

AI BASED CHATBOTS TO ANSWER FAQS

A MINI PROJECT REPORT

Submitted by

NEWTTON HABAKUK S [711719104057]

SANJAY K [711719104079]

SUVEKSHA A [711719104097]

THANUSU C [711719104100]

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

KGiSL INSTITUTE OF TECHNOLOGY, SARAVANAMPATTI

ANNA UNIVERSITY :: CHENNAI 600 025

JUNE 2022

ANNA UNIVERSITY : CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report “**AI BASED CHATBOTS TO ANSWER FAQS**” is the bonafide work of “**NEWTTON HABAKUK S, SANJAY K, SUVEKSHA A, THANUSU C**” who carried out the project work under my supervision.

SIGNATURE

Dr. RAMKUMAR R P

Head Of The Department,

Computer Science and Engineering,
KGiSL Institute of Technology,
Saravanampatti,
Coimbatore-35

SIGNATURE

Ms. ARUNA T N

Supervisor,

Assistant Professor
Computer Science and Engineering,
KGiSL Institute of Technology,
Saravanampatti,
Coimbatore-35

Submitted for the Anna University Viva-Voce examination held on _____

Internal Examiner

External Examiner

ACKNOWLEDGEMENT

We wish to express our deep sense of gratitude to our beloved Chairman, **Padma Shri Dr. G. BAKTHAVATHSALAM**, KGiSL Educational Institutions, for having provided the facilities during the course of our study in the college.

We express our heartfelt gratitude to our venerated Managing Director, **Dr. ASHOK BAKTHAVATHSALAM**, KGiSL Educational Institutions, who gave the opportunity to frame the project to the full satisfaction.

We are very grateful to **Mr. R. ARAVIND KUMAR**, CEO, **Dr. M SELVAM, M.E., Ph.D.**, Principal and **Dr. S. SURESH KUMAR, M.E., Ph.D.**, Vice Principal for their valuable guidance and blessings.

We would like to thank **Dr. RAMKUMAR R P, M.E., Ph.D.**, Head of the Department, Department of Computer Science and Engineering for his unwavering support during the entire course of our project work and who modelled us both technically and morally for achieving greater success in this project work.

We express our sincere thanks to our faculty guide **Ms. T. N. ARUNA, M.E.**, Assistant Professor, Department of Computer Science and Engineering.

We also thank all the **faculty members** of our department for their help in making this project a successful one.

Finally, we take this opportunity to extend our deep appreciation to our family and friends, for all they meant to us during the crucial times of the completion of our project.

ABSTRACT

Chatbot is a computer program that simulates and processes human conversation (either written or spoken), allowing humans to interact with digital devices as if they were communicating with a real person. Chatbot's can be as simple as rudimentary programs that answer a simple query with a single-line response, or as sophisticated as digital assistants that learn and evolve to deliver increasing levels of personalization as they gather and process information. Chatbot's boost operational efficiency and bring cost savings to businesses while offering convenience and added services to internal employees and external customers. They allow companies to easily resolve many types of customer queries and issues while reducing the need for human interaction. With chatbot's, a business can scale, personalize, and be proactive all at the same time—which is an important differentiator. The scope of this project is to develop an AI chatbot to answer college related queries like admissions, course details, etc, using Artificial Intelligence and Machine Learning algorithms. Integrate the chatbot into a website with authentication capabilities.

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGENO.
	ABSTRACT	i
	LIST OF TABLES	ii
	LIST OF FIGURES	vi
	LIST OF ABBREVIATIONS	vii
1.	INTRODUCTION	1
	1.1 PROBLEM DEFINITION	1
	1.2 OBJECTIVE OF THE PROJECT	1
	1.3 SIGNIFICANCE OF THE PROJECT	2
	1.4 OUTLINE OF THE PROJECT	2
2.	LITERATURE REVIEW	3
3.	SYSTEM ANALYSIS	5
	3.1 EXISTING SYSTEM	5
	3.2 DRAWBACKS	5
	3.3 PROPOSED SYSTEM	5
	3.4 FEASIBILITY STUDY	6
	3.4.1 Tests of Feasibility	6
	3.4.1.1 Technical Feasibility	7
	3.4.1.2 Operational Feasibility	7
	3.4.1.3 Economical Feasibility	8
4.	SYSTEM SPECIFICATION	9
	4.1 FUNCTIONAL REQUIREMENTS	9
	4.1.1 Chatting	9
	4.1.2 Searching	9

4.1.3 Login	9
4.1.4 Administrative system	9
4.2 NON-FUNCTIONAL REQUIREMENTS	10
4.2.1 User Interface	10
4.2.2 Scalability	10
4.2.3 Security	10
4.2.4 Maintainability	10
4.2.5 Exception handling	10
4.2.6 Ethics	10
4.3 HARDWARE REQUIREMENTS	11
4.4 SOFTWARE REQUIREMENTS	11
5. SOFTWARE DESCRIPTION	12
5.1 FRONT END	12
5.1.1 HTML	12
5.1.1.1 Features	12
5.1.1.2 Advantages	13
5.1.2 CSS	14
5.1.2.1. Features	14
5.1.2.2. Advantages	14
5.1.3 NLP	15
5.1.3.1 Advantages	15
5.2 BACK END	15
5.2.1 Firebase	15
5.2.2 Features of Firebase	15
5.2.3 Advantages of Firebase	16
5.2.4 JSON	17
6. PROJECT DESCRIPTION	18
6.1 OVERVIEW OF THE PROJECT	18
6.2 MODULE DESCRIPTION	18

6.2.1 Login	19
6.2.2 Website	19
6.2.3 Chabot	19
6.2.4 Database	19
6.2.5 Dataset	19
6.3 DATA FLOW DIAGRAM	19
6.3.1 DFD Level 0	20
6.3.2 DFD Level 1	20
6.3.3 DFD Level 2	21
6.4 ER DIAGRAM	22
6.4.1 Chatbot	23
6.5 DATABASE DESIGN	23
6.6 INPUT DESIGN	25
6.7 OUTPUT DESIGN	26
7. SYSTEM TESTING	27
7.1 TESTING METHODS	27
7.2 TYPES OF TESTING	27
7.2.1 Unit Testing	27
7.2.2 Integration Testing	27
7.2.3 Functional Testing	28
7.2.4 Stress Testing	28
7.2.5 Acceptance Testing	29
7.2.6 White Box Testing	29
7.2.7 BlackBox Testing	30
7.2.7.1 Methods of Black Box Testing	30
7.3 TESTING STRATEGY	30
8. SYSTEM IMPLEMENTATION	32
8.1 ACCOUNT CREATION	32
8.2 BOT CREATION	32
8.3 CREATING BOT INTENTS	32

	8.4 BUILDING THE BOT	32
	8.4.1 CREATING TRIGGERS	33
	8.4.2 REQUIREMENTS	33
	8.4.3 ACTIONS	33
	8.5 BOT CONNECTING	33
	8.6 BOT HOSTING	34
	8.7 BOT ANALYTICS	34
9.	CONCLUSION & FUTURE ENHANCEMENTS	35
	9.1 CONCLUSION	35
	9.2 FUTURE ENHANCEMENT	35
10.	APPENDIX	36
	10.1. SOURCE CODE	36
	10.2 SCREENSHOTS	59
11.	REFERENCES	61

LIST OF FIGURES

FIG.NO	FIGURE NAME	PAGE NO
6.3	Data Flow diagram	19
6.1	DFD Level 0	20
6.2	DFD Level 1	20
6.3	DFD Level 2	21

LIST OF ABBREVIATIONS

HTML	Hypertext Markup Language
CSS	Cascading Style Sheet
JS	Java Script
AI	Artificial Intelligence

CHAPTER 1

INTRODUCTION

Chatbot's is one of the blooming technologies to interact with the users more efficiently. It can be used 24*7 with the same effect. Chatbot's reduces human power and answers the queries based on the database collection it has. Conventionally, to enquire about any college people either visit the college physically or ask the college members. So it is very difficult for non-college members to get to know about the college activities.

Chatbot's on the other hand, interacts with the users and answers all the queries with high efficiency. People only have to update the activities into the database so that chatbot gets trained with the dataset. The authentication factor enables the administrator to identify the hacker who tries to inject any malwares into the websites.

1.1 PROBLEM DEFINITION

- College related Chatbot is simply a chatbot that responds to user queries.
- UI plays a major role in this Project. The flexible UI makes the web app easily interact with the Client.
- AI-powered chatbot used for enquiring and helps people to sort out a clear idea about college activities and admission processes.
- Clearly defines the college activities like admissions, transport availability, fees structure, etc.

1.2 OBJECTIVE OF THE PROJECT

The objective of this project is to contribute to the solution of the problem of direct communication between the students and the College management. The main objectives of the project are as follows:

- **Algorithm:** AI-powered chatbot uses the reinforcement Algorithm. To develop a keyword matching algorithm and a string distance comparison algorithm and combine them in order to retrieve the best possible answer.

- **Interface:** To develop a web interface that aims to give the ability to potential clients and naive users to submit questions in a chatbot and get convincing replies.

In this chatbot, our main objectives are as follows.

- To minimize the time consumption.
- To has 24*7 access to the College managment.
- Has eliminated most of the flaws in the existing chatbot applications.

1.3 SIGNIFICANCE OF THE PROJECT

- It is important to provide an accurate reply to queries.
- Helps clients with enriching customer experience.
- Easy and Interactive objectives.

1.4 OUTLINE OF THE PROJECT

College chatbot provides a complete set of methods to support Admissions.

All users of this system need the same functionality to complete their varied tasks as follows:

- Different UI from other College website
- Chatbot to answer all queries.
- Provide details related to admissions, transport facility, courses,etc.

CHAPTER 2

LITERATURE REVIEW

The purpose of a chatbot system is to simulate a human conversation. The chatbot architecture integrates a language model and computational algorithm to emulate information chat communication between a human user and a computer using natural language. c

In E-commerce, the chatbot helps in information retrieval tasks, such as searching and browsing, as menu-based navigation poses difficulties in locating the appropriate information. The dialogue system provides additional information on products and simplifies the decision-making process to find a product that satisfies the customer's requirements. A chatbot is Entertainment Markets, in which, we can imagine that chatbots can act as a talking book for children and provide foreign language instruction or can be a tutor in the Intelligent Tutoring System.

One such study used an ALICE system to help Chinese university students practice their conversational English skills. The study was qualitative in nature and used pre-existing conversational English skills. The study focused more on user attitudes rather than on chatterbot efficiency. It was discovered that 62% of users chatted for 10 lines or less and that 8.5% of the time ALICE bot has no specific pattern to match the given input and had to rely on root-level generic responses. In all of these conversational entities, one thing is common; and that is, they are having the difficulty of maintaining dialogue for a sustainable period of time.

Another tutoring study focused on using ALICE as a course enhancement tools with Social and Political Theory knowledge. Chatterbot development is reasonably well studied ever since the Turing Imitation Game (TIG) was first proposed. Eliza was the first famous chatbot and ALICE was another milestone. The Loebner Prize and The Chatter Challenge are both annual competitions that have their roots in TIG. This focus is on; however, whether with the text exchange alone, we can replicate human "behavior".This The study found that most subjects used the system as a search engine rather than as a conversation partner.

Dialog systems can adequately carry out the conversations with the user and can log the conversations which can be a good source for knowledge acquisition for the domain-specific topics. Therefore, techniques of knowledge acquisition were rightly used in their system AZ-ALICE chatbot that is an extension in ALICE chatterbot and tested their system. From this study, we have learned about the functionalities of a chatbot.

CHAPTER 3

SYSTEM ANALYSIS

System analysis is a problem solving technique that decay a system into component pieces of purpose of studying how well those component parts work and interact to accomplish their purpose. The following chapter provides a detailed description of the existing system. It also provides an overview of the proposed system and feasibility of college bot.

3.1 EXISTING SYSTEM

In the earlier days students had to visit the college to enquire about details like courses ,fee structure ,admission process and other information's about the college ,which is a tiresome process as well as long process for both parents as well as students. Nowadays there are many changes occurred in the Education system with help of advanced technology. Everything is happening over the internet without any trouble. In those days for enquiring about courses we have to visit the college, but as the days are passing away it's completing changing. Collecting the course details, fee structure manually will be hectic procedure and it also needs manpower. For reducing that manpower and avoid such difficulties and time consuming many devices or systems were emerged day by day.

3.2 DRAWBACKS

- Deficit in informing non-college members about the college and its daily activities.
- Separate group of people as centre point of communication for clearing doubts related to college.

3.3 PROPOSED SYSTEM

Chatbot is a computer program that simulates and processes human conversation

(either written or spoken), allowing humans to interact with digital devices as if they were communicating with a real person. Chatbot's can be as simple as rudimentary programs that answer a simple query with a single-line response, or as sophisticated as digital assistants that learn and evolve to deliver increasing levels of personalization as they gather and process information.

Chatbot's boost operational efficiency and bring cost savings to businesses while offering convenience and added services to internal employees and external customers. They allow companies to easily resolve many types of customer queries and issues while reducing the need for human interaction. With chatbot's, a business can scale, personalize, and be proactive all at the same time—which is an important differentiator. The scope of this project is to develop an AI chatbot to answer college related queries like admissions, course details, etc, using Artificial Intelligence and Machine Learning algorithms. Integrate the chatbot into a website with authentication capabilities.

3.4 FEASIBILITY STUDY

An analysis and evaluation of a proposed project to determine if it is technically feasible, is feasible within the estimated cost, and will be profitable. Feasibility studies are almost always conducted where large sums are at stake. A feasibility study aims to objectively and rationally uncover the strengths and weaknesses of an existing insurance agency applications and threats present in the environment, the resources required to carry through, and ultimately the prospects for success in the college related Chatbot.

3.4.1 Tests of Feasibility

Feasibility study is conducted once the problem is clearly understood. Feasibility study is necessary to determine that the proposed system in insurance bot is feasible by considering the technical, operational, and economical factors. By having a detailed feasibility study the management in the will have a clear-cut view of the proposed system of the insurance bot. Feasibility study encompasses the following things:

- Technical Feasibility

- Economical Feasibility
- Operational feasibility

3.4.1.1 Technical Feasibility

A large part of determining resources has to do with assessing technical feasibility. It considers the technical requirements of the proposed project of banking bot. The technical requirements are then compared to the technical capability of the banking system. The systems project is considered technically feasible if the internal technical capability is sufficient to support the banking system's requirements. The analyst must find out whether current technical resources can be upgraded or added to in a manner that fulfils the request under consideration.

The essential questions that help in testing the operational feasibility of a system include the following:

- Is the project feasible within the limits of current technology?
- Does technology exist at all?
- Is it available within given resource constraints?
- Is it a practical proposition?
- Manpower- programmers, testers & debuggers
- Software and hardware
- Are the current technical resources sufficient for the new system?
- Can they be upgraded to provide the level of technology necessary for the new system?

3.4.1.2 Operational Feasibility

Operational feasibility is dependent on human resources available for the project and involves projecting whether the system will be used if it is developed and implemented. Operational feasibility is a measure of how well a proposed system in insurance bot solves the problems, and takes advantage of the opportunities identified

during scope definition and how it satisfies the requirements identified in the requirements analysis phase of insurance bot development.

The essential questions that help in testing the operational feasibility of a system include the following:

- Does current mode of operation provide adequate throughput and response time?
- Does current mode provide end users and managers with timely, pertinent, accurate and useful formatted information?
- Does current mode of operation provide cost-effective information services to the business?
- Could there be a reduction in cost and or an increase in benefits?

3.4.1.3 Economical Feasibility

Economic analysis could also be referred to as cost/benefit analysis. It is the most frequently used method for evaluating the effectiveness of a new system of the banking bot. In economic analysis the procedure is to determine the benefits and savings that are expected from a candidate system and compare them with costs.

If benefits outweigh costs, then the decision is made to design and implement the banking bot. An entrepreneur must accurately weigh the cost versus benefits before taking an action.

Possible questions raised in economic analysis are:

- Is the system cost effective?
- Do benefits outweigh costs and system study?

CHAPTER 4

SYSTEM SPECIFICATION

4.1 FUNCTIONAL REQUIREMENTS

4.1.1 Chatting

- The system should allow users to chat.
- The system shall inform the user if an answer is not available.
- The system shall inform the user about spelling mistakes.
- The system shall inform the user about the validity of the sentence.

4.1.2 Searching

- The system should allow users to search for information about their login details.
- The system should allow users to search for information about courses, fees structure, admissions, etc.
- The system should allow users to search for information about the login.

4.1.3 Login

The system should maintain a database to store all the login credentials.

4.1.4 Administrative system

- Information management: The administrator should be able to add, update and delete college related activities.
- Login management: The administrator should be able to view and maintain login credentials.

4.2 NON-FUNCTIONAL REQUIREMENTS

4.2.1 User Interface

- The system shall maintain an easy to use interface across all functionality and for all users
- The client's user interface should be compatible with all commonly used browsers, such as Internet explorer, Firefox, Google chrome and Safari.

4.2.2 Scalability

The system shall be able to scale based on the number of users accounts using the system.

4.2.3 Security

- The administrative system should be protected from unauthorized access.
- The database should be protected from attacks and unauthorized access.
- The interface should be protected from attacks.
- All passwords should be stored as a secure hash of the administrator password.

4.2.4 Maintainability

- The system should be easy to maintain.
- There should be a clear separation between the interface and the business logic code.
- There should be a clear separation between the data access objects that map the database and the business logic code.

4.2.5 Exception handling

Exceptions should be reported effectively to the user if they occur.

4.2.6 Ethics

The system shall not store or process any information about its users.

4.3 HARDWARE REQUIREMENTS

Processor	:	Dual core processor
RAM	:	2 GB
Hard Disk	:	250 GB
Monitor	:	16’’ Color Monitor
Keyboard	:	Standard 110 keys
Pointing Device	:	Mouse
Smart Phone	:	Any type

4.4 SOFTWARE REQUIREMENTS

Programming Language	:	Python
Operating System	:	Windows/Ubuntu/Linux/Mac
Front End	:	Html, CSS, JavaScript
Back End	:	Firebase
Web Browser	:	Mozilla Firefox, Google Chrome

CHAPTER 5

SOFTWARE DESCRIPTION

A software requirements specification (SRS) is a description of a software system to be developed. It lays out functional and non-functional requirements, and may include a set of use cases that describe user interactions that the software must provide. Software requirements specification establishes the basis for an agreement between users and chat bot on what the software product is to do as well as what it is not expected to do. Software requirements specification permits a rigorous assessment of requirements before design can begin and reduces later redesign. It should also provide a realistic basis for estimating product costs, risks, and schedules.

5.1 FRONT END

The front end is designed using basic web development technologies like HTML, CSS, and JavaScript. It is the collaborative end to end bot platform made by developers for the developers. Here all the bot tools are integrated and it allows automatic detection of entities. It uses Machine Learning. API Connectivity is been done using PHP.

5.1.1 HTML

HTML stands for HyperText Markup Language. It is used to design web pages using a markup language. HTML is the combination of Hypertext and Markup language. Hypertext defines the link between the web pages. A markup language is used to define the text document within tag which defines the structure of web pages. This language is used to annotate (make notes for the computer) text so that a machine can understand it and manipulate text accordingly. Most markup languages (e.g.HTML) are human-readable. The language uses tags to define what manipulation has to be done on the text.

5.1.1.1 Features

- Simple: It is very simple and easy to use, compare to other markup language it is very simple and easy, this is widely used all over the world.

- **Interpreted:** It is an interpreted language, i.e. there is no need for compilation.
- **Open Source:** Open source means you no need to pay for use html, you can free download and use.
- **Platform Independent:** HTML code will be run on every platform, Linux, Unix, Mac OS X, Windows.

5.1.1.2 Advantages

- **HTML is Browser friendly:** HTML upholds a heap of programs like Google Chrome, Mozilla Firefox, Opera, and so on This makes HTML very program amicable because of its quality among the top current programs.
- **Allowed to utilize:** Since HTML is open-source, it's allowed to utilize. It's an incredible benefit for organizations and people the same as there is no compelling reason to buy the additional products for composing HTML code as it were.
- **Simple to learn:** HTML is the fundamental reinforcement of web advancement. It is so natural to discover that school understudies can likewise utilize HTML to make their own fundamental site with pictures and tones.
- **Basic design:** HTML is a basic language and has a proper construction with predefined labels and traits. Along these lines, in the event that you practice routinely, you'll become accustomed to the straightforward HTML structure and will actually want to compose HTML code without any problem.
- **Lightweight and quick:** HTML is a lightweight markup language. It saves time for the clients by decreasing stacking time.
- **Utilization of formats:** HTML permits the utilization of formats for developers. In this way, it makes the errand of the web engineers significantly simpler by diminishing the time taken to compose the plan codes.
- **Information stockpiling:** HTML upholds Data Storage. HTML web stockpiling gives two items to put away information. They are as per the following:
`window.sessionStorage` – This stores the information for one meeting.
`window.localStorage` – This stores the information with no termination date

5.1.2 CSS

CSS handles the look and feel part of a web page. Using CSS, you can control the color of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, etc.

5.1.2.1 Features

- **Opportunity in Web designing:** If anyone wants to begin a career in web designing professionally, it is essential to have knowledge of CSS and HTML.
- **Website Design:** With the use of CSS, we can control various styles, such as the text color, the font style, the spacing among paragraphs, column size and layout, background color and images, design of the layout, display variations for distinct screens and device sizes, and many other effects as well.
- **Web Control:** CSS has controlling power on the documents of HTML, so it is easy to learn. It is integrated with the HTML and the XHTML markup languages.
- **Other Languages:** Once we have knowledge of some basics of CSS and HTML, other associated technologies like Angular, PHP, and JavaScript are become clearer to understand.

5.1.2.2 Advantages

- CSS plays an important role, by using CSS you simply got to specify a repeated style for element once & use it multiple times as because CSS will automatically apply the required styles.
- The main advantage of CSS is that style is applied consistently across variety of sites. One instruction can control several areas which is advantageous.
- Web designers needs to use few lines of programming for every page improving site speed.
- Cascading sheet not only simplifies website development, but also simplifies the maintenance as a change of one line of code affects the whole web site and maintenance time.
- It is less complex therefore the efforts are significantly reduced.

- It helps to form spontaneous and consistent changes.
- CSS changes are device friendly. With people employing a batch of various ranges of smart devices to access websites over the web, there's a requirement for responsive web design.
- It has the power for re-positioning. It helps us to determine the changes within the position of web elements who are there on the page.
- These bandwidth savings are substantial figures of insignificant tags that are indistinct from a mess of pages.
- Easy for the user to customize the online page
- It reduces the file transfer size.

5.1.3 NLP

Natural-language processing (NLP) is an area of computer science and artificial intelligence concerned with the interactions between computers and human (natural) languages, in particular how to program computers to fruitfully process large amounts of natural language data.

5.1.3.1 Advantages

- No training.
- Relives burden of learning syntax.

5.2 BACK END

The back end is designed using Firebase, whose primary function is to store data securely and retrieve it later, as requested by other software applications.

5.2.1 Firebase

Google Firebase is Google-backed application development software which allows developers to develop Android, IOS, and Web apps. For reporting and fixing app crashes, tracking analytics, creating marketing and product experiments, firebase provides several tools.

Firebase manages real-time data in the database. So, it easily and quickly exchanges the data to and from the database. Hence, for developing mobile apps such as live

streaming, chat messaging, etc., we can use Firebase. Firebase allows syncing real-time data across all devices - iOS, Android, and Web - without refreshing the screen. Firebase provides integration to Google Advertising, AdMob, Data Studio, BigQuery, DoubleClick, Play Store, and Slack to develop our apps with efficient and accurate management and maintenance. Everything from databases, analytics to crash reports are included in Firebase. So, the app development team can stay focused on improving the user experience. Firebase applications can be deployed over a secured connection to the firebase server. Firebase offers a simple control dashboard.

5.2.2 Features of Firebase

- **Cloud Messaging** : Firebase allows us to deliver and receive messages in a more reliable way across platforms.
- **Authentication** : Firebase has little friction with acclaimed authentication.
- **Test Lab** : Test in the lab instead on your users.
- **Hosting** : Firebase delivers web content faster.
- **Remote Configuration** : It allows us to customize our app on the go.
- **Dynamic Links** : Dynamic Links are smart URLs which dynamically change behavior for providing the best experience across different platforms.
- **Crash Reporting** : It keeps our app stable.
- **Real-time Database** : It can store and sync app data in real-time.
- **Storage** : We can easily store the file in the database.

5.2.3 Advantages of Firebase

- Fast and Safe hosting
- Reliable and Extensive databases
- Google Analytics
- Free Multi-Platform firebase authentication
- Firebase Testing Service to Improve App Quality
- Free Use of Firebase Dynamic Links

5.2.4 JSON

JSON, or JavaScript Object Notation, is a minimal, readable format for structuring data. It is used primarily to transmit data between a server and web application, as an alternative to XML.

The two primary parts that make up JSON are keys and values. Together they make a key/value pair.

- **Key:** A key is always a string enclosed in quotation marks.
- **Value:** A value can be a string, number, boolean expression, array, or object.
- **Key/Value Pair:** A key value pair follows a specific syntax, with the key followed by a colon followed by the value. Key/value pairs are comma separated.

CHAPTER 6

PROJECT DESCRIPTION

Chat bot are used in most of the MNC. Few of the operations such as suggesting courses, fees details, admission process and other process can be performed using this bot. This bot is been deployed using user needs and it is used to integrate few of the details.

6.1 OVERVIEW OF THE PROJECT

The college admission chat bot provides a complete set of methods in order to perform the conversation. All users of the chat bot need the same functionality to complete their varied tasks.

The college admission chat bot allows you to:

- Ask query
- Search courses available
- Check the availability of transport
- All the conversations are secured with help of database.
- All the new query can be monitored by bot admin
- Bot analytics can be performed in order to improve efficiency.

6.2 MODULE DESCRIPTION

6.2.1 Login

If the user is new to the application then he/she want to register it with their email-id and password. If they are a registered user then they can login to the page using their credentials

6.2.2 Website

Website consists of the details related to college, admission links, contact details of the college, etc. Along with this a chatbot would present in the bottom right corner.

6.2.3 Chatbot

Chatbot in the website will interact with the user to clarify their queries related to the college. It receives the query and search for the response in its dataset and reply the user accordingly.

6.2.4 Database

Login credentials of the users are stored in the database for future verifications if any intruder tries to send any malicious messages. It is also used to verify the user during the login process.

6.2.5 Dataset

Dataset of the chatbot consists of the FAQs in terms of keywords and responses. Administrator will update the dataset as and when any new updates are being done in the website.

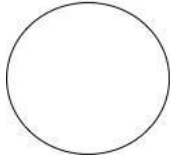
6.3 DATA FLOW DIAGRAM

Data flow diagram is used to describe how the information is processed and stored and identifies how the information flows through the processes. Data flow diagram illustrates how the data is processed by a system in terms of inputs and outputs. The data flow diagram also depicts the flow of the process and it has various levels. The initial level is context level which describes the entire system functionality and the next level describes each and every sub module in the main system as a separate process or describes all the process involved in the system separately.

Data flow diagram are made up of number of symbols,



Square representing external entities, which are sources or destinations of data.



Circle representing processes, which take data as input, do something to it and output it.



Arrows representing the data flows, which can either, be electronic data or physical items.



Parallel lines representing data stores, including electronic stores such as databases or XML files and physical stores

6.3.1 DFD Level 0

The users of the chat bot are user and bot. The output of the system is reports. The users store and retrieve from the database.

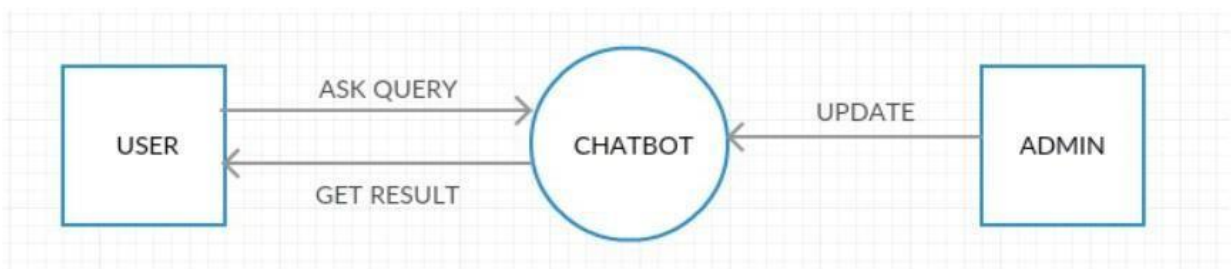


Fig.6.1 DFD Level 0

6.3.1 DFD Level 1

In the DFD level 1, all the user login are done by using the user login db, the user give the user name and password as input ,the input will check with the user db and the it

record the details and allow the user to login into the webpage

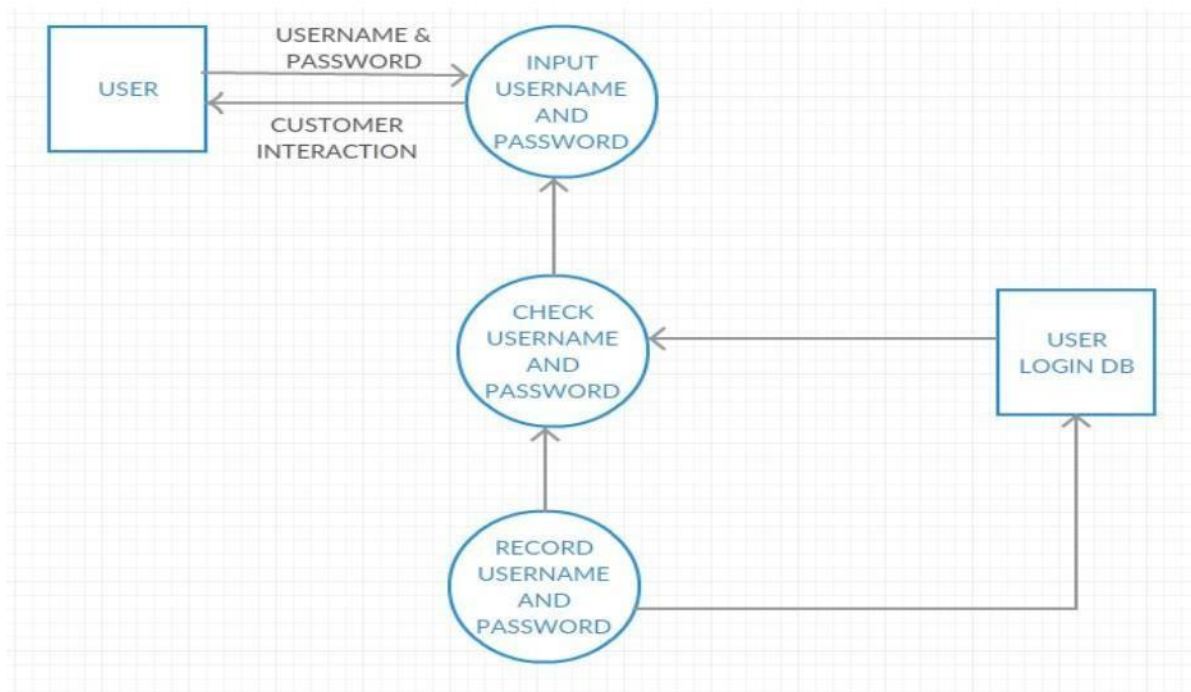


Fig.6.2 DFD Level 1

6.3.3 DFD level 2

The user type the query in the chatbot and then the chatbot will fetch the keyword From the entered query and the chatbot will search the result for the query in the query db and show the result to the user

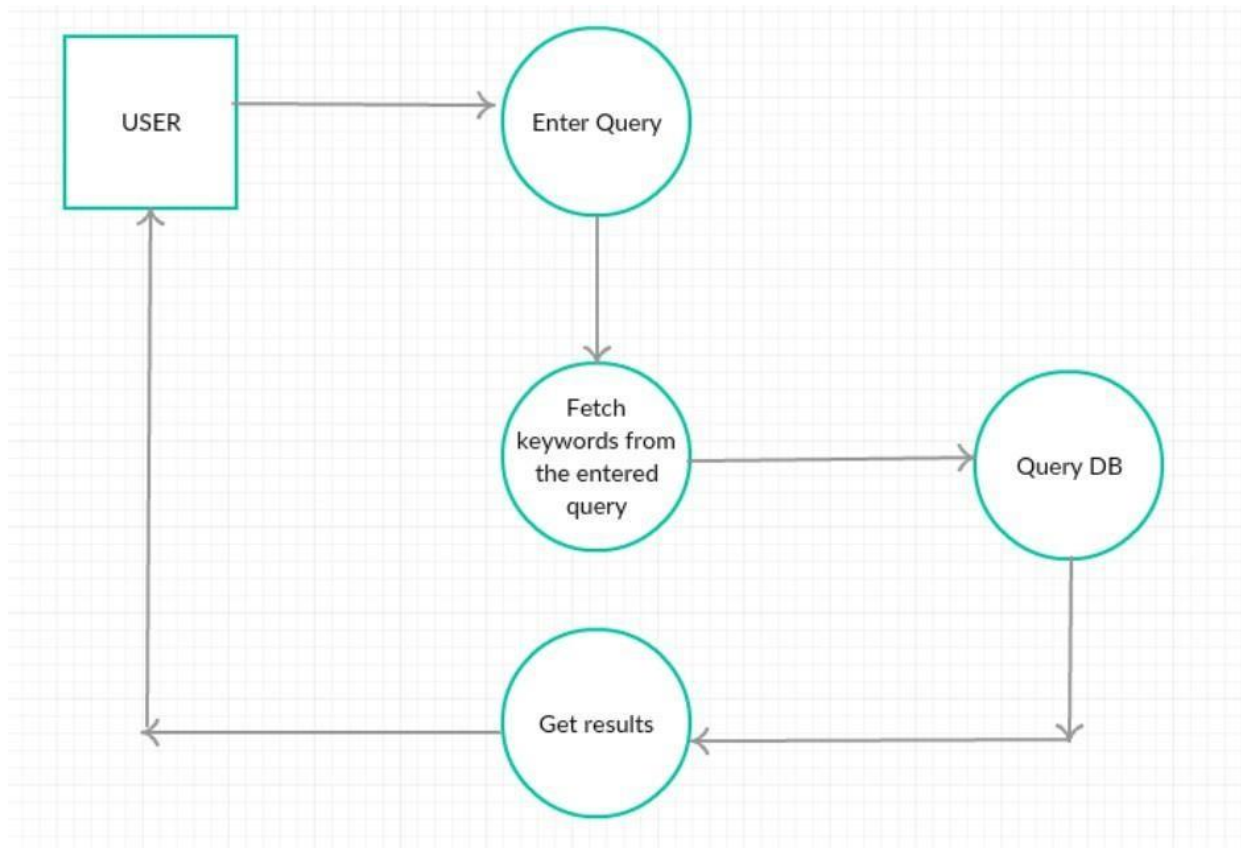


Fig.6.3 DFD Level 2

6.4 ER DIAGRAM

An entity–relationship model (ER model) describes inter-related things of interest in a specific domain of knowledge. An ER model is composed of entity types (which classify the things of interest) and specifies relationships that can exist between instances of those entity types. In software engineering an ER model is commonly formed to represent things that a business needs to remember in order to perform business processes. Consequently, the ER model becomes an abstract data model that defines a data or information structure that can be implemented in a database, typically a relational database.

Entity–relationship modeling was developed for database design by Peter Chen and published in a 1976 paper. Some ER modelers show super and subtype entities connected by generalization-specialization relationships, and an ER model can be used also in the specification of domain-specific ontology.

6.4.1 Chatbot

In our project we used the chatbot to answer the query which is asked by the user related to travel insurance. we use the separate login page for user and admin

All the information's are stored in the database. the chatbot will answer the user query and store the unanswered query in the db

In login page we collect the basic details such as mobile number,username,password in admin page we collect the username and password

The ER Diagram for all the process flow is been given as follows.

The Diagram is been given for all the processes such as the process flow is been specified.

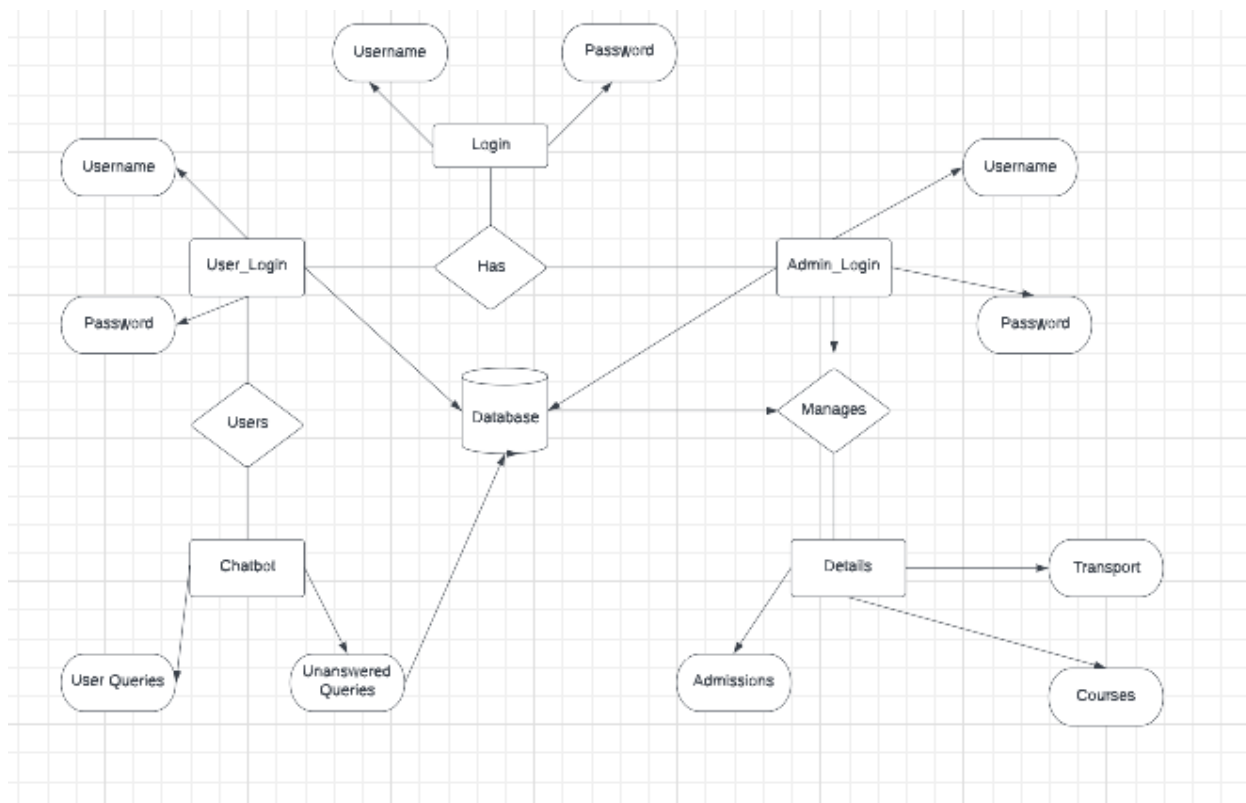


Fig.6.4 ER DIAGRAM

6.5 DATABASE DESIGN

Database design is the process of producing a detailed data model of database. This data model contains all the needed logical and physical design choices and physical storage parameters needed to generate a design in a data definition language, which can then be used to create a database. A fully attributed data model contains detailed attributes for each entity.

The term database design can be used to describe many different parts of the design of an overall database system. Principally, and most correctly, it can be thought of as the logical design of the base data structures used to store the data. In the relational model these are the tables and views. In an object database the entities and relationships map directly to object classes and named relationships. However, the term database design could also be used to apply to the overall process of designing, not just the base data structures, but also the forms and queries used as part of the overall database application within the database management system (DBMS).

TABLE 6.6.1 User login detail

S.NO	FIELD NAME	DATATYPE
1	user name	Varchar
2	passsword	Varchar

TABLE 6.6.2 Admin login details

S.NO	FIELD NAME	DATATYPE
1	Admin name	varchar
2	password	Varchar

TABLE 6.6.3 User's Details

S.NO	FIELD NAME	FIELD TYPE
1	User's name	Varchar
3	password	Varchar

6.6 INPUT DESIGN

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

- What data should be given as input?
- How the data should be arranged or coded?
- The dialog to guide the operating personnel in providing input.
- Methods for preparing input validations and steps to follow when error occur.

Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.

It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.

When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow.

6.7 OUTPUT DESIGN

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

CHAPTER 7

SYSTEM TESTING

System Testing is a level of the software testing where complete and integrated software is tested. The purpose of this test is to evaluate the system's compliance with the specified requirements. By definition of ISTQB system testing is the process of testing an integrated system to verify that it meets specified.

7.1 TESTING METHODS

Software Testing Type is a classification of different testing activities into categories, each having, a defined test objective, test strategy, and test deliverables. The goal of having a testing type is to validate the Application under Test for the defined Test Objective.

For instance, the goal of Accessibility testing is to validate the AUT to be accessible by disabled people. So, if your Software solution must be disabled friendly, you check it against Accessibility Test Cases.

7.2 TYPES OF TESTING

7.2.1 Unit Testing

In computer programming, unit testing is a software testing method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures, are tested to determine whether they are fit for use.

In this chatbot, every units of code is been tested and the correctness of every module is been ensured.

7.2.2 Integration Testing

Integration testing (sometimes called integration and testing, abbreviated I&T) is the phase in software testing in which individual software modules are combined and tested as a group. It occurs after unit testing and before validation testing. Integration testing takes as its input modules that have been unit tested, groups them in larger aggregates,

applies tests defined in an integration test plan to those aggregates, and delivers as its output the integrated system ready for system testing.

In this bot, the units are been tested as a whole and the testing was successful.

7.2.3 Functional Testing

Functional testing is a quality assurance (QA) process and a type of black-box testing that bases its test cases on the specifications of the software component under test. Functions are tested by feeding them input and examining the output, and internal program structure is rarely considered (unlike white-box testing). Functional testing usually describes what the system does. Functional testing does not imply that you are testing a function (method) of your module or class. Functional testing tests a slice of functionality of the whole system.

Functional testing has many types:

- Smoke testing
- Sanity testing
- Regression testing
- Usability testing

7.2.4 Stress Testing

Stress testing a Non-Functional testing technique that is performed as part of performance testing. During stress testing, the system is monitored after subjecting the system to overload to ensure that the system can sustain the stress.

Reasons can include:

- to determine breaking points or safe usage limits
- to confirm mathematical model is accurate enough in predicting breaking points or safe usage limits
- to confirm intended specifications are being met
- to determine modes of failure (how exactly a system fails)
- to test stable operation of a part or system outside standard usage

The recovery of the system from such phase (after stress) is very critical as it is highly likely to happen in production environment.

In this chatbot, whole of the modules are been tested and it has the safe usage measures.

7.2.5 Acceptance Testing

Acceptance Testing is a level of the software testing where a system is tested for acceptability. The purpose of this test is to evaluate the system's compliance with the business requirements and assess whether it is acceptable for delivery.

Formal testing with respect to user needs, requirements, and business processes conducted to determine whether or not a system satisfies the acceptance criteria and to enable the user, customers or other authorized entity to determine whether or not to accept the system.

In this bot, the customer's acceptance is been monitored and it is been put into usage.

7.2.6 White Box Testing

White Box Testing is the testing of a software solution's internal coding and infrastructure. It focuses primarily on strengthening security, the flow of inputs and outputs through the application, and improving design and usability. White box testing is also known as Clear Box testing, Open Box testing, Structural testing, Transparent Box testing, Code-Based testing, and Glass Box are testing. It is one of two parts of the "box testing" approach of software testing. Its counter-part, black box testing, involves testing from an external or end-user type perspective. On the other hand, White box testing is based on the inner workings of an application and revolves around internal testing.

The term "white box" was used because of the see-through box concept. The clear box or white box name symbolizes the ability to see through the software's outer shell (or "box") into its inner workings. Likewise, the "black box" in "black box testing" symbolizes not being able to see the inner workings of the software so that only the end-user experience can be tested.

In this chatbot, all the inner functionality is been tested and it is been correctly implemented.

7.2.7 BlackBox Testing

Black box testing is a software testing techniques in which functionality of the software under test (SUT) is tested without looking at the internal code structure, implementation details and knowledge of internal paths of the software. This type of testing is based entirely on the software requirements and specifications.

In this chatbot, the implementation part is been checked for its correctness.

7.2.7.1 Methods of Black Box Testing

There are many types of Black Box Testing but following are the prominent ones -

- Functional testing - This black box testing type is related to functional requirements of a system; it is done by software testers.
- Non-functional testing - This type of black box testing is not related to testing of a specific functionality, but non-functional requirements such as performance, scalability, usability.
- Regression testing - Regression testing is done after code fixes, upgrades or any other system maintenance to check the new code has not affected the existing code.

7.3 TESTING STRATEGY

Test Strategy is also known as test approach defines how testing would be carried out. Test approach has two techniques:

- Proactive - An approach in which the test design process is initiated as early as possible in order to find and fix the defects before the build is created.
- Reactive - An approach in which the testing is not started until after design and coding are completed.

Test strategy calls for implementing two entirely different methodologies for testing this project. The chatbot includes a fair amount of manual UI-based testing.

In this chatbot, proactive approach is been used for testing. Since proactive approach is efficient it is been used in this bot.

CHAPTER 8

SYSTEM IMPLEMENTATION

The system is been implemented as follows:

8.1 ACCOUNT CREATION

It is handy to create an account in the recast framework in which it prompts for a simple login with the mail id/MobileNumber and the password credentials, on verification of the mail id, it allows the user to create bots on the framework.

8.2 BOT CREATION

To create a bot, click on the plus symbol which is available on the framework screen. Click on the new bot text that navigates to the form that prompts for the bot name, description of the bot, nature of visibility whether it can be public or private, default bot conversations such as small talk, temperature etc. The things that are needed can be selected by the user and finally bot is created.

8.3 CREATING BOT INTENTS

The first step is to create the bot intents. Generally intent is that our main intention to chat with the bot. There are 32 default intents in this framework and the intents can be created if needed and the keywords must be made to map in the same so that if the intent is made to be matched in the user's query, then the corresponding action can be made to perform.

8.4 BUILDING THE BOT

The bot is then made to build either by creating new skills or by selecting any of the existing skill sets available. Bot building forms the main task in creating the bot which allows creating triggers, setting the corresponding requirements and performs necessary actions. This is main motive in performing the building task.

8.4.1 CREATING TRIGGERS

Trigger creation forms the main part in which the intents that are created or forked can be invoked in this tab. The triggers can be made to perform the actions such as creating the corresponding action when the intent is present or not present or missing. The default will be present and this can be altered based on the condition required. The actions will be triggered only based on this condition.

8.4.2 REQUIREMENTS

The requirements tab allows the user to perform the relevant tasks that are been pertained based on the condition given on the trigger tab. Generally, the requirements are been specified in special cases if not required then it can be omitted.

8.4.3 ACTIONS

The actions are been generally performed with the creation of entities. The entities are the data bucket that is used to store the values. The entities are been used to map the keywords with the intent and it is been made to give the response for the user's queries. In this actions tab, the way by which the bot responds to the query will be given. It can be added with text, buttons, images, animated GIFs, quick replies, cards, links and fallbacks. All these contribute to the bot training.

8.5 BOT CONNECTING

Bot connecting allows the developer to connect the bot with any of the social channels such as messenger, kik, web chat etc. We have implemented this bot with the messenger. The steps followed are as below:

- Go to the bot connector tab
- Select the ways of deploying the bot
- In which select the web chat
- Create my channel tab and create the web channel
- Open the ngrok tool and copy url of ngrok in the bot connecting tab.

8.6 BOT HOSTING

The bot is been hosted with ngrok which is the public URL in order to connect the bot with API that results in displaying the request and response slug. This is done using request token. The bot is been connected with the slug and with the API for the accurate response.

8.7 BOT ANALYTICS

Bot analytics is made just to analyze the performance of the bot and it will have all the logs of user's queries and it also count the number of occurrence of the intent and whether the transaction is successful or not. If the user's intent is not found, then it displays the default value as the no reply.

CHAPTER 9

CONCLUSION & FUTURE ENHANCEMENTS

9.1 CONCLUSION

A lack of trust isn't the only reason why insurance companies have a hard time selling their products. A lack of knowledge about insurance and its benefits is also at play. Millennials in particular have been found to buy insurance less often than their predecessors, and that's mostly because they don't know enough about it. Online chatbots have the potential to change this scenario. Thus this project travel insurance chat bot will be more efficient while it is been put into practice and it helps the customers to easily perform the user's action of performing various tasks. The user can chat with out chat bot regarding travel insurance .we use the php framwork for our project and mysql database for our travel insurance chatbot. the chat So user will have no issues in using this bot.

9.2 FUTURE ENHANCEMENT

This project is focused on bot that easily interact with the user. The future enhancements can be done by using the voice recognition mechanism and the bot can be deployed in various social channels and it can be made to implement in different languages.

CHAPTER 10

APPENDIX

10.1. SOURCE CODE

App.js

```
class Chatbox {
  constructor() {
    this.args = {
      openButton: document.querySelector('.chatbox_button'),
      chatBox: document.querySelector('.chatbox support'),
      sendButton: document.querySelector('.send-button')
    }
    this.state = false;
    this.messages = [];
  }

  display() {
    const { openButton, chatBox, sendButton } = this.args;
    openButton.addEventListener('click', () => this.toggleState(chatBox))
    sendButton.addEventListener('click', () => this.onSendButton(chatBox))
    const node = chatBox.querySelector('input');
    node.addEventListener("keyup", ({key}) => {
      if (key === "Enter") {
        this.onSendButton(chatBox)
      }
    })
  }

  toggleState(chatbox) {
    this.state = !this.state;
    // show or hides the box
    if(this.state) {
      chatbox.classList.add('chatbox--active')
    } else {
      chatbox.classList.remove('chatbox--active')
    }
  }

  onSendButton(chatbox) {
    var textField = chatbox.querySelector('input');
```

```

let text1 = textField.value
if (text1 === "") {
    return;
}
let msg1 = { name: "User", message: text1 }
this.messages.push(msg1);
fetch('http://127.0.0.1:5000/predict', {
    method: 'POST',
    body: JSON.stringify({ message: text1 }),
    mode: 'cors',
    headers: {
        'Content-Type': 'application/json'
    },
})
.then(r => r.json())
.then(r => {
    let msg2 = { name: "Sam", message: r.answer };
    this.messages.push(msg2);
    this.updateChatText(chatbox)
    textField.value = ""

}).catch((error) => {
    console.error('Error:', error);
    this.updateChatText(chatbox)
    textField.value = ""
});
}

updateChatText(chatbox) {
    var html = "";
    this.messages.slice().reverse().forEach(function(item, index) {
        if (item.name === "Sam")
        {
            html += '<div class="messages__item messages__item--visitor">' +
            item.message + '</div>'
        }
        else
        {
            html += '<div class="messages__item messages__item--operator">' +
            item.message + '</div>'
        }
    });
}

```

```

    }
  });
  const chatmessage = chatbox.querySelector('.chatbox_messages');
  chatmessage.innerHTML = html;
}
}
document.querySelector('.main');
const chatbox = new Chatbox();
chatbox.display();

```

style.css

```

* {
  box-sizing: border-box;
  margin: 0;
  padding: 0;
}

*, html {
  --primaryGradient: linear-gradient(93.12deg, #581B98 0.52%, #9C1DE7 100%);
  --secondaryGradient: linear-gradient(268.91deg, #581B98 -2.14%, #9C1DE7 99.69%);
  --primaryBoxShadow: 0px 10px 15px rgba(0, 0, 0, 0.1);
  --secondaryBoxShadow: 0px -10px 15px rgba(0, 0, 0, 0.1);
  --primary: #581B98;
}

/* CHATBOX
===== */
.chatbox {
  position: fixed;
  bottom: 30px;
  right: 30px;
}

/* CONTENT IS CLOSE */
.chatbox_support {
  display: flex;
  flex-direction: column;
  background: #eee;

```



```

width: 300px;
height: 350px;

z-index: -123456;

opacity: 0;

transition: all .5s ease-in-out;
}

/* CONTENT ISOPEN */

.chatbox--active {

    transform: translateY(-40px);
    z-index: 123456;
    opacity: 1;
}

/* BUTTON */

.chatbox_button {
    text-align: right;
}

.send_button {
    padding: 6px;
    background: transparent;
    border: none;
    outline: none;
    cursor: pointer;
}

/* HEADER */

.chatbox_header {
    position: sticky fixed;
    top: 0;
    background: orange;
}

/* MESSAGES */

.chatbox_messages {
    margin-top: auto;
    display: flex;
    overflow-y: scroll;
    flex-direction: column-reverse;

```

```

}

.messages_item {
    background: orange;
    max-width: 60.6%;
    width: fit-content;
}

.messages__item--operator {
    margin-left: auto;
}

.messages__item--visitor {
    margin-right: auto;
}

/* FOOTER */

.chatbox_footer

```

App.py

```

from xmlrpc.client import ResponseError

from flask import Flask,render_template,request,jsonify
from chat import get_response
app=Flask(__name__)

@app.route("/",methods=["GET"])
def index_get():
    return render_template("base.html")

@app.post("/predict")
def predict():
    text = request.get_json().get("message")
    response=get_response(text)
    message = {"answer":response}
    return jsonify(message)

if __name__==" __main__ ":
    app.run(debug=True)

```

chat.py

```

import random
import json
import torch
from model import NeuralNet

```

```

from nltk_utils import bag_of_words, tokenize
device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')
with open('intents.json', 'r') as json_data:
    intents = json.load(json_data)

FILE = "data.pth"

data = torch.load(FILE)

input_size = data["input_size"]
hidden_size = data["hidden_size"]
output_size = data["output_size"]
all_words = data['all_words']
tags = data['tags']
model_state = data["model_state"]

model = NeuralNet(input_size, hidden_size, output_size).to(device)
model.load_state_dict(model_state)
model.eval()
bot_name = "Sam"

def get_response(msg):
    sentence = tokenize(msg)
    X = bag_of_words(sentence, all_words)
    X = X.reshape(1, X.shape[0])
    X = torch.from_numpy(X).to(device)

    output = model(X), predicted = torch.max(output, dim=1)
    tag = tags[predicted.item()]

    probs = torch.softmax(output, dim=1)
    prob = probs[0][predicted.item()]
    if prob.item() > 0.75:
        for intent in intents['intents']:
            if tag == intent["tag"]:
                return random.choice(intent['responses'])

    return "I do not understand..."

if __name__ == "__main__":
    print("Let's chat! (type 'quit' to exit)")
    while True:
        # sentence = "do you use credit cards?"
        sentence = input("You: ")
        if sentence == "quit":
            break

```

```
resp = get_response(sentence)
print(resp)
```

intents.json

```
{
  "intents": [
    {
      "tag": "greeting",
      "patterns": [
        "Hi",
        "Hey",
        "How are you",
        "Is anyone there?",
        "Hello",
        "Good day"
      ],
      "responses": [
        "Hey :-)",
        "Hello, thanks for visiting",
        "Hi there, what can I do for you?",
        "Hi there, how can I help?"
      ]
    },
    {
      "tag": "goodbye",
      "patterns": ["Bye", "See you later", "Goodbye"],
      "responses": [
        "See you later, thanks for visiting",
        "Have a nice day",
        "Bye! Come back again soon."
      ]
    },
    {
      "tag": "thanks",
      "patterns": ["Thanks", "Thank you", "That's helpful", "Thank's a lot!"],
```

```

    "responses": ["Happy to help!", "Any time!", "My pleasure"]
  },
  {
    "tag": "courses",
    "patterns": [
      "courses",
      "Which courses do you have?",
      "do you offers BE courses",
      "do you offers BTech courses",
      "What kinds of courses are there?"
    ],
    "responses": [

```

```

"BE.CSE\nBE.MECH\nBE.CIVIL\nBE.ECE\nB.Tech.IT\nB.Tech.AIDS\nB.Tech.CSBS
"

```

```

  ]
},
{
  "tag": "course list",
  "patterns": [
    "Do you have cse?",
    "Do you have mech?",
    "Do you have civil?",
    "Do you have ece?",
    "Do you have aids?",
    "Do you have csbs?",
    "Do you have Computer Science Engineering?",
    "Do you have Electronics and Communications Engineering?",
    "Do you have Electronics and Communications Engineering?",
    "Do you have construction investigation various information levelling",
    "Do you have Mechanical Engineering?",
    "Do you have Artificial Intelligence and Data Science?",
    "Do you have Computer Science and Business Systems?"
  ],
  "responses": [
    "Yes"
  ]
},
{
  "tag": "transport",

```

```

"patterns": [
  "does transport is having on this collage?",
  "does bus facility available",
  "transport",
  "bus",
  "bus facility available for mettupalayam",

  "bus facility available for SATHYAMANGALAM",
  "bus facility available for SINGANALLUR",
  "bus facility available for VADAVALLI",
  "bus facility available for PERUR",
  "bus facility available for SIVANANDA COLONY",
  "bus facility available for PERUR",
  "bus facility available for TIRUPUR",

  "bus facility available for KANUVAPALAYAM(THEAKKAMPATTI)"

```

```

],

```

```

"responses": [
  "yes"
]

```

```

},

```

```

{

```

```

  "tag": "mtp",
  "patterns": [
    "bus route for mettupalayam",
    "bus route for karamadai",
    "bus route for sirumugai",
    "bus route for Alangombu",
    "bus route for Vellankanni",
    "bus route for Mathampalayam Pirivu",
    "bus route for Periyamathampalayam",
    "bus route for ctc"

```

```

],

```

```

"responses": [

```

```

  "KG Denim\nBank Stop\nTheater Medu\nFour Corner\nAlangombu\nAlangombu
Colony\nBangalaMedu\nMeenachi\nCTC\nVellankanni\nKuttaiyur\nGandhiNagar\nTeacher
s Colony\nMathampalayam\nKaramadai\nKGISL"

```

```

]

```

```

},

```

```

{

```

```

  "tag": "SATHYAMANGALAM",

```

```

"patterns": [
    "bus route for Sathyamangalam",
    "bus route for SRT Corner",
    "bus route for VRT Mill",
    "bus route for Puliampatti",
    "bus route for Sellappampalayam",
    "bus route for Annur",
    "bus route for Kovilpalayam",
    "bus route for Kariyaampalayam",
    "bus route for Coffee Kadai"
],
"responses": [
    "Sathyamangalam\nSRT Corner\nPuliampatti\nFour
Corner\nSellappampalayam\nAnnur\nKariyaampalayam\nKovilpalayam\nKGISL"
]
},
{
    "tag": "SINGANALLUR ",
    "patterns": [
        "bus route for Chinnampalayam",
        "bus route for Ondipudur",
        "bus route for Singanallur",
        "bus route for Hopes",
        "bus route for Peelamedu",
        "bus route for Laksmi Mills"
    ],
    "responses": [
        "Chinnampalayam\nIrugur\nOndipudur\nSinganallur\nESI\nHopes\nPeelamedu\nLaksm i-
Mills\nKGISL"
    ]
},
{
    "tag": "VADAVALLI",
    "patterns": [
        "bus route for BharathiarUniversity",
        "bus route for Kavundampalayam",
        "bus route for Martin",
        "bus route for Edayarpalayam",

```

```

    "bus route for Venkatapuram"
  ],
  "responses": [
    "Bharathiar-University\nAlamaram\nVadavalli-School\nPN-
Pudur\nVenkatapuram\nEdayarpalayam\nTVS-Nagar\nSaravana-
Nagar\nKavundampalayam\nCheran-Colony\nMartin-Apartment\nKGISL"
  ]
},
{
  "tag": "PERUR",
  "patterns": [
    "bus route for Pooluvapatti",
    "bus route for Kalampalayam",
    "bus route for Perur",

    "bus route for Telungupalayam",
    "bus route for Selvapuram",
    "bus route for Sukruvarpettai",
    "bus route for Gandhi Park",
    "bus route for RS Puram",
    "bus route for Flower Market",
    "bus route for North Coimbatore",
    "bus route for Murugan Mills",
    "bus route for Saibaba Kovil",
    "bus route for Thudiyalur"
  ],
  "responses": [
    "Pooluvapatti\nKalampalayam\nPerur\nTelungupalayam\nLIC-
Colony\nSelvapuram\nSukruvarpettai\nGandhi-Park\nRS-Puram\nFlower-
Market\nNorth-Coimbatore\nMurugan-Mills\nSaibaba-Kovil\nThudiyalur\nKGISL"
  ]
},
{
  "tag": "SIVANANDA COLONY",
  "patterns": [
    "bus route for Sivananda colony",
    "bus route for Lakshmipuram",
    "bus route for Ganapathi",

```



```

    "bus route for SuriyaHospital",
    "bus route for BharathiNagar",
    "bus route for AmmanKovil"
  ],
  "responses": [
    "Sivananda-colony\nLakshmipuram\nGanapathi\nSuriya-Hospital\nBharathi-
Nagar\nAmman-Kovil\nKGISL"
  ]
},
{
  "tag": "KANUVAPALAYAM",
  "patterns": [
    "bus route for BadrakaliammanKovil",
    "bus route for Theakkampatti ",
    "bus route for Presscolony",
    "bus route for VeerapandiPirivu",
    "bus route for Jothipuram",
    "bus route for Vannan-kovil",
    "bus route for Pricol",
    "bus route for Periyanaickenpalayam",
    "bus route for LMW",
    "bus route for Narasimanaickenpalayam",
    "bus route for RakkipalyamPirivu",
    "bus route for NGGO-Colony",
    "bus route for Vadamadurai"
  ],
  "responses": [
    "Badrakaliamman-Kovil\nKanuvapalyam\nKaramadai\nPress-colony\nVeerapandi-
Pirivu\nJothipuram\nVannan-kovil\nPricol\nPn-palayam\nLMW\nNSN-
Palayam\nRakkipalyam\nNGGO-Colony\nVadamadurai\nKGISL"
  ]
},
{
  "tag": "KANUVAPALAYAM",
  "patterns": [
    "bus route for Kanakkampalayam",
    "bus route for KANUVAPALAYAM",
    "bus route for Shanthi-Theater ",
    "bus route for Nataraj-Theater",

```

```

        "bus route for Kumar-Nagar",
        "bus route for Anupparpalayam",
        "bus route for Ammapalayam",
        "bus route for Poondi",
        "bus route for Avinashi",
        "bus route for Attaiyampalayam",
        "bus route for Karuvalur"

    ],

    "responses": [

        "Kanakampalayam\nAnna-Nagar\nShanthi-Theater\nNataraj-Theater\nRailway-Station\nAnupparpalayam\nAmmalayam\nPoondi\nAvinashi\nAttaiyampalayam\nKaruvalur\nKGISL"

    ]

}

]

}

```

model.py

```

import torch

import torch.nn as nn

class NeuralNet(nn.Module):

    def __init__(self, input_size, hidden_size, num_classes):
        super(NeuralNet, self). __init__()
        self.l1 = nn.Linear(input_size, hidden_size)
        self.l2 = nn.Linear(hidden_size, hidden_size)
        self.l3 = nn.Linear(hidden_size, num_classes)
        self.relu = nn.ReLU()

    def forward(self, x):
        out = self.l1(x)
        out = self.relu(out)
        out = self.l2(out)
        out = self.relu(out)
        out = self.l3(out)
        # no activation and no softmax at the end
        return out

```

nltk_utils.py

```
import numpy as np
import nltk
# nltk.download('punkt')
from nltk.stem.porter import PorterStemmerstemmer = PorterStemmer()

def tokenize(sentence):
    return nltk.word_tokenize(sentence)

def stem(word):
    return stemmer.stem(word.lower())

def bag_of_words(tokenized_sentence, words):
    sentence_words = [stem(word) for word in tokenized_sentence]
    bag = np.zeros(len(words), dtype=np.float32)
    for idx, w in enumerate(words):
        if w in sentence_words:
            bag[idx] = 1

    return bag
```

train.py

```
import numpy as np
import nltk
from nltk.stem.porter import PorterStemmer
stemmer = PorterStemmer()

def tokenize(sentence):
    return nltk.word_tokenize(sentence)

def stem(word):
    return stemmer.stem(word.lower())

def bag_of_words(tokenized_sentence, words):
    sentence_words = [stem(word) for word in tokenized_sentence]
    bag = np.zeros(len(words), dtype=np.float32)
    for idx, w in enumerate(words):
        if w in sentence_words:
            bag[idx] = 1

    return bag
```

```

all_words = []
tags = []
xy = []

for intent in intents['intents']:
    tag = intent['tag']
    tags.append(tag)
    for pattern in intent['patterns']:
        w = tokenize(pattern)
        all_words.extend(w)
        xy.append((w, tag))

ignore_words = ['?', '!', '!']

all_words = [stem(w) for w in all_words if w not in ignore_words]
all_words = sorted(set(all_words))
tags = sorted(set(tags))

print(len(xy), "patterns")
print(len(tags), "tags:", tags)
print(len(all_words), "unique stemmed words:", all_words)

X_train = []
y_train = []
for (pattern_sentence, tag) in xy:

    bag = bag_of_words(pattern_sentence, all_words)
    X_train.append(bag)
    label = tags.index(tag)
    y_train.append(label)

X_train = np.array(X_train)
y_train = np.array(y_train)
num_epochs = 1000
batch_size = 8
learning_rate = 0.001 input_size = len(X_train[0])hidden_size = 8
output_size = len(tags)
print(input_size, output_size)

class ChatDataset(Dataset):
    def __init__(self):
        self.n_samples = len(X_train)
        self.x_data = X_train
        self.y_data = y_train

```

```

def __getitem__(self, index):
    return self.x_data[index], self.y_data[index]

def __len__(self):
    return self.n_samples

dataset = ChatDataset()
train_loader = DataLoader(dataset=dataset,
                           batch_size=batch_size,
                           shuffle=True,
                           num_workers=0)

device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')
model = NeuralNet(input_size, hidden_size, output_size).to(device)
criterion = nn.CrossEntropyLoss()
optimizer = torch.optim.Adam(model.parameters(), lr=learning_rate)

for epoch in range(num_epochs):
    for (words, labels) in train_loader:
        words = words.to(device)
        labels = labels.to(dtype=torch.long).to(device)
        outputs = model(words)

        # if y would be one-hot, we must apply
        # labels = torch.max(labels, 1)[1]
        loss = criterion(outputs, labels)

        optimizer.zero_grad()
        loss.backward()
        optimizer.step()

    if (epoch+1) % 100 == 0:
        print(f'Epoch [{epoch+1}/{num_epochs}], Loss: {loss.item():.4f}')
    print(f'final loss: {loss.item():.4f}')
data = {
    "model_state": model.state_dict(),
    "input_size": input_size,
    "hidden_size": hidden_size,
    "output_size": output_size,
    "all_words": all_words,
    "tags": tags

```

```
}
```

```
FILE = "data.pth"  
torch.save(data, FILE)  
print(f'training complete. file saved to {FILE}')
```

Website source:

Index.html:

```
<!DOCTYPE html>  
  
<html>  
  
<head>  
  <link rel="stylesheet" href="style.css" type="text/css">  
</head>  
  
<body>  
  <div id="main">  
    <div class="navbar">  
      <a></a>  
      <a href="#">HOME</a>  
      <a href="https://www.kgkite.ac.in/academics/">ACADEMICS</a>  
      <a href="https://www.kgkite.ac.in/admission/">ADMISSION</a>  
      <a href="https://www.kgkite.ac.in/placements/about-the-training-and-  
placement/">PLACEMENT</a>  
      <a href="https://ecampus.kgisliim.ac.in/ecampus/">ECAMPUS</a>  
      <a href="https://www.kgkite.ac.in/about-us/">ABOUT US</a>  
    </div>  
    <div id="banner">  
        
    </div>  
    <div id="content1">  
      <h4 class="heading">Smart India Hackathon</h4>  
      <p class="para1" align="justify">SIH 2022 brings the next generation evolution by  
inclusion of new methodology to inculcate the culture of startup and innovation
```

ecosystem across different age groups

</p>

READ MORE

</div>

<div id="content2">

<h4 class="heading">Six Days International Tech Talk Series</h4>

<p class="para1" align="justify">The Department of Computer Science and Engineering and Artificial Intelligence & Data Science is organizing Six Days International Tech Talk Series </p>

READ MORE

</div>

<div id="content3">

<h4 class="heading">CSE & AI departments partner with NoviTech</h4>

<p class="para1" align="justify">The CSE and AI & DataScience Departments of KGiSL Institute of Technology are excited to partner with NoviTech.</P>

READ MORE

</div>

<div class="f-social">


```
</div>
</div>
</nav>
</body>
</html>
```

Style.css

```
.navbar
{
    overflow: hidden;
    background-color: #333;
    position: fixed;
    top: 0;

    width: 82.5%;
}

.navbar a
{
    float: left;
    display: block;
    color: #f2f2f2;
    text-align: center;
    padding: 15px 30px;
    text-decoration: none;
    font-size: 18px;
}

.navbar a:hover
{
    background: #ddd;
    color: black;
}

.main
{
    padding: 16px;
    margin-top: 65px;
    height: 1500px; /* Used in this example to enable scrolling */
    width: 70%;
```



```

    margin :0px auto;
}
#logo
{
    display: inline-block;
    vertical-align: top;
    width: 50px;
    height: 20px;
    margin-right: 0px;
    margin-top: 0px; /* if you want it vertically middle of the navbar. */
}
body
{
    background-color: white;
    margin-left: 120px;
    margin-right: 120px;
}
#header{
    height:80px;
    background-color: rgb(255, 255, 255);
}
#header ul{
    float:right;
    margin-top: 20px;
}
#header ul li{
    display: inline;
    margin-right: 20px;
}
#header ul li a{
    color: #ff0000;

    text-decoration: none;
    font-size: 18px;
}
#banner img{
    width:100%;
    padding-bottom: 20px;
    margin-top: 40px;
}

```

```

}

#content1{
    width: 32%;
    height: 350px;
    background-color: #7300ff;
    float: left;
    margin-right: 2%;
}

#content2{
    width: 32%;
    height: 350px;
    background-color: rgb(255, 0, 25);
    float: left;

}

#content3{
    width: 32%;
    height: 350px;
    background-color: rgb(0, 38, 255);
    float: right;
}

.heading{
    color: white;
    margin-left: 20px;
    font-size: 28px;
}

.para1{
    color: white;
    margin-left: 20px;
    margin-right: 8px;
}

.more{
    text-decoration: none;
    color: black;
    background-color: white;
    padding: 8px;
    margin-left: 20px;
}

#bottom1
{

```

```

margin-top: 15px;
/* width:100%; */

/* background-color: rgb(0, 0, 0); */
float: left;
font-family: 'Lucida Sans';
}

#bottom2
{
margin-top: 15px;
width:28%;

background-color: rgb(255, 255, 255);
float: left;
}

#bottom1 h1
{
/* color: black; */

/* margin-left:20px; */
font-family: 'Lucida Sans';
text-align: center;
}

#bottom p{
color: rgb(255, 255, 255);
margin-left: 20px;
margin-right: 8px;
}

.f-social
{
overflow: hidden;
top: 50px;
width: 100%;
}

.f-social a
{
float: right;

```

```
display: block;
text-align: center;
padding: 10px 90px;

text-decoration: none;
font-size: 18px;
}
```

10.2 SCREENSHOTS

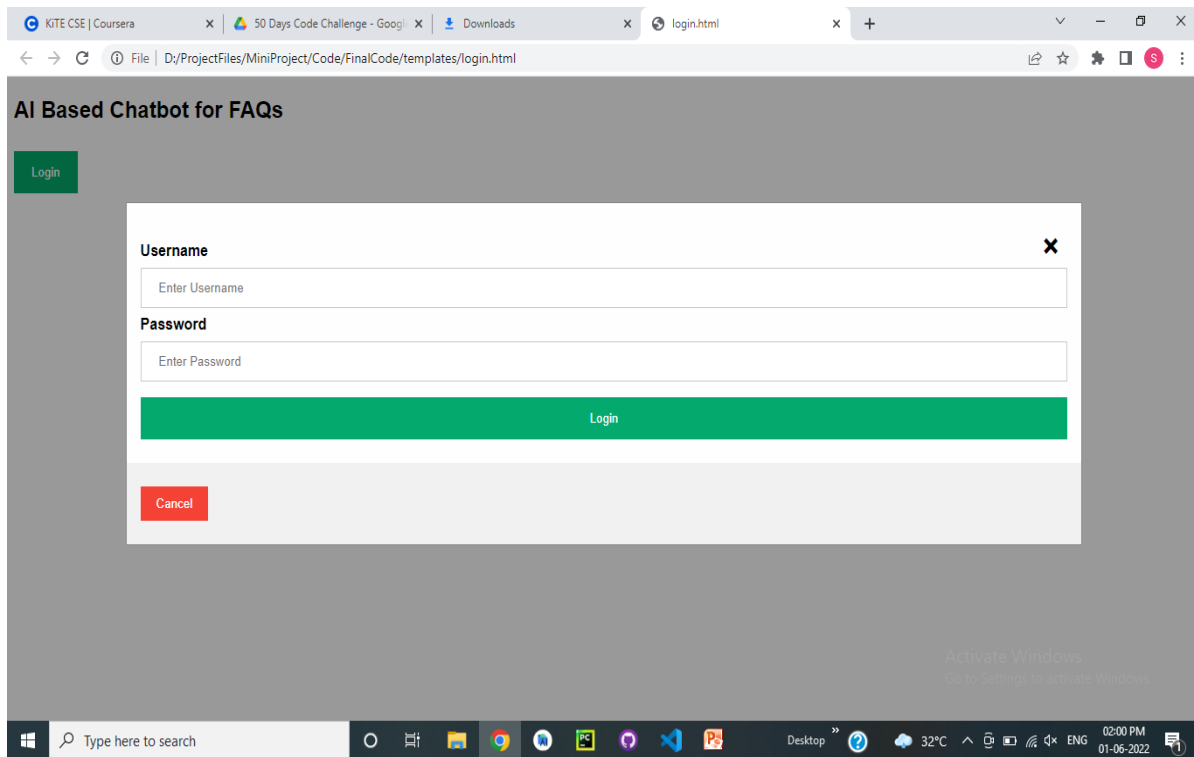


Figure 10.2.1 Intro screen

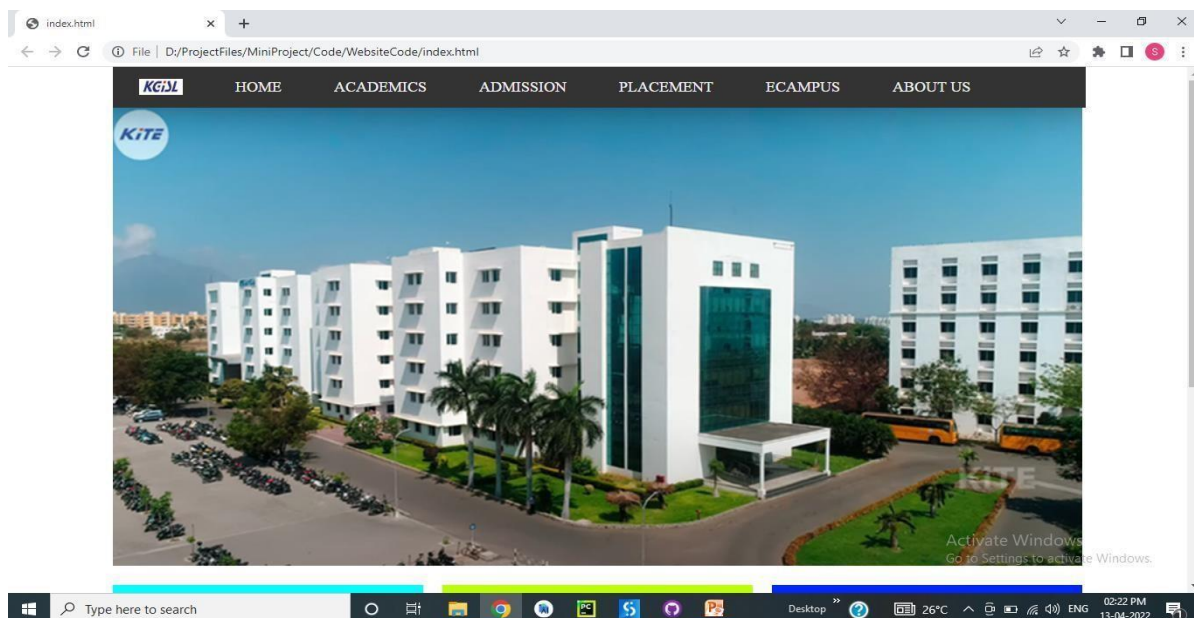


Figure 10.2.2 Home screen

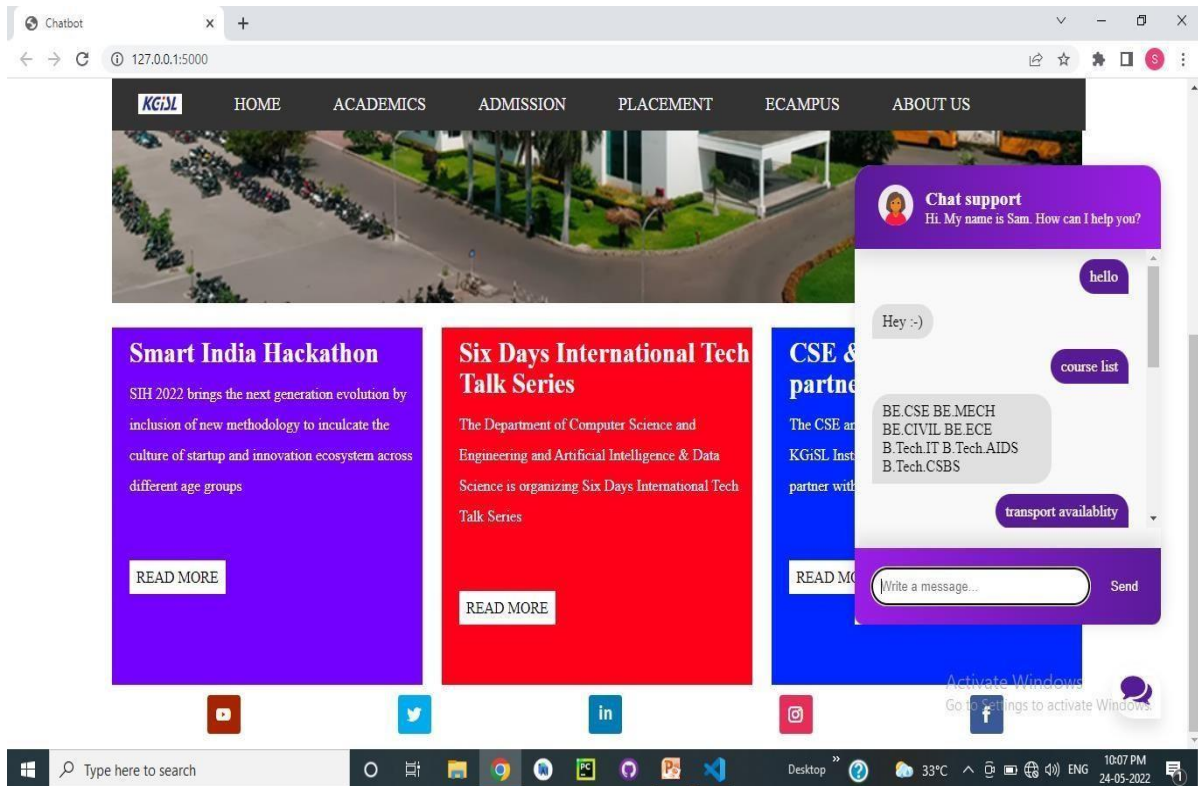


Figure 10.2.3. Manage users screen

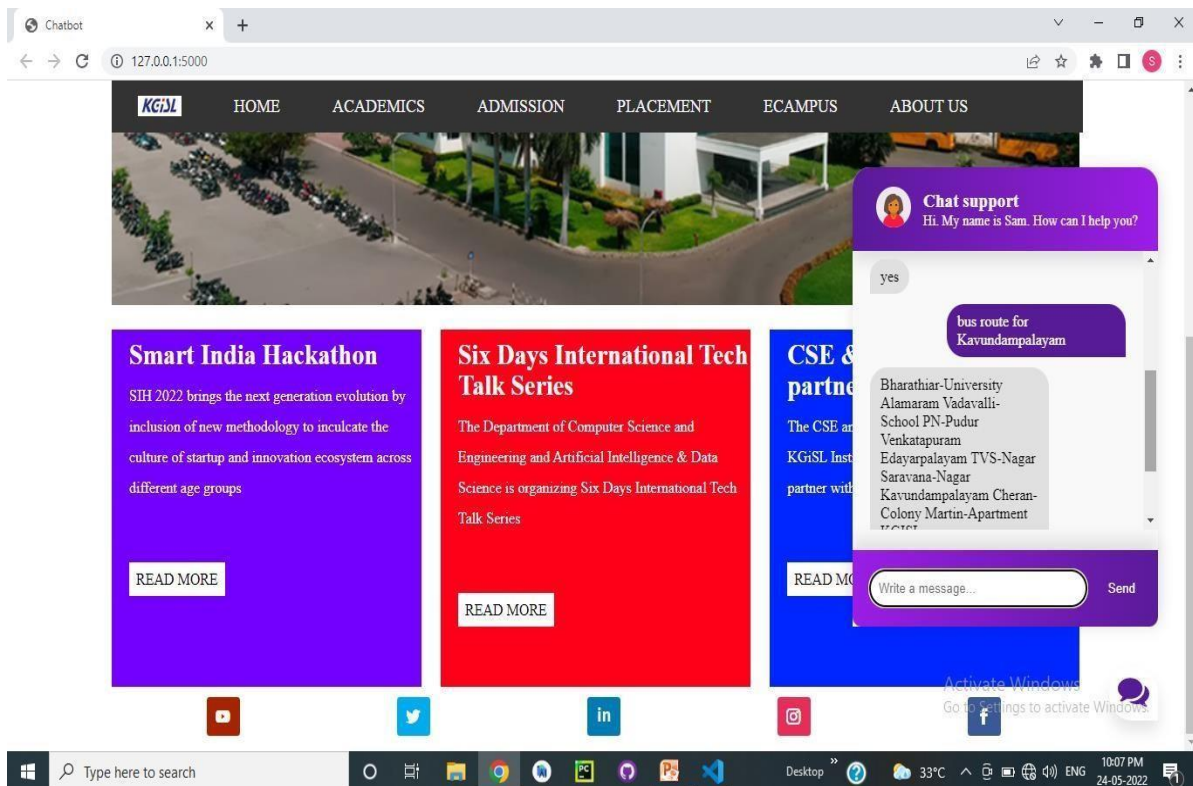


Figure 10.2.4 Chatbot screen

CHAPTER 11

REFERENCES

- [1] Ajinkya Huddar; Chaitanya Bysani; Chintan Suchak; Uttam D Kolekar; Kaushiki Upadhyaya: “Dexter the College FAQ Chatbot”
- [2] Naing Naing Khin; Khin Mar Soe: “University Chatbot using Artificial Intelligence Markup Language”
- [3] Neelkumar P. Patel; Devangi R. Parikh; Darshan A. Patel; Ronak R. Patel: “AI and Web-Based Human-Like Interactive University Chatbot (UNIBOT)”
- [4] Heru Agus Santoso; Nurul Anisa Sri Winarsih; Edy Mulyanto; Galuh Wilujeng saraswati; Septian Engg: “Dinus Intelligent Assistance (DINA) Chatbot for University Admission Services”
- [5] Bhavika R. Ranoliya_, Nidhi Raghuwanshi_ and Sanjay Singh: “Chatbot for University Related FAQs”
- [6] Shafquat Hussain, Omid Ameri Sianaki, and Nedat Ababneh: “A Survey on Conversational Agents/Chatbots Classification and Design Techniques”
- [7] Shahriare Satu, Hasnat Parvez and Shamim-AI-Mamun: “Review of integrated applications with AIML based chatbot”
- [8] Gwo-Jen Hwang & Ching-Yi Chang: “A review of opportunities and challenges of chatbots in education”
- [9] Tatwadarshi P. Nagarhalli, Vinod Vaze and N. K. Rana: “A Review of Current Trends in the Development of Chatbot Systems”
- [10] AM Rahman, Abdullah Al Mamun and Alma Islam: “Programming challenges of Chatbot: Current and Future Prospective”