figures

List of tables

Chapter 1
INTRODUCTION

1.1.Introduction

1.1.1 What is chatbot:

A computer program that can understand human language and communicate with a user via a website or messaging app is known as a chatbot (conversational interface, AI agent). Online jobs that chatbots can accomplish include simple question answering, call scheduling, and client feedback gathering. Bots are used by brands to streamline corporate operations, improve customer service, and cut support expenses.

1.1.2 The difference between chatbots and bots:

A chatbot is a conversational interface that speaks or texts to users. The bot, on the other hand, is a kind of software that automates repetitive tasks like looking up information and adding it to websites but does not engage in user-to-user communication.

1.1.3 What is AI chatbot:

Artificial intelligence (AI) chatbots are built on machine learning and Natural Language Processing (NLP). They are powered by data, which they use to freely respond to user questions. AI chatbots are capable of learning on their own and improving with each interaction with users.

1.1.4 What is Rule based chatbot:

AI agents that operate according to rules deliver answers that have already been predetermined. They can only appropriately respond to questions that they were designed to. Rule-based bots can't develop on their own.

1.2.Background

The first chatbot ever was developed by MIT professor Joseph Weizenbaum in the 1960s. It was called ELIZA. You'll read more about ELIZA and other popular chatbots that were developed in the second half of the 20th century later on.

In the year 2009, a company called *WeChat* in China created a more advanced Chatbot.

1.2.1 History of the chatbots:

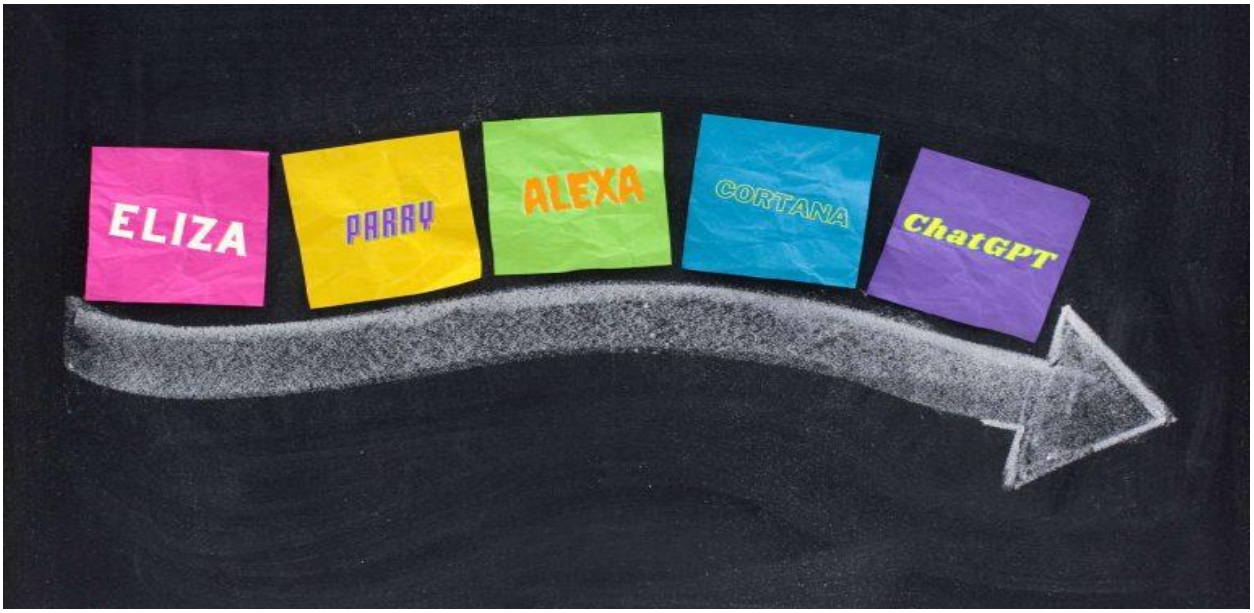


Figure 1: History of chatbots

ELIZA: [1]

As was previously mentioned, ELIZA was the first chatbot. It was developed in 1966 by Joseph Weizenbaum and simulates conversation using pattern matching and replacement techniques.

The software was created in a way that it resembles human speech. The words that users entered into a computer were passed on to the chatbot ELIZA, who subsequently matched them to a list of potentially prepared responses.

PARRY: [2]

American psychiatrist Kenneth Colby developed PARRY in 1972. A schizophrenia patient was portrayed by computer software. It makes an effort to mimic the illness. It is a program that uses natural language and simulates human thought.

The way PARRY operates is by a complex system of presumptions, attributions, and "emotional responses" brought on by shifting linguistic input weights. An adaptation of the Turing test was used to test PARRY to validate the research.

SIRI: [3]

Apple created Siri in 2010 for iOS; it is a smart personal assistant and learning navigator

with a natural language user interface. It set the stage for all subsequent AI bots and PAs.

A novel Apple service in which customers might contact Siri through Messages and have conversations with her is described in a patent application by the USPTO. The new patent is comparable to one that was published in the latter part of last year, but it now further integrates with audio, video, and image data. Apple's patent describes a Siri that may accomplish current tasks without the user needing to communicate aloud, much as other messaging and Facebook Messenger.

GOOGLE ASSISTANT: [4]

At Google Inch in 2012, Google Now was introduced. It provides information, offers recommendations, and carries out tasks by sending requests to several web services. It was a component of a set of UI improvements and enhancements for mobile search that also included a female-voiced personal assistant to take on Apple's Siri. The original purpose of Google Now was to provide contextually relevant information depending on the time of day and location. With a wide range of content categories given on cards, it expanded to become considerably more intricate and complex.

ALEXA: [5]

Amazon created Alexa, an intelligent personal assistant. Since its debut in 2014, it has been included in a variety of gadgets, including the Amazon Echo, Echo Dot, Echo Show, and more. Additionally, there is an Alexa app and more items made by independent manufacturers include Alexa. Alexa will assist you if you simply ask her to play some music or find you an Italian restaurant.

ChatGPT: [6]

The large-scale language model ChatGPT was developed by OpenAI. In 2021, the OpenAI team established it. It is intended to help users create text that appears human-like from the input. Several activities, including conversation generating and language translation, can be accomplished with ChatGPT.

The model can produce content that is frequently impossible to differentiate from language written by a human because it has been trained on a vast amount of data. ChatGPT has received appreciation for its capacity to produce language that sounds natural and for the range of uses, it may find. By the way, ChatGPT produced this abstract.

1.3.Aim of the project:

As we know Python is a suitable language for scriptwriters and developers. Let's write a script for KFUEIT Smart Assistant using Python. The query for the assistant can be manipulated as per the user's need.

The most important aim of this project is to develop such system that will help the students regarding giving specific commands. Also, develop such a system that will help the new students with any admission query.

1.4.Advantages of the project:

The advantages of KFUEIT Smart Assistant (AI Chatbot) is following

- The main advantage of a chatbot is developing a system that answers any query related to admission on the KFUEIT website.
- The chatbot act as a proof of concept in increasing usability and reducing learning curves for new users of the KFUEIT website.

1.5.Feasibility report of the project:

Before determining whether the system we must develop is feasible or not, we believe it is important to clarify what the term "feasibility" means. The feasibility of a system's development is a measure of how beneficial or practical it will be for the organization. It's a survey to help with the investigation of the system. Its goal is to provide information that will aid a more in-depth investigation in the future.

1.5.1 Economic feasibility:

The most important feasibility in this project is economical feasible. Economical feasible solution means that is this solution is costly effective for the customer? So, in our case, our project is AI based chatbot for KFUEIT which is not expensive too much.

1.5.2 Technical feasibility:

Today everyone is familiar with the use of any website chatbot. There is no need for any special skills required by the students or to make them aware of the website chatbot that how to use it. So, the project is technically feasible too.

Chapter 2
USER INTERFACE

2.1. User Interface: [7]

User interface design focuses on maximizing usability and user experience while creating user interfaces for various hardware and software, such as PCs, mobile devices, and web applications. User interface (UI) design in software development is the process of creating morally and aesthetically appealing interfaces.

Easy-to-use interfaces are what designers strive to create. Graphical interfaces and other types of user interface design are referred to as UI design. Making user interaction and experience as easy as possible is the aim of the user interface.

User Interface have different types

- Graphical User Interface (GUI)
- Command Line Interface (CLI)
- Menu-driven User Interface (MUI)
- Touch User Interface
- Voice User Interface (VUI)

Good user interface design facilitates finishing the task without drawing unnecessary attention to it. Graphic design and typography are used to support it, it also influences how the user performs certain interactions and improves the aesthetic appeal of the design; design aesthetics may enhance or distract users to use the functions of the interface. The design process must be balanced between technical and visual elements.

2.2. Student User Interface:

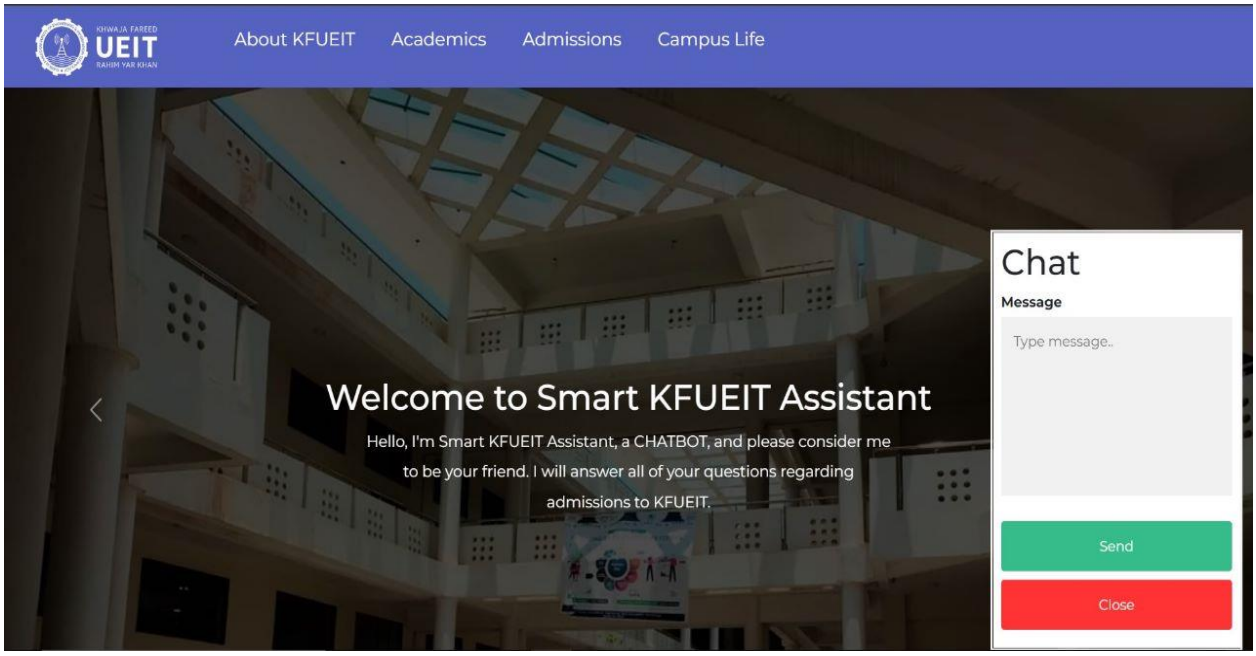
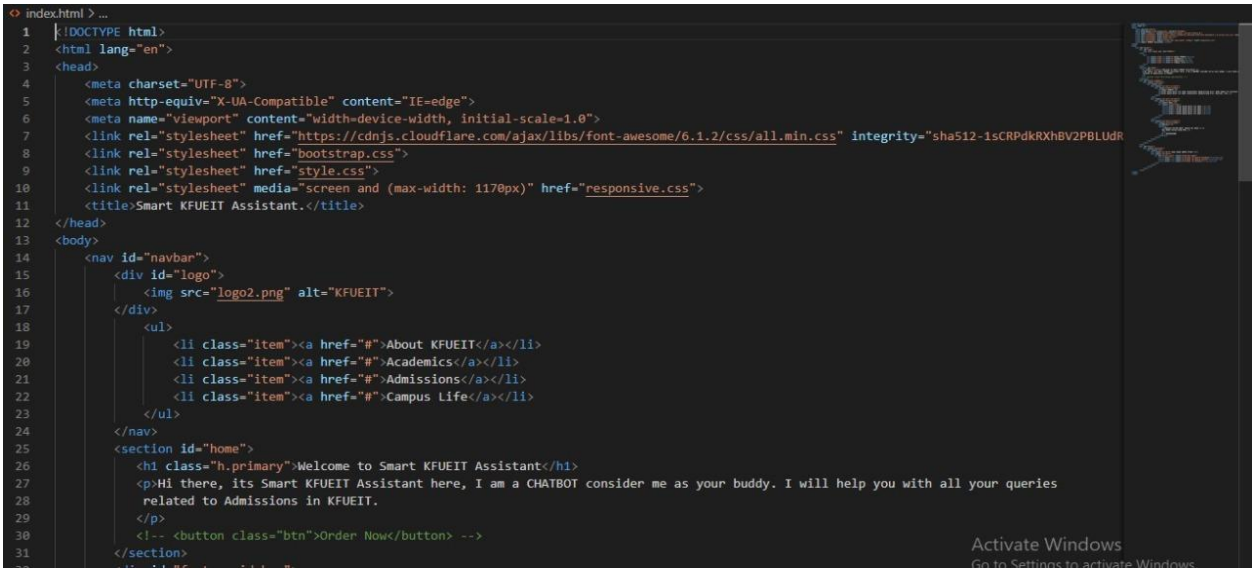


Figure 2: Student User Interface

we have designed a student user interface on the dummy website where we used different languages to make the dummy website and the user interface of our project chatbot.

Some of the coding screenshots are shown below:

HTML:



```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.1.2/css/all.min.css" integrity="sha512-1sCRPdkRXhBV2PBLUdR
    <link rel="stylesheet" href="bootstrap.css">
    <link rel="stylesheet" href="style.css">
    <link rel="stylesheet" media="screen and (max-width: 1170px)" href="responsive.css">
    <title>Smart KFUEIT Assistant.</title>
</head>
<body>
    <nav id="navbar">
        <div id="logo">
            <img src="logo2.png" alt="KFUEIT">
        </div>
            <ul>
                <li class="item"><a href="#">About KFUEIT</a></li>
                <li class="item"><a href="#">Academics</a></li>
                <li class="item"><a href="#">Admissions</a></li>
                <li class="item"><a href="#">Campus Life</a></li>
            </ul>
        </nav>
        <section id="home">
            <h1 class="h.primary">Welcome to Smart KFUEIT Assistant</h1>
            <p>Hi there, its Smart KFUEIT Assistant here, I am a CHATBOT consider me as your buddy. I will help you with all your queries
             related to Admissions in KFUEIT.
            </p>
            <!-- <button class="btn">Order Now</button> -->
        </section>
```

Figure 3:HTML

CSS:

```
54    }
55    #navbar ul li a:hover{
56      color: black;
57      background-color: white;
58    }
59
60    /* Home Section */
61    #home{
62      display: flex;
63      flex-direction: column;
64      padding: 3px 200px;
65      margin: auto;
66      height: 680px;
67      justify-content: center;
68      align-items: center;
69    }
70    #home ::before{
71      content: "";
72      position: absolute;
73      background: url('bg3.jpg')no-repeat center center/cover;
74      height: 750px;
75      top: 0%;
76      left: 0%;
77      width: 100%;
78      z-index: -1;
79      opacity: 0.89;
80
81    }
```

Figure 4:CSS

Chapter 3
METHODOLOGIES

3.1. Agile Methodology:

According to the agile model, each project must be handled uniquely and the current methodologies must be modified to best meet the project's goals. To deliver specific features for a release, tasks in Agile are separated into time boxes or brief periods.

Iterative approach is taken and working software build is delivered after each iteration. Each build is incremental in terms of features; the final build holds all the features required by the customer.

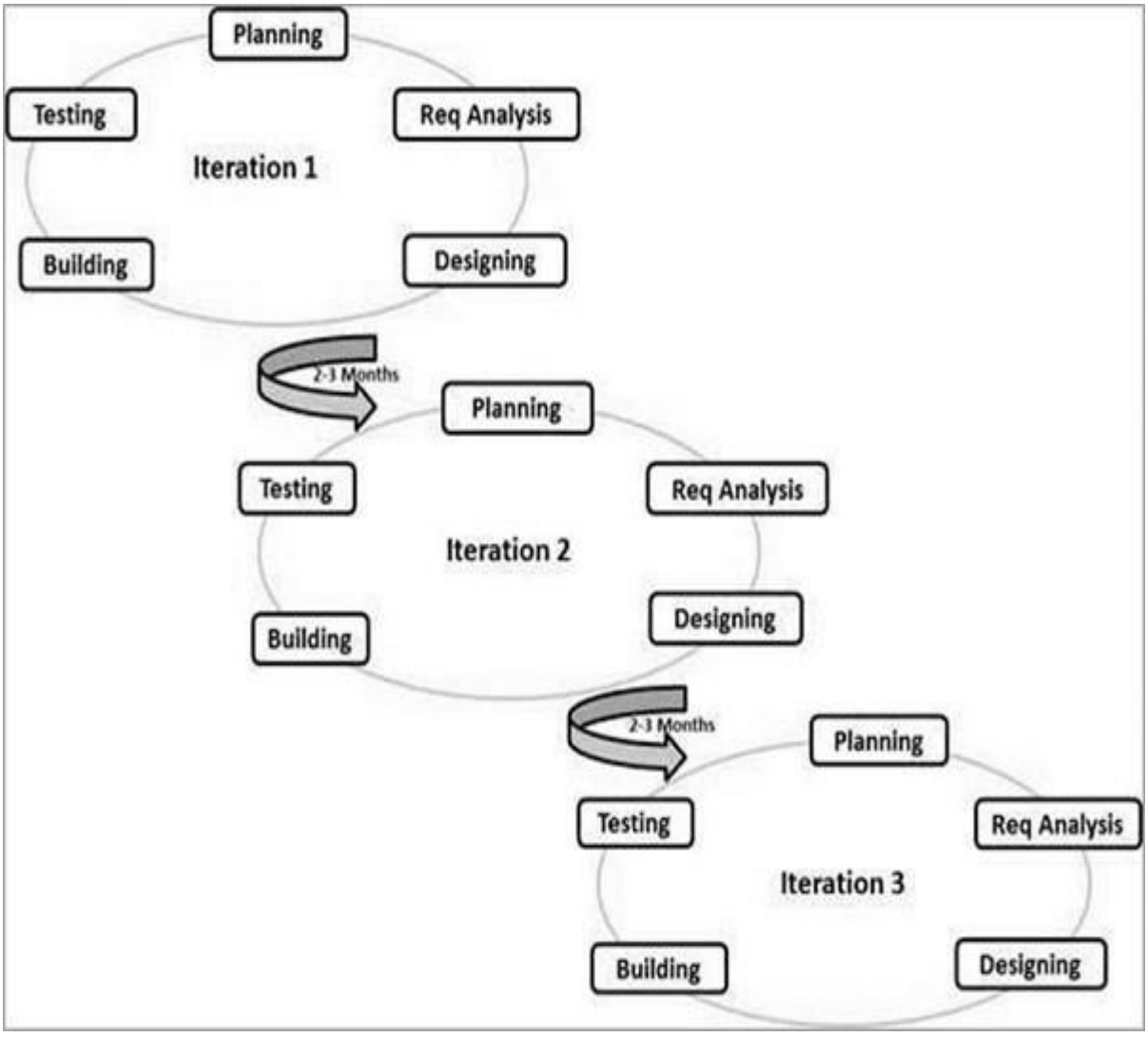


Figure 5: Agile Methodology

3.1.1 Advantages of Agile model:

1. It is a method of software development that is incredibly realistic.
2. encourages collaboration and cross-training.
3. Functionality can be quickly built and proven.
4. Minimum resources are needed.
5. Adaptable to changing or stable needs
6. early delivery of partially operational solutions.
7. a good model for continuously changing surroundings.
8. Simple to use documentation and few restrictions.
9. Concurrent development and delivery are possible.
10. Little to no planning is necessary.
11. simple to control.
12. gives developers flexibility.

3.1.2 Disadvantages of Agile model:

- Cannot be used to manage complex dependencies.
- More vulnerability to extensibility, maintainability, and sustainability.
- Without a comprehensive plan, an agile leader, and an agile PM practice, it will not succeed.
- Since virtually little documentation is produced, there is a high degree of individual dependency.
- Due to a lack of documentation, transferring technology to new team members may be fairly difficult.

3.1.3 What cause Agile models to fail?

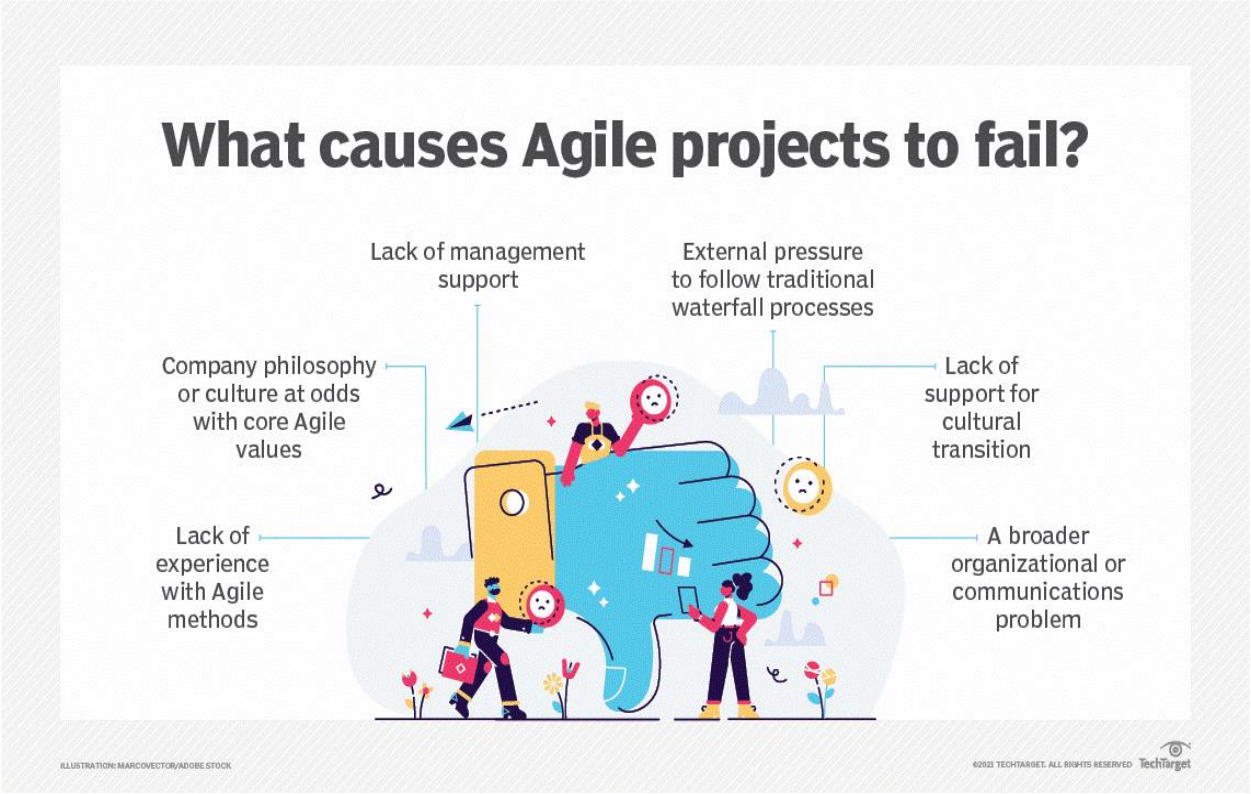


Figure 6: Agile projects fail

3.2. Agile model for AI based chatbot:

- **Define the project goals:** Define the project goals and objectives, including the intended audience, desired outcomes, and key performance indicators.
- **Identify user personas:** Identify the target audience and create user personas, including their needs, preferences, and pain points.
- **Create a backlog:** Create a backlog of features and functionalities for the chatbot, including natural language processing, machine learning algorithms, and the ability to understand user intent.
- **Plan the sprints:** Break down the backlog into smaller tasks and plan sprints. Define the scope of each sprint, timeline, and team members responsible for each task.
- **Develop a prototype:** Develop a working prototype of the chatbot that includes basic features and functionalities, such as the ability to answer user questions,

provide educational content, and personalize responses based on user input.

- **Test and iterate:** Test the chatbot with a small group of users and iterate based on feedback. Continuously improve the chatbot's accuracy, efficiency, and effectiveness.
- **Deploy and monitor:** Deploy the chatbot to a larger group of users and monitor its performance. Use analytics to track user interactions and identify areas for improvement.
- **Maintain and update:** Maintain the chatbot by fixing bugs, updating the content, and improving the user experience. Update the chatbot's algorithms and data models to improve accuracy and efficiency.
- **Continuously improve:** Continuously improve the chatbot based on feedback and changing user needs. Use A/B testing and experimentation to test new features and functionalities.

3.3. Student Database Design:

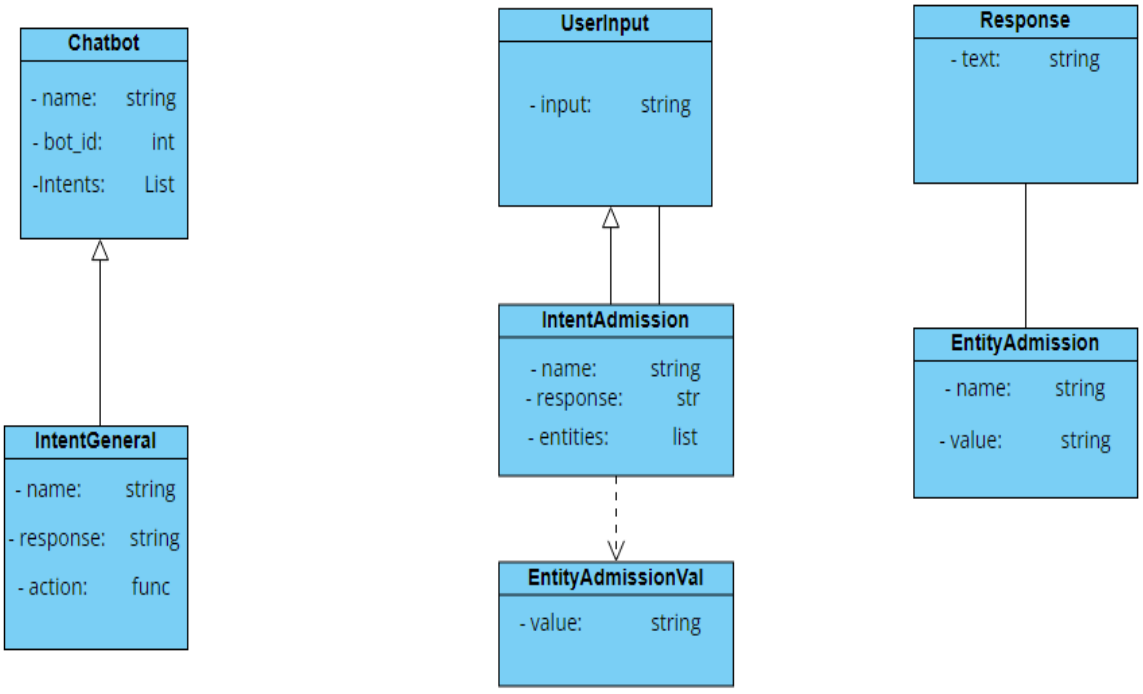


Figure 7: Student Database Design

3.3.1 What students can ask from the KFUEIT Smart Assistant:

KFUEIT Smart Assistant can assist students with a wide range of inquiries related to university websites. Here are some examples:

- **Admissions:**

Students can inquire with me about the admission criteria, deadlines, and procedures for admission. KFUEIT Smart Assistant can give you information about the application forms, transcripts, test results, and other admissions requirements documents.

- **Program and Courses:**

Students can ask questions about the various programs and courses the university offers. They might enquire about the course outline, timetable, requirements, and level of difficulty.

- **Financial Aids:**

KFUEIT Smart Assistant can help students find information on scholarships, grants, and other forms of financial aid. They can ask about the eligibility requirements, application procedures, and deadlines.

- **Academic Supports:**

Students can ask me about academic resources such as tutoring, study groups, academic advising, and library services.

- **Carrer Services:**

KFUEIT Smart Assistant can provide information on career services offered by the university, such as job search resources, resume-building workshops, and networking events.

Chapter 4
SYSTEM ANALYSIS

4.1. Feasibility Study:

As the name implies, a feasibility study seeks to determine the viability of a project or proposal. It is a determination of how feasible a project or idea is.

Any project or plan's early design phase includes a feasibility assessment. It is carried out to unbiasedly identify the advantages and disadvantages of a potential venture or an established company. The resources needed for the project, the likelihood of success, and the opportunities and hazards existing in the surrounding environment can all be identified and evaluated.

We are going to develop the new chatbot system for KFUEIT which is feasible. As our system is helpful for the student and easy to use for the student.

4.2. Market Feasibility Study:

This type of study assesses the market potential for the proposed project or business idea by examining market demand, competition, and market trends. It assesses the industry type, the existing marketing characteristics and improvements to make it better, the growth evident and needed, competitive environment of the company's products and services.

Our project Market Feasibility:

The first step in determining feasibility is to assess the market demand for the chatbot. This can be done by analyzing the number of prospective students who visit the university's website, the number of inquiries received, and the types of questions asked. If there is a significant demand for information and assistance related to the admission process, then a chatbot would be a viable solution.

4.3. Technical Feasibility Study:

A technical feasibility assessment looks at whether your company has the technical resources and expertise necessary to complete the project. The main goal of technical analysis is to determine whether your company has the capabilities required to carry out a project, including the manufacturing capacity, facility requirements,

raw materials, supply chain, and other inputs.

Our project Technical Feasibility:

The KFUEIT Smart Assistant will require the integration of natural language processing technology and may require custom development to integrate with the university's existing admission system. This will require technical expertise and resources. It is important to ensure that the university has the necessary technical expertise and resources to maintain the KFUEIT Smart Assistant.

4.4. Financial Feasibility Study:

This type of study assesses the financial feasibility of a project or business idea by analyzing the cost and benefits of the proposed project or business idea.

Our Project Financial Feasibility: Developing and maintaining a chatbot requires a significant investment. The university will need to allocate resources for the development of the chatbot, including technical development, training data acquisition, and ongoing maintenance. It is important to evaluate the financial feasibility of the project to ensure that the benefits outweigh the costs.

4.5. Operational Feasibility Study:

This type of study analyzes the practicality and effectiveness of implementing the project or business idea in term of operational processes, including staffing, equipment, and other resources needed.

Our Project Operational Feasibility: The chatbot must be designed to fit the university's operational processes and integrate with existing systems. It is important to assess whether the university has the necessary operational resources and processes to support the chatbot, such as providing training for staff who will manage and maintain the chatbot.

4.6. Legal Feasibility Study:

This type of study examines the legal aspects of the proposed project or business idea, including legal requirements, compliance with regulations, and potential risks.

Our Project Legal Feasibility: The chatbot must comply with applicable laws and regulations related to data privacy, security, and accessibility. We must ensure that the chatbot is compliant with all legal requirements.

Chapter 5
TESTING

5.1. Testing:

Testing is done in order to eliminate any bugs that could cause any malfunction in the system. Therefore, it is key to building successful systems. Testing of this project is conducted by the developers of the project. While testing the project, errors were fixed on spot. After developing the proposed chatbot for KFUEIT named as "**KFUEIT Smart Assistant**", We have covered all the software testing steps to check whether our project meet with the expected requirements or not. So, our project has **no bugs, no error** and also **working properly** in all tested steps.

5.2. Module Testing:

Module testing, commonly referred to as unit testing, is a sort of software testing in which individual software system components or units are examined independently of one another. Assuring that each module or component of the software system operates as intended and satisfies its functional criteria is the goal of module testing. It includes identifying the module, create test case, execute tests etc.

Project Module Testing:

Here are some steps that include in module testing phase in our project.

- **Identify the Module:** Describe the various components of the chatbot or Smart Assistant. This might consist of modules for welcoming the user, taking care of user input, providing details regarding admission standards and application deadlines, and assisting the user with the application process.
- **Create Test Case:** For every module, create test cases that cover all possible scenarios and edge cases. Providing both legal and invalid input, testing the Smart Assistant capacity to identify and reply to misspelled words, and testing the Smart Assistant capacity to handle numerous inputs in a single message are a few examples of test cases for the module that handles user input.
- **Prepare Test Data:** Gather test information that can be used to evaluate the modules. The test data should be representative of the types of input that the chatbot is designed to handle.

- **Execute Test Data:** Utilize the prepared test data to run the test cases. This will make it easier to spot any flaws or problems with the chatbot's modules.
- **Fix defects:** Fix any defects or issues that were identified during testing.

5.3. Black Box Testing:

Black box testing is a kind of software testing that focuses on evaluating a software application's functioning without being aware of its underlying operations or implementation details. The tester does not have access to the system's internal code; instead, all that is available to him are the input and output of the system. In black box testing, the tester sends several inputs into the system and then watches to see what happens. This form of testing's primary goal is to find flaws or problems in the system's functionality and make sure it complies with the requirements and standards.

Project Black Box Testing:

When performing black box testing for an KFUEIT Smart Assistant, the following steps can be taken:

- **Define the expected behavior of the Smart Assistant:** It is crucial to specify the desired behavior and performance of the Smart Assistant before testing it. Understanding the Assistant intended use, the kinds of queries it is made to answer, and the degree of accuracy necessary in its responses are all part of this.
- **Identify test cases:** Make a set of test cases that covers every scenario imaginable based on the desired behavior. These test cases need to contain common inquiries that candidates might make during the admissions process.
- **Execute the test cases:** Use the test cases to interact with the Smart Assistant and observe its behavior. Record the responses to each test case and compare them to the expected behavior.
- **Identify defects:** Find any flaws or differences between the chatbot's real behavior and the expected behavior.
- **Report defects:** Any flaws found during testing should be documented and reported to the development team. Describe the test case, the actual behavior seen, and the expected behavior in great detail.

5.4. Test Cases for KFUEIT Smart Assistant:

Following test cases are used to find the bugs and error.

Course Available in University:

Test case title	Courses Available in University
Use Case Id	01
Question	Which Courses are available in the university?
Expected Result	The smart Assistant shows all the courses available in the university for undergraduate programs, MS programs and PHD.
Real Result	It shows all the courses available in the university for undergraduate programs, MS programs and PHD.
Test Result	Successful

Table 1: Course availability

Admission requirements for undergraduate program:

Test case title	Admission requirements for undergraduate program:
Use Case Id	02
Question	What are the admission requirements for undergraduate program?
Expected Result	The smart Assistant shows all the requirements required for the undergraduate programs.

Real Result	It shows all the requirements required for the undergraduate programs.
Test Result	Successful

Table 2: Admission Requirements

Deadline for graduate programs:

Test case title	Deadline for graduate programs
Use Case Id	03
Question	What is the deadline to apply for the graduate programs?
Expected Result	The smart Assistant shows all the deadline date for each program that include in graduate program.
Real Result	It shows all the deadline date for each program that include in graduate program.
Test Result	Successful

Table 3: Deadline for graduate program 1

List of degree for fall semester:

Test case title	List of degree for fall semester
Use Case Id	04

Question	Which degree program are included in the fall semester?
Expected Result	The smart Assistant shows the list of all the degree programs that are conducted by the university.
Real Result	It shows the list of all the degree programs that are conducted by the university.
Test Result	Successful

Table 4: List of Degree

Fee Structure:

Test case title	Fee Structure
Use Case Id	05
Question	What is the fee structure for undergraduate fall semester degree program?
Expected Result	The smart Assistant shows the fee structure of all the degree programs that are conducted by the university.
Real Result	It shows the fee structure of all the degree programs that are conducted by the university.
Test Result	Successful

Table 5: Fee Structure 1

Eligibility Criteria:

Test case title	Eligibility Criteria
Use Case Id	06
Question	What are the eligibility criteria for applying to a BS degree program?
Expected Result	The smart Assistant shows the eligibility criteria of all the BS degree programs that are conducted by the university.
Real Result	It shows the eligibility criteria of all the BS degree programs that are conducted by the university.
Test Result	Successful

Table 6: Eligibility Criteria 1

5.5. Branch Testing:

Branch testing is a form of software testing method where the various control flow branches of a program are examined to make sure they function as expected. A program's conditional expressions (if-then-else), loops, and jumps often dictate the sequence in which its instructions are executed, which is referred to as control flow.

In branch testing, the tester develops test cases that put each potential control flow branch to the test. For instance, the tester would develop test cases that assess the program's behavior both when the condition is true and when it is false if the program contained an if-then-else statement.

Project Branch Testing:

In the branch testing, each outcome from a code module is tested as if the outcomes are binary, you need to test both True and False outcomes. Our project “KFUEIT Smart Assistant” pass the branch test.

Chapter 6
RESULTS AND CONCLUSIONS

6.1. Results:

6.1.1 Visual Studio Code:

The Project Smart KFUEIT Assistant developed in visual studio code with the use of python language.

The Screenshots of visual studio code are following:

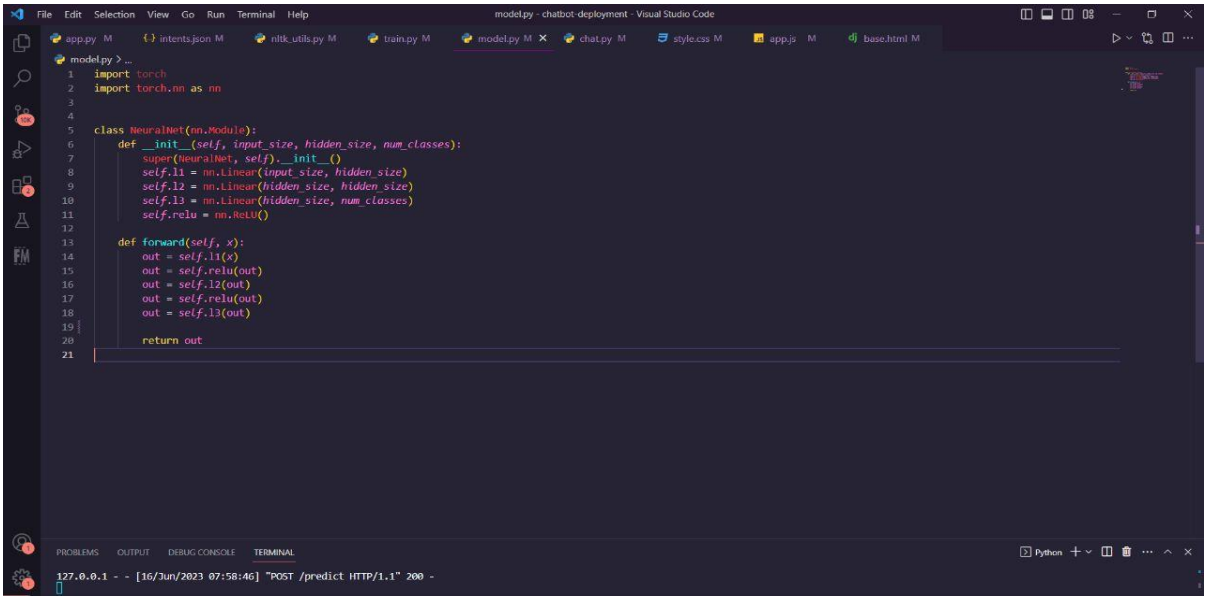


Figure 7: deep learning module

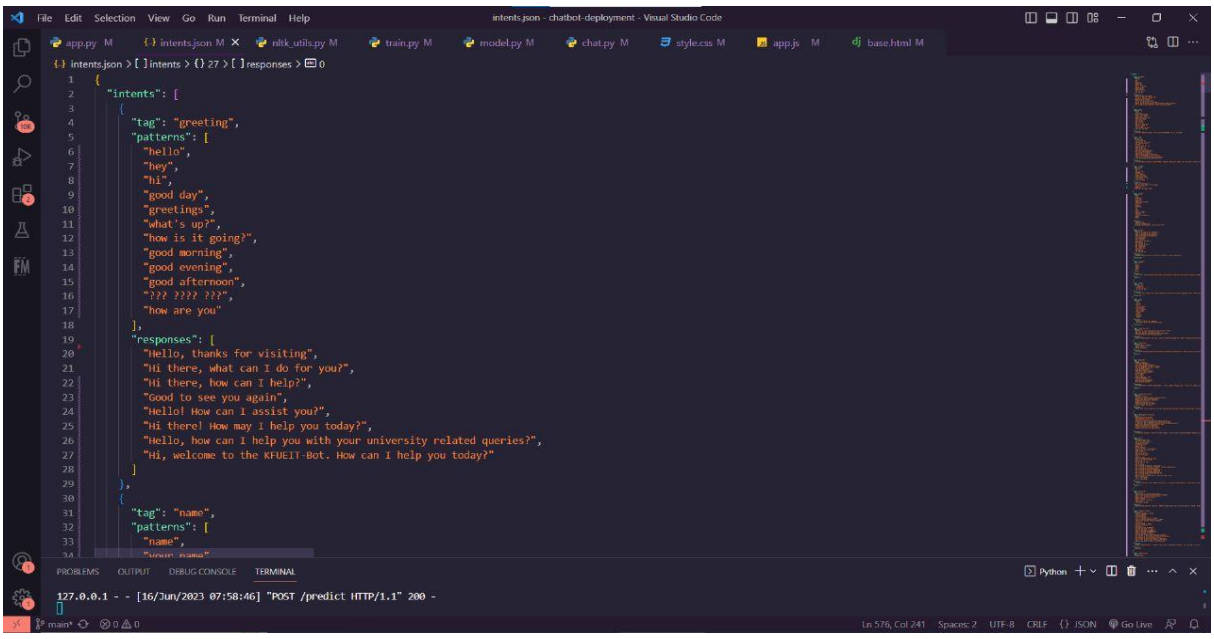


Figure 8: Intents

6.1.2 Chatbot Screenshots:

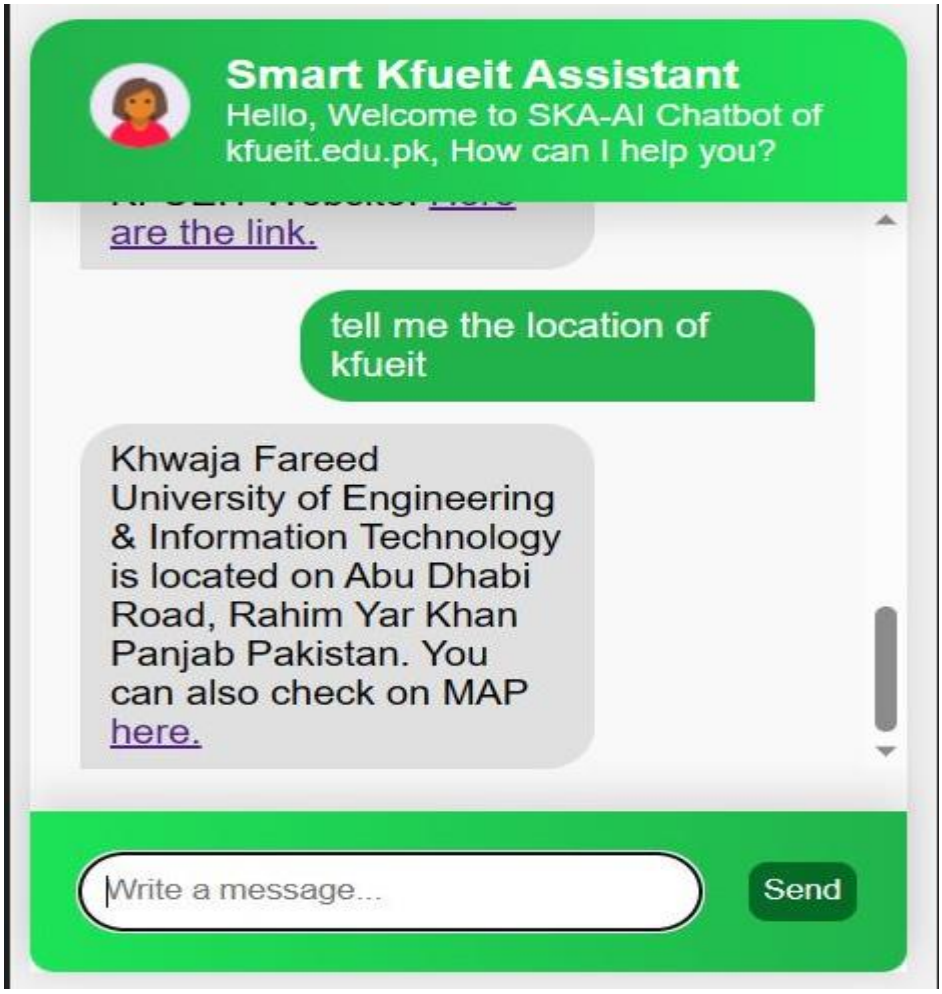


Figure 9: Location

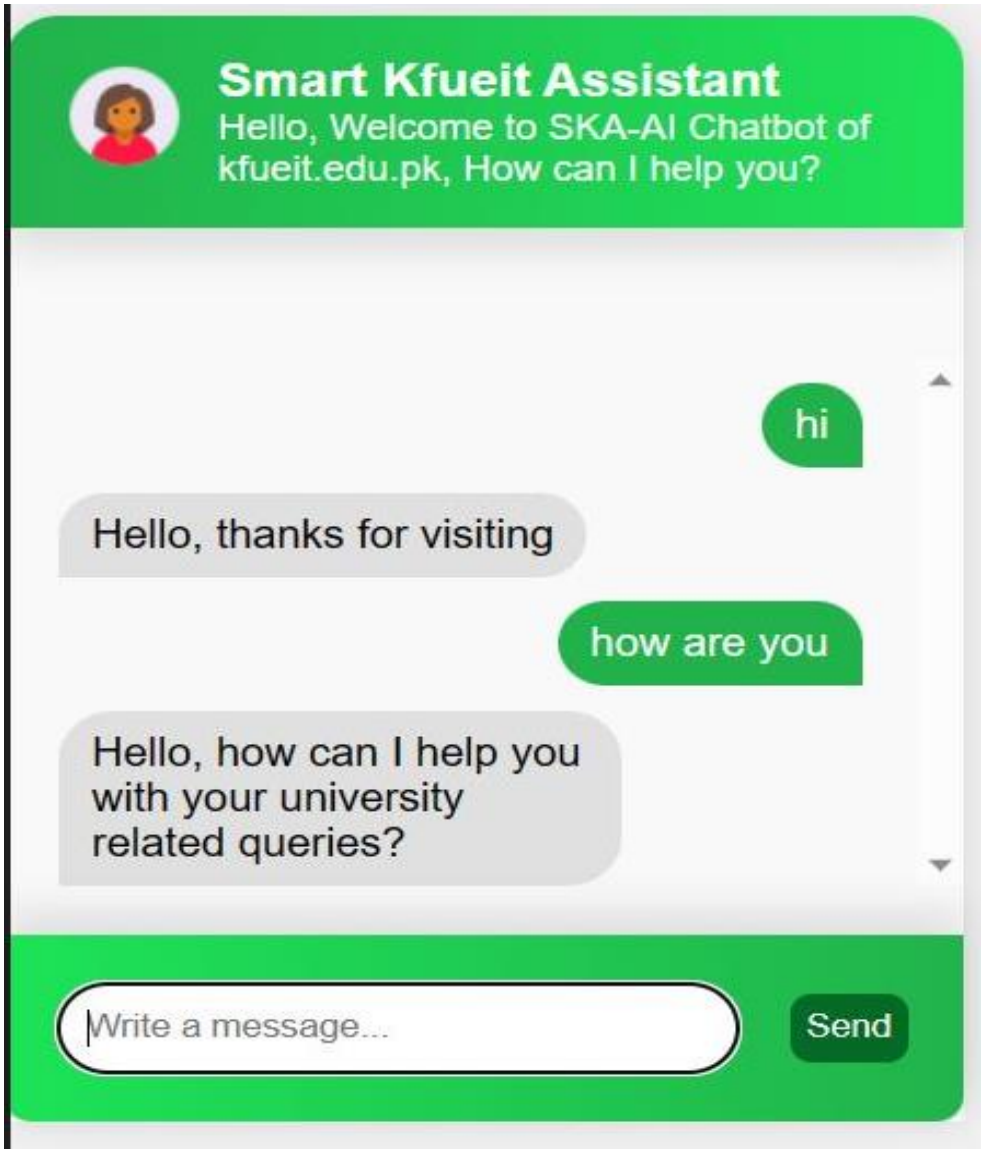


Figure 10: Hello Chat

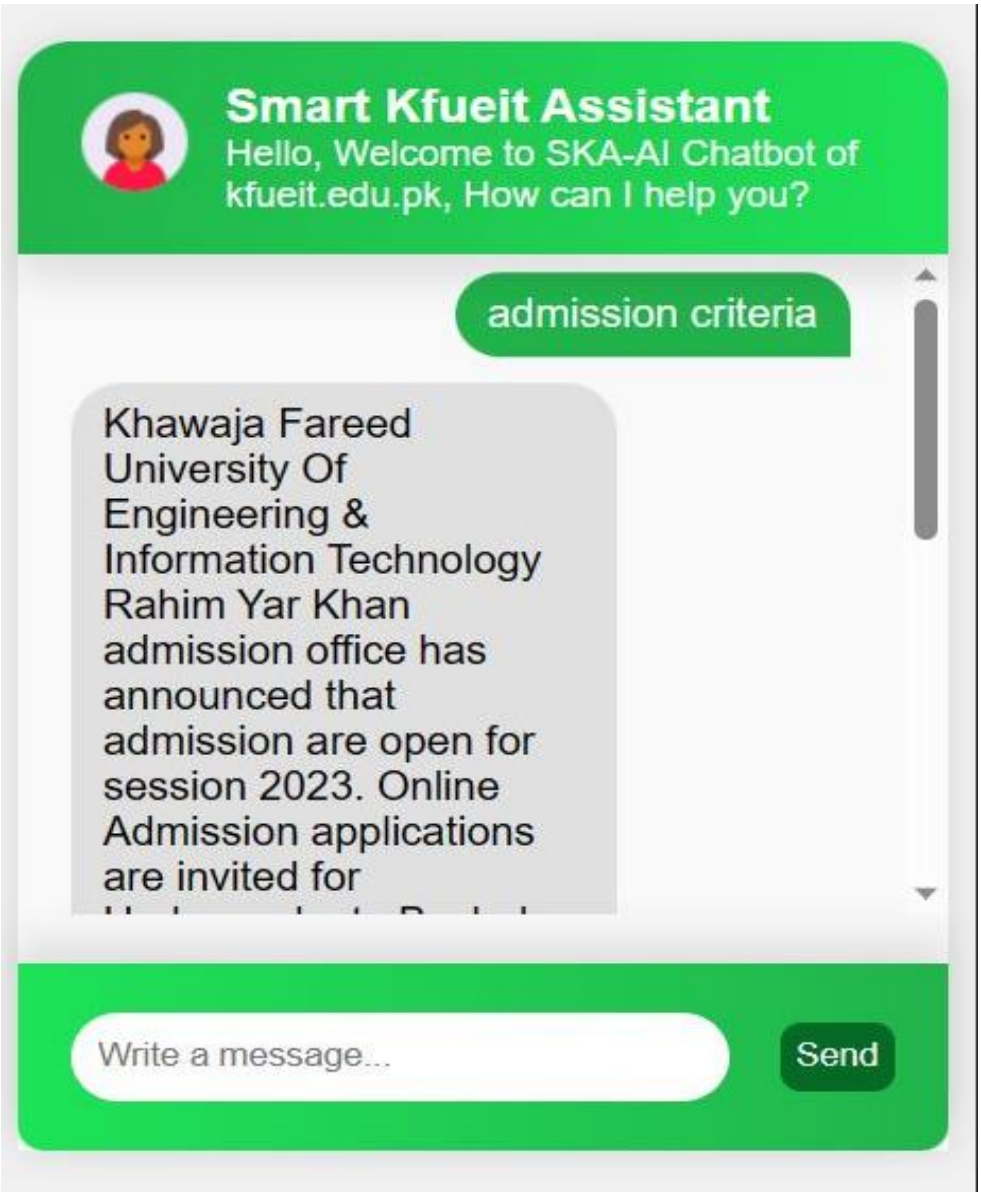


Figure 11: Admission criteria

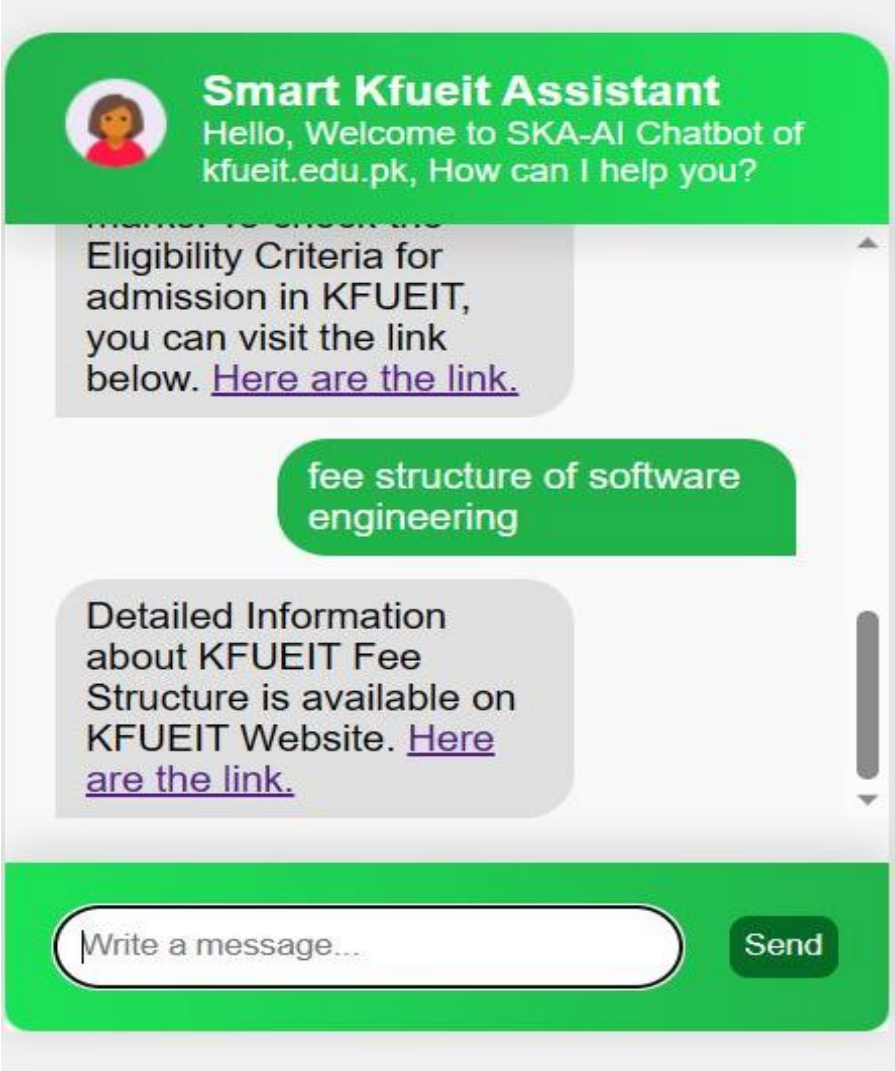


Figure 12: Fee Structure

6.2. Conclusion:

First of all, we introduce chatbots, their differences, AI and rule-based chatbots, and their history. It highlights the creation of ELIZA and the rise of popular chatbots like Siri, Google Assistant, and Alexa. The goal is to create a smart assistant chatbot for KFUEIT to assist students with admissions inquiries. Then we discussed the necessity of maximizing usability and user experience of user interface design. A student user interface for the chatbot was shown along with screenshots of the HTML and CSS coding after several forms of user interfaces were explored. Then Studies on market feasibility, technological viability, financial viability, and operational viability were all done. Both the technical know-how needed and the market demand for the chatbot were

assessed. Additionally taken into account were the operational procedures and financial investment. In summary, we gave an introduction to the idea of chatbots, their history of development, the purpose and benefits of the KFUEIT Smart Assistant project, the design of the user interface, the Agile methodology used, and the feasibility study. The succeeding chapters provide the groundwork for the development and effective use of the chatbot system.

References

[1] G. D. Michael Wallace, "ELIZA: a very basic Rogerian psychotherapist chatbot," [Online]. Available: http://psych.fullerton.edu/mbirnbaum/psych101/eliza.htm.

[2] K. Colby, "Parry," [Online]. Available: https://www.botlibre.com/browse?id=857177.

[3] A. Cheyer, "Siri," [Online]. Available: https://www.apple.com/in/siri/.

[4] S. B. Larry Page, "Hey google," [Online]. Available: https://assistant.google.com/.

[5] A. Jassy, "Amazon," [Online]. Available: https://www.amazon.com/Amazon-Echo-And-Alexa-Devices/b?ie=UTF8&node=9818047011.

[6] S. Altman, "Introducing chatgpt," [Online]. Available: https://chat.openai.com/.

[7] I. Team, "What is User Interface," [Online]. Available: https://www.indeed.com/career-advice/career-development/user-interface.