

A Thesis/Project/Dissertation Report

on

CHATBOT ASSISTANT

*Submitted in partial fulfillment of the
requirement for the award of the degree of*

B.TECH IN

COMPUTER SCIENCE AND ENGINEERING



(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

**Under The Supervision of
Dr. Ajay Shanker Singh
(Professor)**

Submitted By

**Aryan Gautam 19SCSE1010060
Mahima Singh 19SCSE1010743**

**SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA
INDIA
DECEMBER, 2021**



**SCHOOL OF COMPUTING SCIENCE AND
ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA**

CANDIDATE'S DECLARATION

I/We hereby certify that the work which is being presented in the “**CHATBOT ASSISTANT**” in partial fulfillment of the requirements for the award of the Bachelor of Technology submitted in the School of Computing Science and Engineering of Galgotias University, Greater Noida, is an original work carried out during the period of **Aug, 2021 to Dec, 2021**, under the supervision of **Dr. Ajay Shanker Singh**, Department of Computer Science and Engineering of School of Computing Science and Engineering , Galgotias University, Greater Noida.

The matter presented in the thesis/project/dissertation has not been submitted by me/us for the award of any other degree of this or any other places.

Aryan Gautam, 19SCSE1010060

Mahima Singh, 19SCSE1010743

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Dr. Ajay Shanker Singh

CERTIFICATE

The Final Thesis/Project/ Dissertation Viva-Voce examination of **Aryan Gautam 19SCSE1010060, Mahima Singh 19SCSE1010743** has been held on _____ and his/her work is recommended for the award of Bachelor of Technology

Signature of Examiner(s)

Signature of Supervisor(s)

Signature of Project Coordinator

Signature of Dean

Date:

Place: Greater Noida

ACKNOWLEDGEMENT

First and foremost, I am profoundly thankful to _____ (**Head Of Department, Computer Science and Engineering**), **Galgotias University, Greater Noida India**, for allowing this paper to be my own work.

I would also like to thank our project supervisor **Dr. Ajay Shanker Singh** for guiding me and sharing their precious time and knowledge with patience. Without their participation and contribution, this project would not have been successful.

Finally I want to express my immense gratitude to my friends and my parents for their endless encouragement and support through all these years. Without whom this project would have been a distant reality.

Thank you.

ABSTRACT

As we all know that nowadays everything is becoming modern and we still have different types of problems but we still suffer from them. There are different types of problems regarding personal assisting software, which helps us in improving our personal life problems. In our day-to-day life We are facing different kinds of problems such as time management and work management, but we can't rely on any untrusted applications to fulfil our requirements.

To overcome this situation we finally get on a conclusion , that what we have to develop for helping our own problems, the solution is the software which can take your command and help you to overcome the situation, the software helps us to give command to the device and can assist us by fulfilling the requirements of us, This is basically a software which can help you to assist with the new modern technology present in the market.

The sample application is developed using Python Kernel and XML's Artificial Intelligence Markup Language(AIML) along with a database file which stores the name, e-mail, and encrypted password to tell the GPA of a student. It is accessed using Firebase. The interface of this app is designed using tkinter(python's GUI toolkit).

What we can conclude from the above situation and usage is that greater the database and more the models and use cases, the better is the reaction produced for the client. In any case, the issues are many. For adjusting more uses, the scope changes from the investigation of AI to language examination. We additionally need to recollect that we are taking a shot at a cell phone. The NLP is very broad and maybe utilizing them on a server and isolating this application into customer and server side application can fathom the speed issue as when we do that the speed of the program won't be restricted to the equipment.

CONTRIBUTION OF INDIVIDUAL TEAM MEMBER

Name of the Student(s)	Enrollment/Admission No.	Contributions
Aryan Gautam	19SCSE1010060	Analysis, Coding
Mahima Singh	19SCSE1010743	Testing, Documentation

List of Figures

Figure No.	Figure Name	Page No.
fig-1.1	<i>screenshot of register window</i>	17
fig-1.2	<i>screenshot of login window</i>	18
fig-1.3	<i>screenshot of chatbot window</i>	18
fig-2.1	<i>screenshot of authentication from database</i>	19
fig-2.2	<i>screenshot of realtime database</i>	20
fig-3.1	<i>screenshot of user registration</i>	21
fig-3.2	<i>screenshot of login authentication</i>	21
fig-3.3	<i>screenshot of chatbot of sample input1</i>	22
fig-3.4	<i>screenshot of chatbot of sample input2</i>	22
fig-3.5	<i>screenshot of chatbot of sample input3</i>	22
fig-3.6	<i>screenshot of chatbot of sample input4</i>	23

Table of Contents

	Candidate's Declaration	I
	Certificate	II
	Acknowledgement	III
	Abstract	IV
	Contribution of Individual Team Members	V
	List of Figures	VI
1.	INTRODUCTION	1-8
	1.1 Introduction 1.2 Formulation of Problem 1.2.1 Tools and Technology Used	
2.	LITERATURE SURVEY/ PROJECT DESIGN	9-13
	2.1 Preface 2.2 Existing System 2.3 Proposed System 2.4 Feasibility Study	
3.	SYSTEM ANALYSIS AND DESIGN	14-28
	3.1 Requirement Specification 1.1. Functional Requirement 1.2. Non-Functional Requirement 3.2 Class Diagram 3.3 Sequence Diagram 3.4 Use-Case Diagram 3.5 Data Flow Diagram 3.6 Algorithm and Pseudo Code	
4.	RESULTS AND DISCUSSIONS	29-39
	4.1 User Interface 4.2 Databases	

	4.3 Module Description	
	4.4 Results and Output	

5.	CONCLUSION AND FUTURE SCOPE	40
	5.1 Conclusion	
	5.2 Future Scope	
6.	TAKE-AWAY FROM THE PROJECT	41
	6.1 Bibliography	

INTRODUCTION

1.1 INTRODUCTION

Artificial Intelligence (A.I) has grown in popularity for simulating conversions between bots and humans, particularly on mobile platforms. The functionality of these chatbots ranges from utilitarian to entertainment, but the value is often not clearly defined. The purpose and need for these chatbots are often not clearly defined. Curiosity and interest may spark an initial interaction with a chatbot, but to add more value to ongoing interactions we should identify a broadly acceptable role that has a defined purpose. What a chatbot is, and how to use one effectively, are new concepts that many struggle to define. A great deal of additional research which should lead to continued improvement in the effectiveness of chatbots in the years to come. The Chatbot has a very bright future because in recent years we are going to see that it will become very common as a website. And it is not that costly so anyone who has a website can afford it. As the prevalence of chat bots in society has reached a new high. Most of the studies on chatbots is using different algorithms and how to create an advanced chatbot. This study is mainly dependent on expert personnel's results or any software or applications. Chat bots can reach out to a large audience on messaging apps and be more effective than humans. They may develop into a capable information-gathering tool in the near future. The aim of the present studies is to create a chat bot with different features, and information regarding different algorithms based on natural language processing. After the growth of Artificial Intelligence, it has made the conversion smooth between humans and bots particularly on mobile platforms as well as desktop apps. Chatbots can be used to solve different problems from known to unknown. The purpose of the chatbot is not clearly defined. But the curiosity and need of the people make the people design chatbots according to their need. In the coming years, chatbots will be common to every website and apps. And it is affordable so anyone can design it using python and

other technology as it is open source and free. There are many ways to design chatbots. First one is rule based chatbot and the other is AI based. So we have designed this chatbot using AI. To represent this technique we need an interface so we have used python's package named tkinter. This interface will take input from the user and display output on it. For various problems we can have some data which needs to be trained to predict and give output. This data needs to be trained from time to time to make chatbot work smoother. The aim of the chatbot is to solve students' queries as it will provide instant response and reduce cost and crowd gathering at help desk at any organizations. One of the most important tasks was the development of an effective chatbot. It is a difficult task and it also led to the evaluation of the NLP (natural language processing). And after the evaluation of the NLP it became easy for the machine to understand the human sentences as the sentence provided by the human may be unstructured in nature which can lead the machine in difficulty to understand it. But there are some challenges, chatbots are not able to solve all user's problems. To solve these problems it requires more and more data and to train the model with these data using tensorflow. A chatbot is an artificial intelligence (AI) software that can simulate a conversation (or a chat) with a user in natural language through messaging applications, websites, mobile apps or through the telephone. Why are chatbots important? A chatbot is often described as one of the most advanced and promising expressions of interaction between humans and machines. However, from a technological point of view, a chatbot only represents the natural evolution of a Question Answering system leveraging Natural Language Processing (NLP). Formulating responses to questions in natural language is one of the most typical Examples of Natural Language Processing applied in various enterprises' end-use applications.

1.2 FORMULATION OF PROBLEM

All the companies want their clients to know about them and they can get more projects or sell their products. So the most primary way to do this is to have a good interactive website in the advanced world of digital marketing. But in this advanced world, just having a website is not enough as it does not cover all the details and can not solve the queries of the clients. Maybe a comment section is

an answer, but it's tedious work, and 'time is money' your client will not wait too long for your response. And you can not assign humans to answer everyone at the same time twenty-four by seven. So for that, you need a chatbot, not a normal one "An Intelligent ChatBot" with Artificial Intelligence technology.

“Artificial intelligence chatbot is a technology that makes interactions between man and machines using natural language possible. A chatbot can give different responses from the same input given by the user according to the current conversation issue". By using our "Intelligent ChatBot" you can overcome all the above-given issues, you do not need humans to do manual work, your clients will be happy. A chatbot is a conventional agent that is capable of communicating with operators by using natural languages. As numerous chatbot platforms already exist, there are still some problems in building data-driven systems because a huge amount of data is required for their development. After the growth of Artificial Intelligence, it has made the conversion smooth between humans and bots particularly on mobile platforms as well as desktop apps. Chatbots can be used to solve different problems from known to unknown. The purpose of the chatbot is not clearly defined. But the curiosity and need of the people make the people design chatbots according to their need. In the coming years, chatbots will be common to every website and apps. And it is affordable so anyone can design it using python and other technology as it is open source and free. There are many ways to design chatbots. First one is rule based chatbot and the other is AI based. So we have designed this chatbot using AI. To represent this technique we need an interface so we have used python's package named tkinter. This interface will take input from the user and display output on it. For various problems we can have some data which needs to be trained to predict and give output. This data needs to be trained from time to time to make chatbot work smoother. The aim of the chatbot is to solve students' queries as it will provide instant response and reduce cost and crowd gathering at help desk at any organizations. One of the most important tasks was the development of an effective chatbot. It is a difficult task and it also led to the evaluation of the NLP (natural language processing). And after the evaluation of the NLP it became

easy for the machine to understand the human sentences as the sentence provided by the human may be unstructured in nature which can lead the machine in difficulty to understand it. But there are some challenges, chatbots are not able to solve all user's problems. To solve these problems it requires more and more data and to train the model with these data using tensorflow. Chatbots are basically AI intelligence bots which can interact with the user or customers depending upon the usage. It is an application of Artificial Intelligence and Machine Learning. Now-a-days technology is increasing rapidly. In this technological world every industry is trying to automate things to provide better services. One of the great applications of automation would be chatbots.

There are basically two types of Chatbots :

Command based: Chatbots that function on predefined rules and can answer only limited queries or questions. Users need to select an option to determine their next step.

Intelligent/AI Chatbots: Chatbots that leverage Machine Learning and Natural Language Understanding to understand the user's language and are intelligent enough to learn from conversations with their users. You can converse via text, speech or even interact with a chatbot using graphical interfaces.

Here in this project we created an AI Chatbot which is focused for Cyber Security Industry and trying to solve the use cases of industry by training it in such a way that it can answer some of the basic queries of cyber security as well other than just the customer services. Building a chatbot can sound daunting, but it's totally doable. AI Chatbot Framework is an AI powered conversational dialog interface built in Python. With this tool, it's easy to create Natural Language conversational scenarios with no coding efforts whatsoever. The smooth UI makes it effortless to create and train conversations to the bot and it continuously gets smarter as it learns from conversations it has with people. AI Chatbot Framework can live on any channel of your choice (such as Messenger, Slack etc.) by integrating it's API with that platform.

You don't need to be an expert at artificial intelligence to create an awesome chatbot that has AI capabilities. With this boilerplate project you can create an AI

powered chatting machine in no time. There may be scores of bugs. So feel free to contribute via pull requests. Many chatbots have been developed that provide a multitude of services through a wide range of methods. A chatbot is a brand-new conversational agent in the high speed changing technology world. With the advance of Artificial Intelligence and machine learning, chatbots are becoming more and more popular. A chatbot is the extension of human interface mediums such as the phone and social platforms. Similarly, Cryptocurrency is a new extension of digital or virtual currency designed to work as a medium of exchange. In the current digital exchange world, investors and interested parties are eager to know more information about, and the capabilities of, this new type of currency. One of the potential paths to retrieve the info automatically and quickly is through a chatbot. The interaction in the format of speech or text between humans and computers is gaining more and more in popularity nowadays. People expect to have similar experiences when they talk to machines as when they talk to human beings. In order to provide suitable responses based on phrases or keywords taken from questions, a dialogue system or program is needed, which is often called a chatbot or a chatterbot. The chatbot is a computer program that has an ability to communicate with people by providing answers to questions. People input natural language speech or text, while the program provides the most feasible intelligent response in the form of text or speech. The improvements in the fields of inter-networking and information technology have been intricate in executing Artificial Intelligent (AI) systems. These systems are drawing nearer to human activities, for example, choosing emotionally supportive networks, robotics, natural language processing, and so forth. Indeed, even in the artificial intelligence fields, there are some hybrid strategies and adaptive techniques that make increasingly complex techniques. That, yet these days there are additionally several Natural Language Processing (NLP) and intelligent systems that could comprehend human language. Artificial intelligent systems learn themselves and retrieve insight by perusing required electronic articles that have existed on the web. A chatbot (otherwise called a chatterbox, Bot, or Artificial Conversational Entity) is an AI program that copies human discussions including content and communication in natural language utilizing artificial intelligence methods, for example, Natural Language Processing (NLP), picture

and video processing, and voice analysis. A Chatbot for college management system has been created utilizing artificial intelligence algorithms that examine the user queries. This chatbot system is an internet application that gives an answer to the broken down queries of a user. Users simply need to choose the classification for inquiries and afterward ask the question to the bot that uses it for noting it. Artificial intelligence has been incorporated to respond to the user's inquiries. Then the user can procure the fitting solutions to their inquiries. The appropriate responses are given utilizing artificial intelligence algorithms. Users won't need to actually go to the college or college website for requests. Users need to enlist to the system and need to login to the system. After login users can get to the different helping pages. There will be different helping pages through which users can chat by asking questions related to college activities. The system answers to users' queries with the assistance of effective Graphical User Interface (GUI). The user can ask about the college related activities with the assistance of this web application. College related activities, for example, admissions, academics, Intake, and other social activities. It will support the undergraduates/other users to be refreshed about the college activities. A chatbot is an Artificial Intelligence program that can converse with people in natural language, the manner in which we collaborate with one another. It can trade a human for some undertakings of replying inquiries. A chatbot is a specialist that assists users in utilizing natural language. It worked as an endeavor to trick people. A few uses of chatbots, for example, User care, customer support and so on utilizes Artificial Intelligence Markup Language to visit with users. One of the foremost objectives of chatbots is to take after a smart human and entangle the recipient of the discussion to comprehend the genuine working along with different designs and abilities for their use has generally widened. These chatbots can adequately trick the user to believe that they are "talking" to an individual, however, they are limited in improving their insight base at runtime, and have typically next to zero methods for keeping track of all the discussion information. Chatbots utilize AI to arrive at counterfeit intelligence, helping them to comprehend the user's question, and what's more, give a suitable reaction. The chatbots are created utilizing the Artificial Intelligence Markup Language (AIML) for imparting or cooperating with the user. This comprises software that

will be made up of utilizing Artificial Intelligence and will assist the user in chatting with a machine. The user can ask the systems like typically did to other humans. The remainder of the paper is as follows: section-II provides literature survey and section-III presents proposed systems with methodology. Section-IV propounds results and discussion and finally, section-V concludes the paper.

The purpose of this project is to showcase the power of chatbots and how they can be an alternative to using an application or even a website. The chatbots should be easy to use, respond in a timely fashion and be all round user friendly. The bots should make the user's interaction as easy and fast as possible to ensure that the user's time is not wasted and that they get what they want without any difficulty or misunderstanding from the bot. The conversation should flow and always keep the user in control of the conversation. Users should come away from their experience with the chatbot and think that it was a fun, easy to use and straightforward interaction that would encourage them to come back without any hesitation. With messaging platforms being the most used type of application in the world, businesses will be looking to take advantage of this and start to develop their own - 8 - chatbots to work along with their social media pages. For example, a person calling a restaurant to see what time they open at or what is the special today, the customer can simply message the page on Facebook and the bot will respond accordingly. This frees up time for real employees to do other work and allows the chatbot to handle the simple tasks. Since users will already have a messaging app installed on their mobile device, there is no need to download a separate application to use the chatbot. This can turn a lot of users away as nowadays there is a plethora of applications available and most users will be fed up of having to download an application that they may only use once or twice.

1.2.1 Tools and Technology Used

- Python 3.8.10
- Nltk - Natural Language Toolkit
- Tkinter GUI
- Pytsx3
- Tensorflow
- Firebase

LITERATURE SURVEY/ PROJECT DESIGN**2.1 Preface**

Chat bots, or conversational interfaces as they are also known, present a new way for individuals to interact with computer systems. Traditionally, to get a question answered by a software program involves using a search engine, or filling out a form. A chat bot allows a user to simply ask questions in the same manner that they would address a human. The most well-known chat bots currently are voice chat bots: Alexa and Siri. However, chat bots are currently being adopted at a high rate on computer chat platforms. In literature, there are many other approaches related to this topic. In one of the research work we have seen that it was a web based e-learning platform where students ask questions and teachers answer it at a later stage. If there are more queries then it leads to delay of time. In this work pandora were used where a bot stores question and answer at later stage using XML style of language i.e. AIML (Artificial Intelligence Markup Language). This AI-bot is trained in such a way that when it is unable to give a response to any question a human interacts with the user to answer the question. Niranjana M. et al. discussed an approach using Bayesian theory to match the request of students and fit out the right response. In particular, Bot accept the user's (student) answer and extract the keyword from the question using a lexical parser then the keyword is being compared with the database. The Bayesian probabilities get all categories from the list. Once the categories are selected keywords are compared using Bayesian theory. The answer to the particular question is selected which is highest is set to the text to speech conversion module and the student receives the answer as voice. Nenkov N. et al. have investigated the understanding of intelligent agents on platform IBM Bluemix through IBM Watson technology. These Chatbots have to automate the exchange between the student and the teacher with the frames of Moodle, the learning management system. Watson is a logical system that connects the capabilities of NLP(Natural Language Processing), analytics, and machine learning techniques. In these cases, a Chatbot alongwith the Facebook Messenger is realized to make simple communication between a teacher and a student. In

literature, there are many other approaches related to this topic. In one of the research work we have seen that it was a web based e-learning platform where students ask questions and teachers answer it at a later stage. If there are more queries then it leads to delay time. In this work pandora were used where a bot stores question and answer at later stage using XML style of language i.e. AIML (Artificial Intelligence Markup Language). This AI-bot is trained in such a way that when it is unable to give a response to any question a human interacts with the user to answer the question. Niranjana M. et al. discussed an approach using Bayesian theory to match the request of students and fit out the right response. In particular, bot accept the user's (student) answer and extract the keyword from the question using a lexical parser then the keyword is being compared with the database. The Bayesian probabilities get all categories from the list. Once the categories are selected keywords are compared using Bayesian theory. The answer to the particular question is selected, which is highest is set to the text to speech conversion module and the student receives the answer by voice. Nenkov N. et al. have investigated the understanding of intelligent agents on platform IBM Bluemix through IBM Watson technology. These Chatbots have to automate the exchange between the student and the teacher with the frames of Moodle, the learning management system. Watson is a logical system that connects the capabilities of NLP(Natural Language Processing), analytics, and machine learning techniques. In these cases, a Chatbot alongwith the Facebook Messenger is realized to make simple communication between a teacher and a student.

By utilizing the field of Artificial Intelligence, one can develop numerous applications, one of which is mentioned in this paper is a college chatbot system. In spite of the fact that chatbots can be deployed in various fields like marketing, education, banking, clinical and finance. Research is being done in making the regular rule based chatbots to be informative, responsive and complete the correspondence in a conversational human language. This requires the incorporation of Natural Language Processing (NLP) and Machine Learning (ML) technologies into the college chatbot system. There are various approaches to do as such. Selecting a fitting technique depends on the area of the chatbot, the

functionalities it expects to give, the language of correspondence, the end client, and so forth. Some of the approaches are versed in this literature survey.

Michael Maudlin created "Chatter Bot Algorithm" in 1994 and published in the book Julia and was used to answer the queries. Taking this initial idea, further projects were developed to create a chatbot system. The user needs to login to the Chat-Bot application. At exactly that point the user is permitted to submit complaints and queries. When a user query is submitted to the bot, the context of the query is recognized and NLP is applied. WordNet calculation [4] and grammatical forms labeling are utilized to distinguish the feeling of the words. User questions are checked in the knowledge database. If the appropriate response is discovered, at that point that answer is sent to that user. If a particular query isn't found in the database such inquiries are replied by the administrator. When the administrator answers the query, at exactly that point the appropriate response is sent to the user. Questions alongside answers are put in the database so that at whatever point such inquiries will be posed with the intention that they get addressed legitimately from the database. Because of this, the administrator doesn't have to address the same query physically any longer. Different algorithms such as Porter Stemmer Algorithm is used for expelling suffixes from words in English. Word request vector process is used for estimating word requests closeness between two sentences.

2.2 Existing System

There are many chatbots available nowadays. Some are designed by either linguistic (rule-based chatbots) or machine learning (AI chatbot) models. Different chatbots have different works depending upon the companies and their requirements.

2.3 Proposed System

This chatbot is designed for the college students who will login into the system and chatbot will ask their subject name and return the result accordingly. And it will also solve students' general queries. This College Chatbot System is a web based application which gives responses to the user queries. The system architecture of

the chatbot system. Firstly, Chatbot responds to the user by greeting him/her and then asks the user to login into the system by providing his/her mail. Then the user finds the buttons in the UI which correspond to the different categories of the college. After going through the buttons the chatbot system asks the user if it is helpful in giving the response. If the user is not able to find the required response he/she can continue the chat with the college chatbot system by briefly elaborating their queries. Then the chatbot system applies Machine Learning algorithms to break down the user queries. Once the user asks a query, the keywords in the query are detected using WordNet Algorithm. As the query description can change from one person to another person. The same query may be asked in a different way by the users. One user asks a query so simply and clearly while another user may request the same query in a completely different manner. So it is required to find what is the exact information the user seeks to know and to find a correct response for the corresponding user query. The chatbot system firstly removes the stop words from the user input, if they are present in the queries asked by the user. After removing the stop words from the user queries, tokenization and lemmatization process are done. Tokenization is a process of taking a set of text or text and breaking it up into its individual words or sentences.

Lemmatization is the process of gathering the different inflected forms of a word so they can be dissected as a solitary item and is a variation of stemming. From there spell checker is used to identify and rectify spelling mistakes present in the query, then by using the sentence similarity and WordNet Algorithm a suitable response is explored in the knowledge database. WordNet is a semantic and lexical database for the English language. It is used to group English words into the set of synonyms called synsets, it gives short definitions and utilization models, and records various relations among these synonym sets or their individuals. If the response is found in the database it is displayed to the user, else the system notifies the admin about missing response in the database and gives a predefined response to the user. Admin can write the missing response into the database by logging into the admin block in the website so that if the user asks the same query next time, he/she may get the suitable response. At the end of conversation the college chatbot system collects the feedback from users to improve the system efficiency.

2.4 Feasibility Study

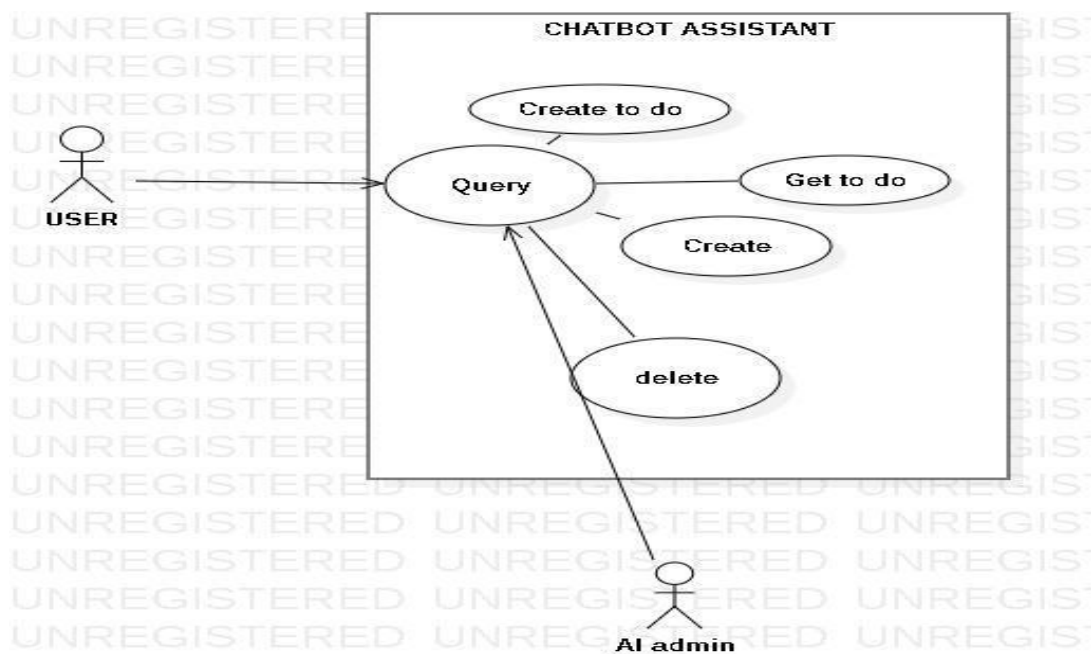
Designing this project is feasible as the libraries used for it are completely free and open source. Final software will be delivered as a desktop application which will be completely free and open source, anyone from our college can access it. All the new queries will be updated from time to time to train the model and make the chatbot work smoother.

SYSTEM ANALYSIS AND DESIGN**3.1 Requirement Specification****3.1.1 Functional Requirement**

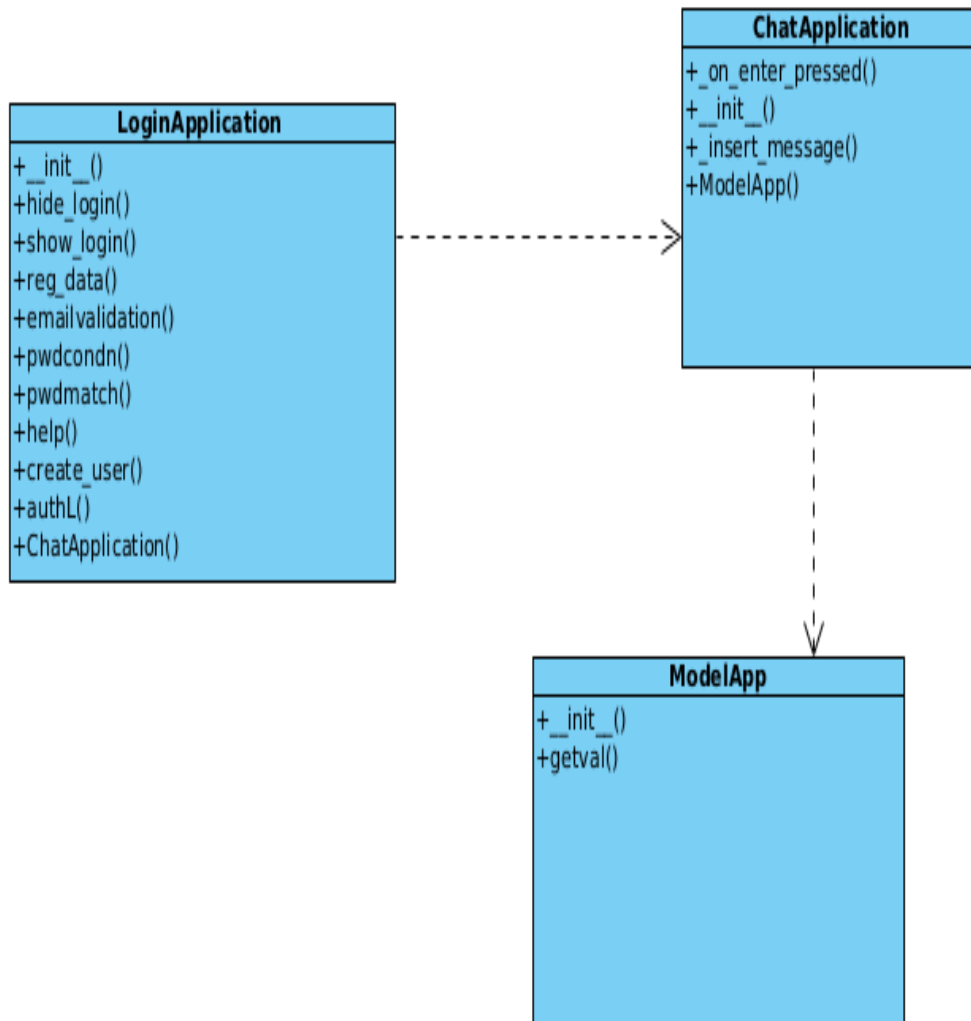
- **Registration**
 - App shall record user's name
 - App shall record user's enrollment no.
 - App shall record user's email
 - App will redirect to login page
- **Chatbot**
 - App shall save general query of student
 - App will reply to student's query
 - All will display GPA of asked subject

3.1.2 Non-Functional Requirement

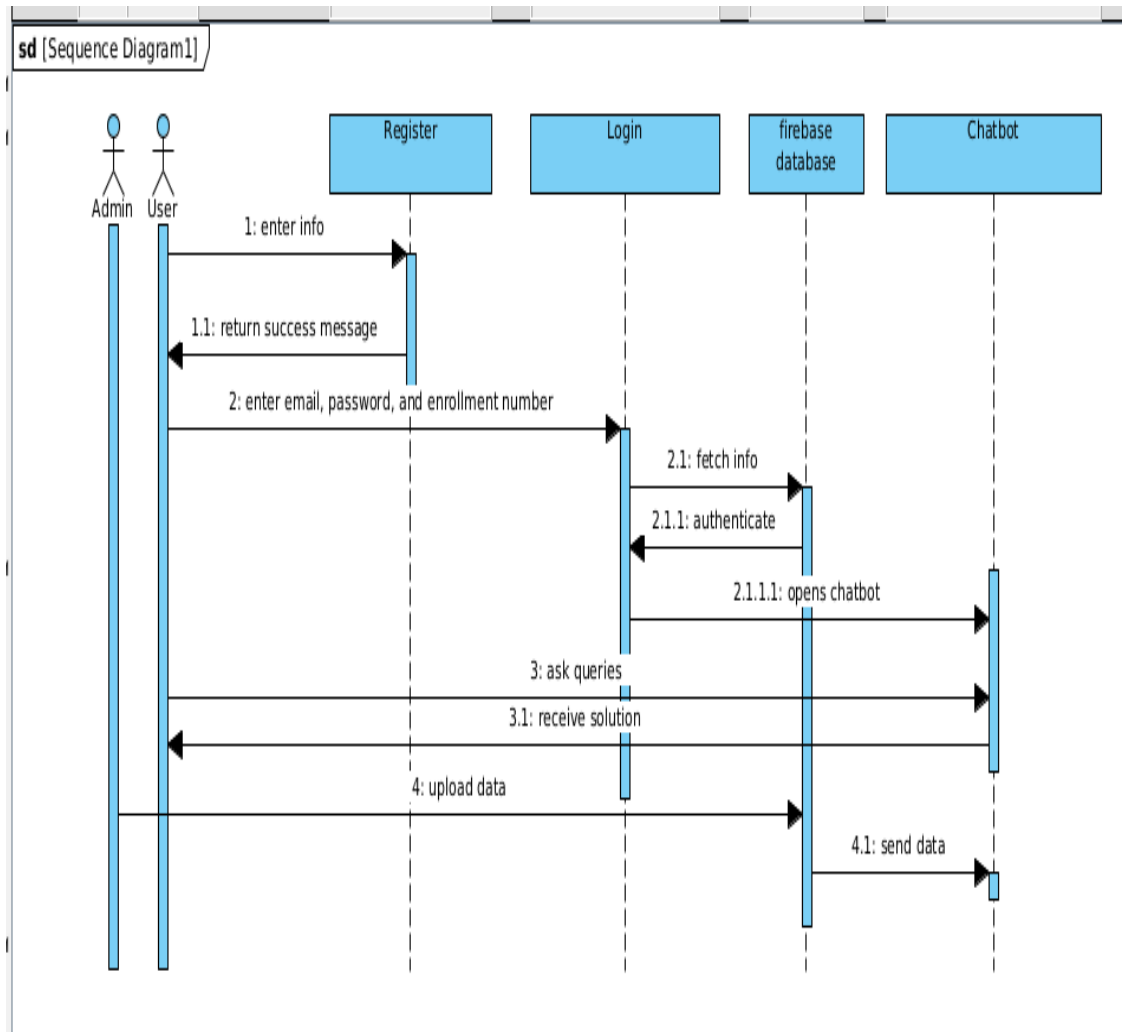
- **Security Requirement**
 - This app is safe to use.
 - Data stored in our database is safe and protected
 - Password stored will be encrypted
- **Safety Requirement**
 - This app requires an internet connection to fetch data from the database.
 - This app need some space to download this application

3.2 Use Case Diagram

3.3 Class Diagram

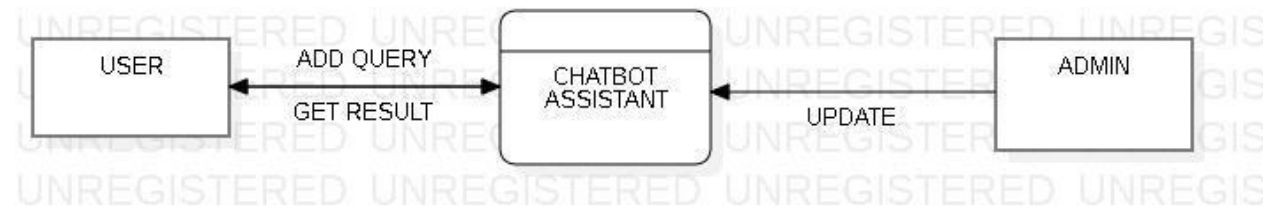


3.4 Sequence Diagram

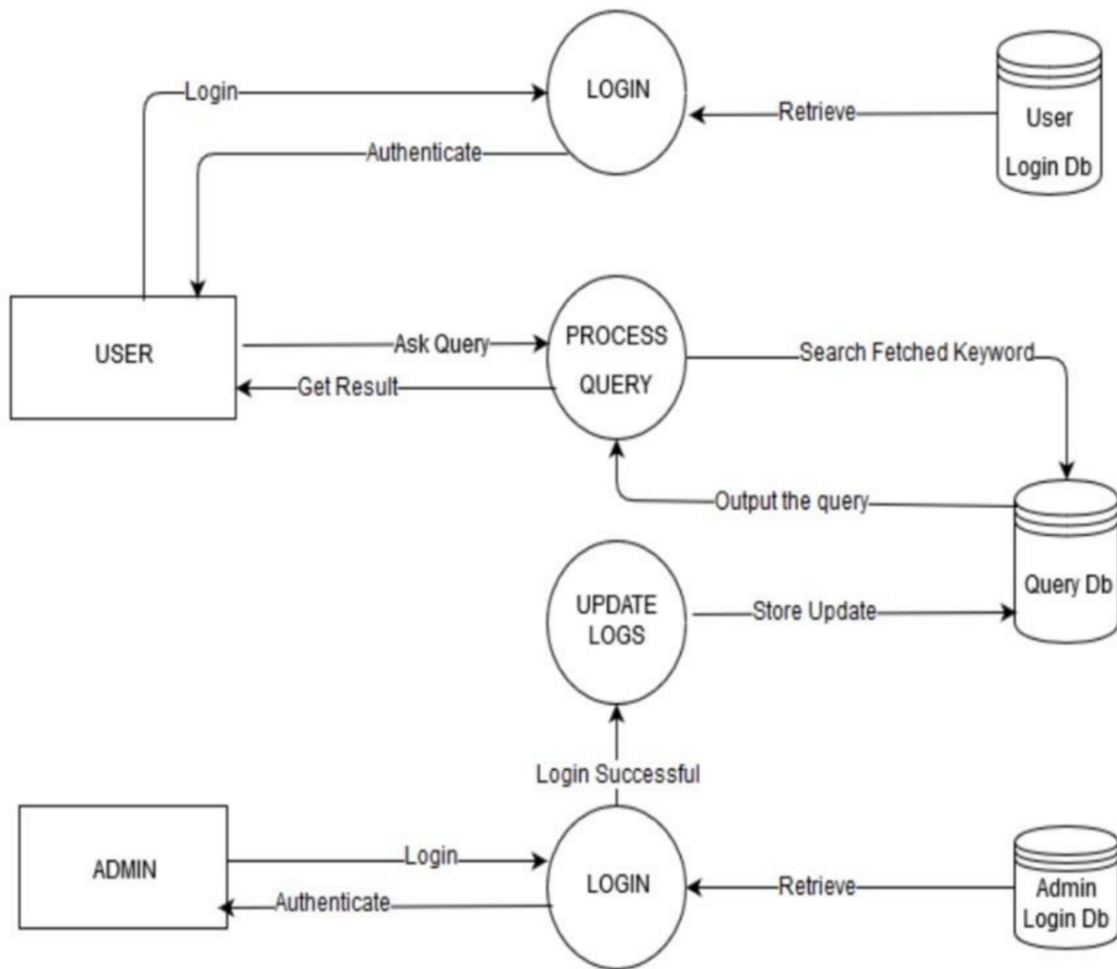


3.5 Use-Case Diagram

Level 0 DFD



Level 1 DFD



3.6 Algorithm and Pseudo Code

- Main.py

```
main.py x
1  from tkinter import *
2  from tkinter import messagebox
3  import tkinter as tk
4  import os
5  from PIL import ImageTk, Image
6  from firebase import firebase
7  import firebase_admin
8  from firebase_admin import credentials, firestore
9  from screeninfo import get_monitors
10 import pyautogui
11 import pyrebase
12 from app import ChatApplication
13
14 class LoginApplication:
15
16     def __init__(self, window):
17         self.window = window
18
19         self.window.title("Student Login")
20         self.window.resizable(width=False, height=False)
21         w, h = pyautogui.size()
22         self.window.configure(width=w, height=h)
23
24         # -----Main background image-----
25         self.path1 = os.path.normpath("images//main_bg.png")
26         bg1 = Image.open(self.path1)
27         self.image1 = ImageTk.PhotoImage(bg1)
28         canvas1 = Canvas(self.window, width=w, height=h)
29         canvas1.pack(fill="both", expand=True)
30         canvas1.create_image(0, 0, image=self.image1, anchor="nw")
31
32         self.f1 = Frame(self.window, width=500, height=600)
33         self.f1.place(x=650, y=40)
34
35         # -----Login background image-----
36         self.path2 = os.path.normpath("images//login_bg.png")
37         bg2 = Image.open(self.path2)
38         self.image2 = ImageTk.PhotoImage(bg2)
39         canvas2 = Canvas(self.f1, width=500, height=600)
40         canvas2.pack(fill="both", expand=True)
41         canvas2.create_image(0, 0, image=self.image2, anchor="nw")
42
43         firebaseConfig = {
44             "apiKey": "AIzaSvAntkBuRiBYLR9vTKttSezRrMf0RtwnIXo".
```

8 selection regions

```
main.py x
44     "apiKey": "AIzaSyAntkBuRiBYLR9yTKttSezRrMf0RtwmIXo",
45     "authDomain": "chatbot-4dfb1.firebaseio.com",
46     "databaseURL": "https://chatbot-4dfb1-default-rtdb.firebaseio.com",
47     "projectId": "chatbot-4dfb1",
48     "storageBucket": "chatbot-4dfb1.appspot.com",
49     "messagingSenderId": "333338334205",
50     "appId": "1:333338334205:web:b40e26cdf2b30c0db33f94",
51     "measurementId": "G-P7EPM3MHDB"
52 }
53
54 firebase = pyrebase.initialize_app(firebaseConfig)
55 self.auth = firebase.auth()
56 firebase = pyrebase.initialize_app(firebaseConfig)
57 self.auth = firebase.auth()
58 self.db = firebase.database()
59 self.store = firebase.storage()
60
61 # ===== Photo Icon =====
62 self.path3 = os.path.normpath("images//i1.png")
63 self.path4 = os.path.normpath("images//i2.png")
64 self.path3_e = os.path.normpath("images//i3.png")
65 load1 = Image.open(self.path3)
66 load2 = Image.open(self.path4)
67 load1_e = Image.open(self.path3_e)
68 render1 = ImageTk.PhotoImage(load1)
69 render2 = ImageTk.PhotoImage(load2)
70 render3 = ImageTk.PhotoImage(load1_e)
71
72 title1 = Label(self.f1, text="Login",
73               font="Lucid 30 bold", fg='#004aad')
74 title1.place(x=150, y=15)
75
76 f2 = Frame(self.f1, width=50, height=50, bd=0)
77 f2.place(x=50, y=150)
78 l1 = Label(f2, image=render1)
79 l1.image = render1
80 l1.place(x=0, y=0)
81
82 f3 = Frame(self.f1, width=50, height=50, bd=0)
83 f3.place(x=50, y=300)
84 l2 = Label(f3, image=render2)
85 l2.image = render2
86 l2.place(x=0, y=0)
87
```

```
main.py x
87
88     f2_e = Frame(self.f1,width=50, height=50, bd=0)
89     f2_e.place(x=50, y=220)
90     ll_e = Label(f2_e, image=render3)
91     ll_e.image=render3
92     ll_e.place(x=0,y=0)
93
94     # ===== input for Enrollment number and password =====
95     self.e1 = Entry(self.f1, font="lucid 15 normal")
96     self.e1.place(x=110, y=150, height=50)
97     self.enroll = Entry(self.f1,font="lucid 15 normal")
98     self.enroll.place(x=110,y=220,height=50)
99     self.e2 = Entry(self.f1, font="lucid 15 normal", show="*")
100    self.e2.place(x=110, y=300, height=50)
101
102    # ===== getting email and password and authenticating =====
103    btns = Button(self.f1, text='Authenticate',
104                 font="lucid 20", bg='#004aad', fg='white',command=self.authL)
105    btns.place(x=130, y=390)
106
107    btnr = Button(self.f1, text='New here?? go to Registration Page',
108                 font="lucid 10", bd=0, command=lambda: self.hide_login())
109    btnr.place(x=100, y=460)
110
111    def hide_login(self):
112        self.f1.place_forget()
113        self.f4 = Frame(self.window, width=450, height=600)
114        self.f4.place(x=650, y=40)
115        canvas3 = Canvas(self.f4, width=450, height=600)
116        canvas3.pack(fill="both", expand=True)
117        canvas3.create_image(0, 0, image=self.image2, anchor="nw")
118        title = Label(self.f4, text="Register Here",
119                    font="Lucid 30 bold", fg='#004aad').place(x=100, y=10)
120        first_name = Label(self.f4, text="First Name",
121                          font="Lucid 15", fg='black').place(x=25, y=90)
122        last_name = Label(self.f4, text="Last Name",
123                          font="Lucid 15", fg='black').place(x=25, y=140)
124        email_id = Label(self.f4, text="Email Address",
125                        font="Lucid 15", fg='black').place(x=25, y=190)
126        enroll_no = Label(self.f4, text="Enrollment No.",
127                          font="Lucid 15", fg='black').place(x=25, y=240)
128        password = Label(self.f4, text="Password",
129                         font="Lucid 15", fg='black').place(x=25, y=290)
130        cpassword = Label(self.f4, text="Confirm Password".
```

Line 44, Column 77

```
File Edit Selection Find View Goto Tools Project Preferences Help
main.py x
126 enroll_no = Label(self.f4, text="Enrollment No.",
127                      font="Lucid 15", fg='black').place(x=25, y=240)
128 password = Label(self.f4, text="Password",
129                      font="Lucid 15", fg='black').place(x=25, y=290)
130 cpassword = Label(self.f4, text="Confirm Password",
131                      font="Lucid 15", fg='black').place(x=25, y=340)
132
133 self.fnameE = Entry(self.f4, font=(
134     'calibre', 10, 'normal'), relief=GROOVE, bd=2)
135 self.fnameE.place(x=230, y=90)
136 self.lnameE = Entry(self.f4, font=(
137     'calibre', 10, 'normal'), relief=GROOVE, bd=2)
138 self.lnameE.place(x=230, y=140)
139
140 self.emailE = Entry(self.f4, font=(
141     'calibre', 10, 'normal'), relief=GROOVE, bd=2)
142 self.emailE.place(x=230, y=190)
143 self.enrollE = Entry(self.f4, font=(
144     'calibre', 10, 'normal'), relief=GROOVE, bd=2)
145 self.enrollE.place(x=230, y=240)
146
147 self.pwdE = Entry(self.f4, show="*", font=('calibre',
148     10, 'normal'), relief=GROOVE, bd=2)
149 self.pwdE.place(x=230, y=290)
150
151 self.cnfpwdE = Entry(
152     self.f4, show="*", font=('calibre', 10, 'normal'), relief=GROOVE, bd=2)
153 self.cnfpwdE.place(x=230, y=340)
154
155 self.phelp = Button(self.f4, text="?", font=('calibre', 10, 'bold'), relief=GROOVE, bd=0, bg='#004aad',
156     fg='white', command=self.help)
157 self.phelp.place(x=140, y=290)
158
159 btn1 = Button(self.f4, text="Submit", bd=2, relief=RAISED,
160     fg='#004aad', command=self.reg_data).place(x=200, y=410)
161 btn1 = Button(self.f4, text="Already have an account?? go to Login Page",
162     font="lucid 10", bd=0, command=self.show_login)
163 btn1.place(x=100, y=470)
164
165 def show_login(self):
166     self.f4.place_forget()
167     self.f1.place(x=650, y=40)
168
169 def req_data(self):
```

```
main.py x
168
169
170     def reg_data(self):
171         self.e = 0
172         self.p1 = 0
173         self.p2 = 0
174         self.p3 = 0
175         if self.fnameE.get() == "" or self.lnameE.get() == "" or self.emailE.get() == "" or self.enrollE.get() == "" \
176            or self.pwdE.get() == "" or self.cnfpwdE.get() == "":
177             messagebox.showinfo(
178                 "Error", "All fields are required", parent=self.window)
179         else:
180             self.emailValidaton()
181             self.pwdcondn()
182             self.pwdmatch()
183             if self.e==1 and self.p1==1 and self.p2==1:
184                 self.create_user()
185                 messagebox.showinfo("congrats", "User Created! You data is getting ready..please login after 15 minute",parent=self.window)
186                 data={"First name ":self.fnameE.get(),
187                       "Last name ":self.lnameE.get(),
188                       "Email ":self.emailE.get(),
189                       "Enrollment Number ":self.enrollE.get()}
190                 self.db.child('intents').child(self.enrollE.get()+"_"+self.emailE.get().replace(".", "_")).child("Info").set(data)
191                 self.db.child('intents').child(self.enrollE.get()+"_"+self.emailE.get().replace(".", "_")).child("Chat_data").set("")
192
193     def emailValidaton(self):
194         regex = "^(\\w|\\.|_|\\-)+@[\\w|\\.|_|\\-]+\\.([\\w]{2,3})$"
195         if re.search(regex, self.emailE.get()):
196             self.e = 1
197         else:
198             messagebox.showwarning(
199                 "Error", "Invalid Email", parent=self.window)
200
201     def pwdcondn(self):
202         password = self.pwdE.get()
203         flag = 0
204         while True:
205             if len(password) < 6:
206                 flag = -1
207                 break
208             elif not re.search("[a-z]", password):
209                 flag = -1
210                 break
211             elif not re.search("[A-Z]", password):
212                 flag = -1
```

Line 44, Column 77

```
File Edit Selection Find View Go Tools Project Preferences Help
main.py x
214         flag = -1
215         break
216     elif re.search("\\s", password):
217         flag = -1
218         break
219     else:
220         flag = 0
221         self.p1 = 1
222         break
223     if flag == -1:
224         messagebox.showerror("Error in password",
225                               "not a valid password", parent=self.window)
226
227     def pwdmatch(self):
228         if self.pwdE.get()==self.cnfpwdE.get():
229             self.p2=1
230         else:
231             messagebox.showerror("Error","password do not match",parent=self.window)
232
233     def help(self):
234         messagebox.showinfo("Password", "1.Minimum 8 characters.\n2.The alphabets must be between [a-z]\n3.At least\
235         one alphabet should be of Upper Case [A-Z]\n4.At least 1 number or digit between [0-9].\n", parent=self.window)
236
237     def create_user(self):
238         user = self.auth.create_user_with_email_and_password(self.emailE.get(),self.pwdE.get())
239         self.show_login()
240
241     def authL(self):
242         userL = self.auth.sign_in_with_email_and_password(self.e1.get(),self.e2.get())
243         messagebox.showinfo("wait","chatbot opens in 5 sec. please have patience.....")
244         ChatApplication(self.window,self.e1.get(),self.enroll.get())
245
246 if __name__ == "__main__":
247     loginApp = Tk()
248     app = LoginApplication(loginApp)
249     loginApp.mainloop()
250
```

Line 44, Column 77

- Model_train.py

```
model_train.py x
1 #importing the libraries
2 from firebase import firebase
3 import tensorflow as tf
4 import numpy as np
5 import pandas as pd
6 import json
7 import nltk
8 from tensorflow.keras.preprocessing.text import Tokenizer
9 from tensorflow.keras.layers import Input, Embedding, LSTM, Dense, GlobalMaxPooling1D, Flatten
10 from tensorflow.keras.models import Model
11 import matplotlib.pyplot as plt
12 from tensorflow.keras.preprocessing.text import Tokenizer
13 from tensorflow.keras.preprocessing.sequence import pad_sequences
14
15
16 class ModelApp:
17     def __init__(self, e, e_roll):
18         self.e = e
19         self.e_roll = e_roll
20         self.e = self.e.replace(".", " ")
21         child_n = self.e_roll + "_" + self.e + "/Chat_data/intents"
22         fb_app = firebase.FirebaseApplication('https://chatbot-4dfb1-default-rtdb.firebaseio.com/intents/', None)
23         result = fb_app.get(child_n, None, params={'print': 'pretty'})
24
25         self.tags = []
26         self.inputs = []
27         self.responses = {}
28         for intent in result:
29             self.responses[intent['tag']] = intent['responses']
30             for lines in intent['input']:
31                 self.inputs.append(lines)
32                 self.tags.append(intent['tag'])
33         #converting to dataframe
34         data = pd.DataFrame({"inputs": self.inputs,
35                             "tags": self.tags})
36         #removing punctuations
37         import string
38         data['inputs'] = data['inputs'].apply(lambda wrd: [ltrs.lower() for ltr in wrd if ltr not in string.punctuation])
39         data['inputs'] = data['inputs'].apply(lambda wrd: ''.join(wrd))
40
41         #tokenize the data
42         self.tokenizer = Tokenizer(num_words=2000)
43         self.tokenizer.fit_on_texts(data['inputs'])
44         self.train = self.tokenizer.texts_to_sequences(data['inputs'])

```

Line 1, Column 1

```
model_train.py x
45
46     #apply padding
47     x_train = pad_sequences(self.train)
48
49     #encoding the outputs
50     from sklearn.preprocessing import LabelEncoder
51     self.le = LabelEncoder()
52     y_train = self.le.fit_transform(data['tags'])
53
54     #input length
55     self.input_shape = x_train.shape[1]
56     #define vocabulary
57     vocabulary = len(self.tokenizer.word_index)
58     #output length
59     output_length = self.le.classes_.shape[0]
60
61     #creating the model
62     i = Input(shape=(self.input_shape,))
63     x = Embedding(vocabulary+1,10)(i)
64     x = LSTM(10,return_sequences=True)(x)
65     x = Flatten()(x)
66     x = Dense(output_length,activation="softmax")(x)
67     self.model = Model(i,x)
68     #compiling the model
69     self.model.compile(loss="sparse_categorical_crossentropy",optimizer='adam',metrics=['accuracy'])
70     #training the model
71     self.train = self.model.fit(x_train,y_train,epochs=200)
72
73     def getval(self):
74         return self.responses, self.model, self.tokenizer, self.le, self.input_shape
75
76
```

- App.py

```
app.py
1  from tkinter import *
2  import tkinter as tk
3  import os
4  import sys
5  from PIL import ImageTk, Image
6  import pyttsx3
7  import random
8  import numpy as np
9  import string
10 import re
11 from tensorflow.keras.preprocessing.sequence import pad_sequences
12 # from model_train import responses,model,tokenizer,le,input_shape
13 from tkinter import messagebox
14 from model_train import ModelApp
15
16
17 BG_GRAY = "#ABB2B9"
18 BG_COLOR = "#17202A"
19 TEXT_COLOR = "#EAECEE"
20
21 FONT = "Helvetica 14"
22 FONT_BOLD = "Helvetica 13 bold"
23
24
25 class ChatApplication:
26
27     def __init__(self,win,email,enroll_no):
28         self.window = Toplevel()
29         self.win =win
30         self.email=email
31         self.enroll_no=enroll_no
32
33         self.win.withdraw()
34
35         self.window.title("Chat")
36         self.window.resizable(width=False, height=False)
37         self.window.configure(width=600, height=400)
38         self.path4 = os.path.normpath("images//bg.png")
39         bgl = Image.open(self.path4).convert("RGB")
40         self.image1 = ImageTk.PhotoImage(bgl)
41         canvas1 = Canvas(self.window, width=600, height=400)
42         canvas1.pack(fill="both", expand=True)
43         canvas1.create_image(0, 0, image=self.image1, anchor="nw")
44
Line 1, Column 1
```



```
app.py
80 self.window.delete(3000, self.window.destroy)
87 regex = '^(\w|\.\_|\/-)+@(\w|\.\_|\/-)+[.]\w{2,3}$'
88 if(re.search(regex, prediction_input)):
89     messagebox.showinfo("Info", "You will receive marksheet in an hour...")
90
91 self.text_widget2.configure(state=NORMAL)
92 self.text_widget2.delete('1.0', END)
93 self.text_widget2.insert(END, msg1)
94 self.text_widget2.configure(state=DISABLED)
95
96 texts_p = []
97 #removing punctuation and converting to lowercase
98 prediction_input = [letters.lower() for letters in prediction_input if letters not in string.punctuation]
99 prediction_input = ''.join(prediction_input)
100 texts_p.append(prediction_input)
101 prediction_input = self.tokenizer.texts_to_sequences(texts_p)
102 prediction_input = np.array(prediction_input).reshape(-1)
103 prediction_input = pad_sequences([prediction_input], self.input_shape)
104
105 #getting output from model
106 output = self.model.predict(prediction_input)
107 output = output.argmax()
108 #finding the right tag and predicting
109 response_tag = self.le.inverse_transform([output])[0]
110
111
112 msg2 = f'{random.choice(self.responses[response_tag])}\n\n'
113 self.text_widget.configure(state=NORMAL)
114 self.text_widget.insert(END, msg2)
115 self.text_widget.configure(state=DISABLED)
116 self.text_widget.see(END)
117
```

RESULTS AND DISCUSSION**4.1 User Interface**

Screen Name	Description
Register	This window asks the user to enter their name, email, enrollment no. and password and confirm password. After data is stored. It shows the notification “user is created”.
Login	This window asks the user to enter email and password and authenticate from firebase’s realtime database. After authentication it redirects to the chatbot window.
Chatbot	It solves the query of the users from the trained model.

Table-1.1

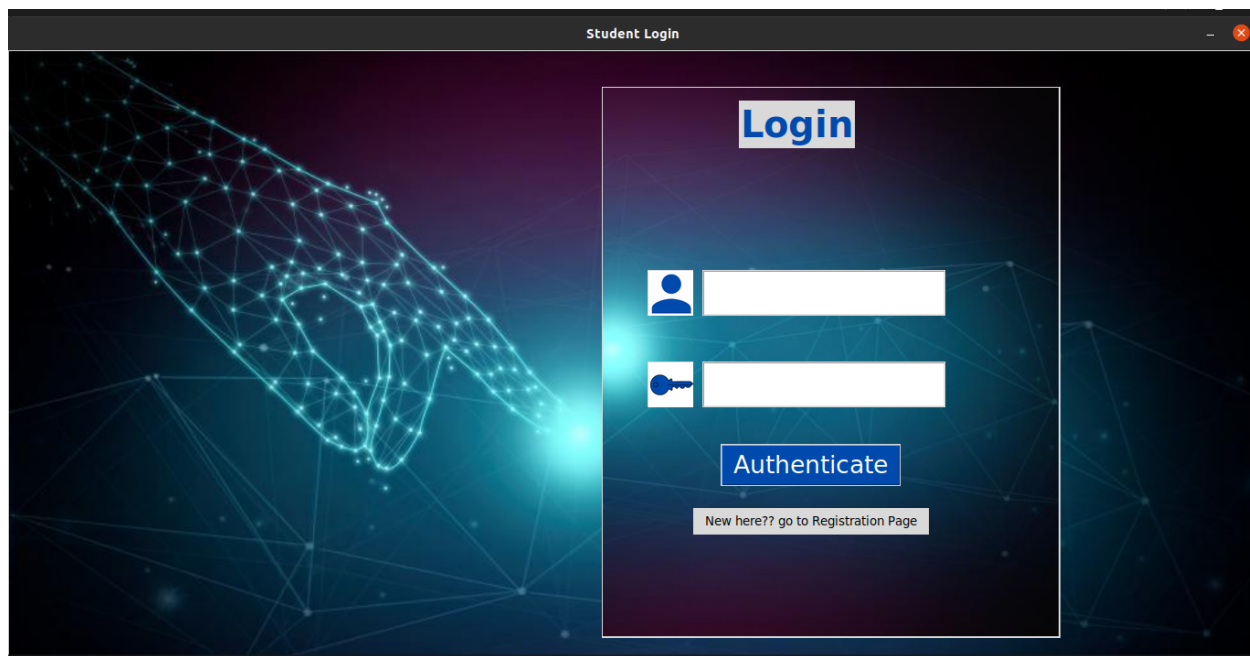
Register



The screenshot shows a web browser window titled "Student Login". The main content area features a dark blue background with a glowing, wireframe-like graphic of a hand reaching out. On the right side, there is a white-bordered box titled "Register Here" in blue text. Inside this box, there are six input fields: "First Name", "Last Name", "Email Address", "Enrollment No.", "Password" (with a small blue question mark icon), and "Confirm Password". Below these fields is a "Submit" button. At the bottom of the box, there is a link that says "Already have an account?? go to Login Page".

fig-1.1 screenshot of register window

Login



The screenshot shows a web browser window titled "Student Login". The main content area features a dark blue background with a glowing, wireframe-like graphic of a hand reaching out. On the right side, there is a white-bordered box titled "Login" in blue text. Inside this box, there are two input fields: the first is preceded by a blue person icon, and the second is preceded by a blue key icon. Below these fields is a blue "Authenticate" button. At the bottom of the box, there is a link that says "New here?? go to Registration Page".

fig-1.2 screenshot of login window

Chatbot

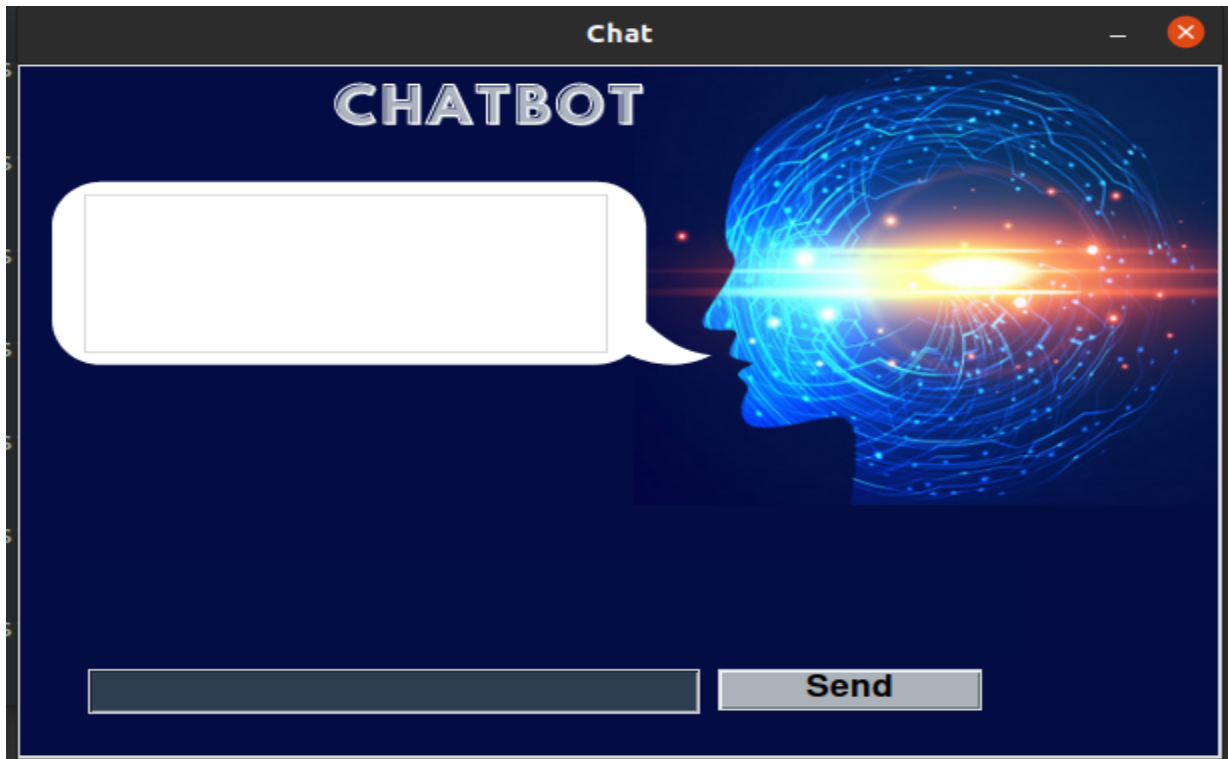


fig-1.3 screenshot of chatbot window

4.2 Database

Database used to store data of this software is Google's Firebase realtime Database. It stores and sync data in realtime. And for login authentication we have used Firebase Authentication which allows users to login into the app using sign in method email & password. When the chatbot app runs it retrieves data from firebase and trains the model and predict from it and return to the user according to their requirement.

Authentication

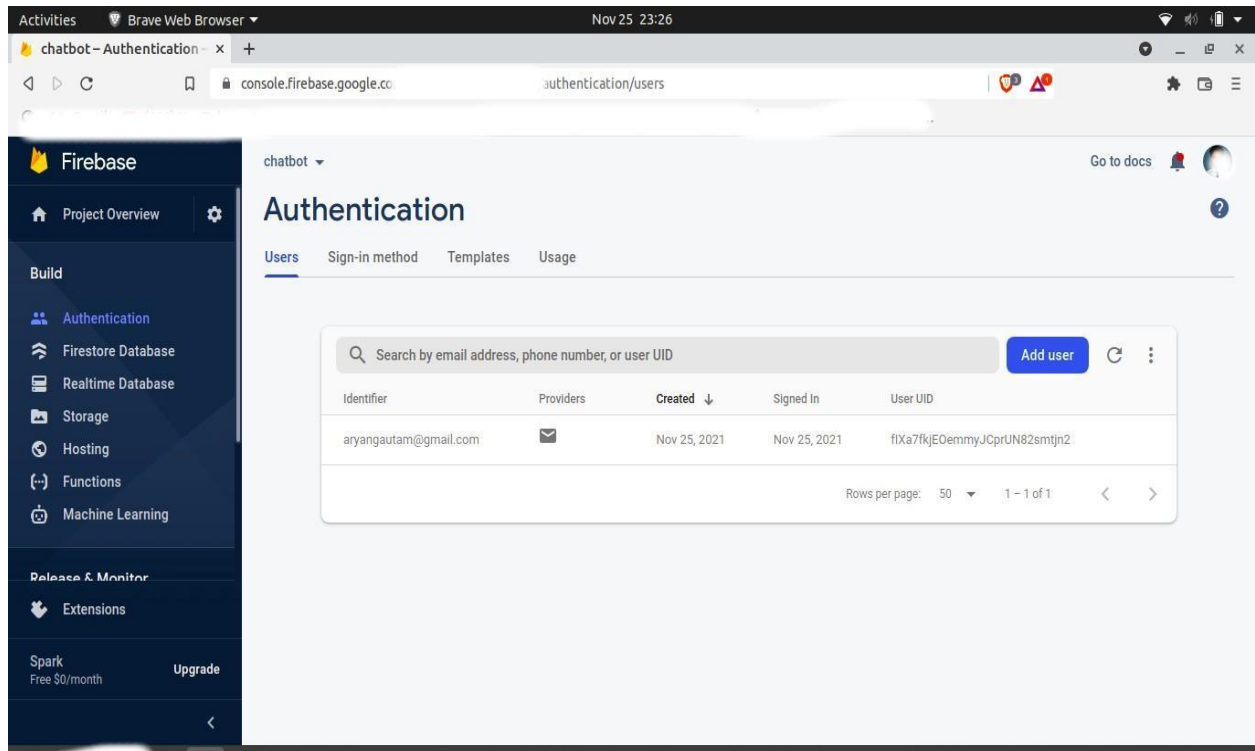


fig-2.1 screenshot of authentication from database

chatbot input and response

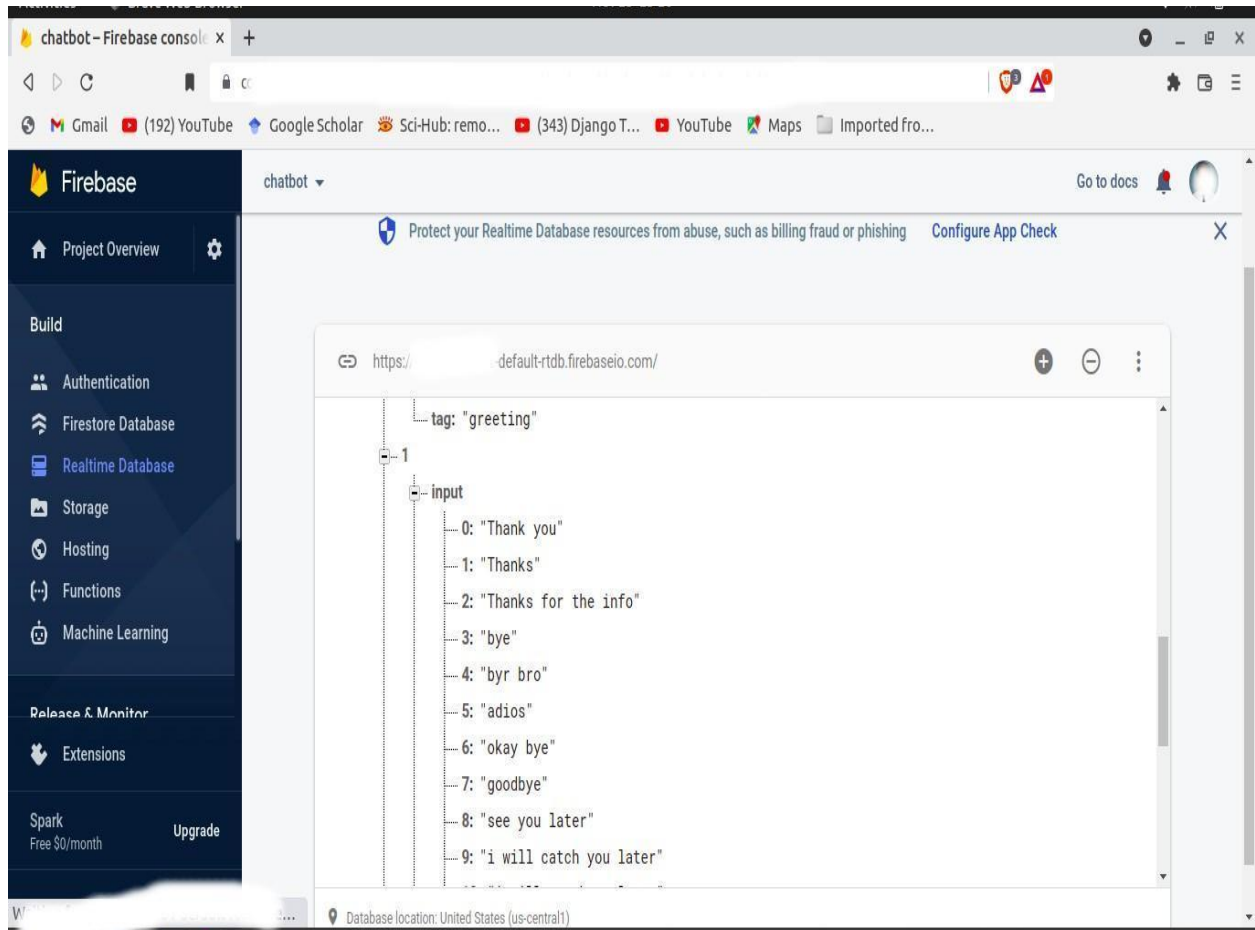


fig-2.2 screenshot of realtime database

4.3 Module Description

There are 3 main modules in chatbot assistant.

- Register
- Login
- Chatbot

Register

This window asks the user to enter their name, email, enrollment no. and password and confirm password. After data is stored. It shows the notification “user is created”.



The image shows a registration form with a dark blue background and a light blue grid pattern. The form is titled "Register Here" in a large, bold, blue font at the top. Below the title, there are six input fields arranged in two columns. The left column contains labels for "First Name", "Last Name", "Email Address", "Enrollment No.", "Password", and "Confirm Password". The right column contains corresponding white input boxes. A small blue button with a white question mark is located between the "Password" and "Confirm Password" labels. Below the input fields, there is a "Submit" button. At the bottom of the form, there is a link that says "Already have an account?? go to Login Page".

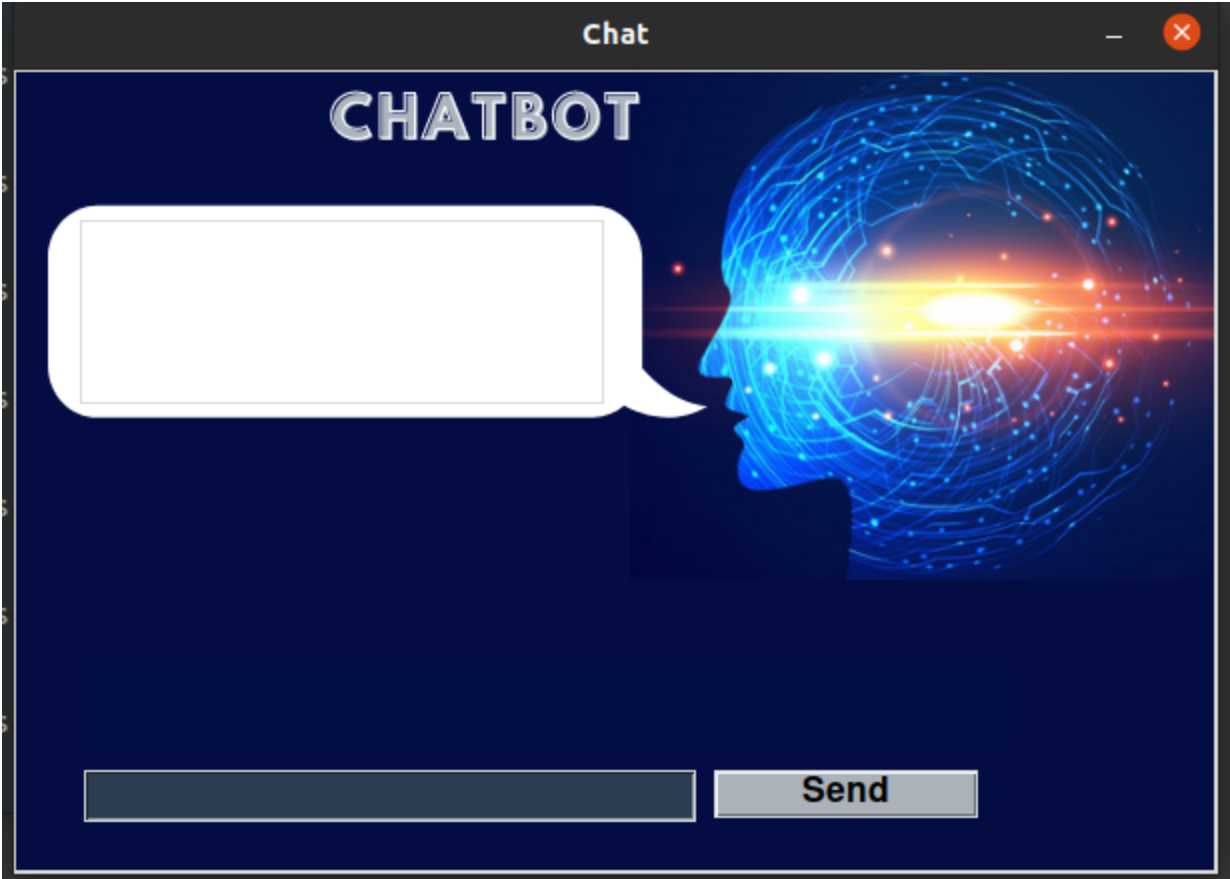
Login

This window asks the user to enter email and password and authenticate from firebase’s realtime database. After authentication it redirects to the chatbot window.



Chatot

It solves the query of the users from the trained model. It also records the user's email id to send a marksheet if they need. If the user types bye the window will be closed.



4.4 Results and Outputs

User enters name, email, password, confirm password and enrollment number into the register window. After clicking on submit button data saves into firebase. After successful creation of the user a dialog.

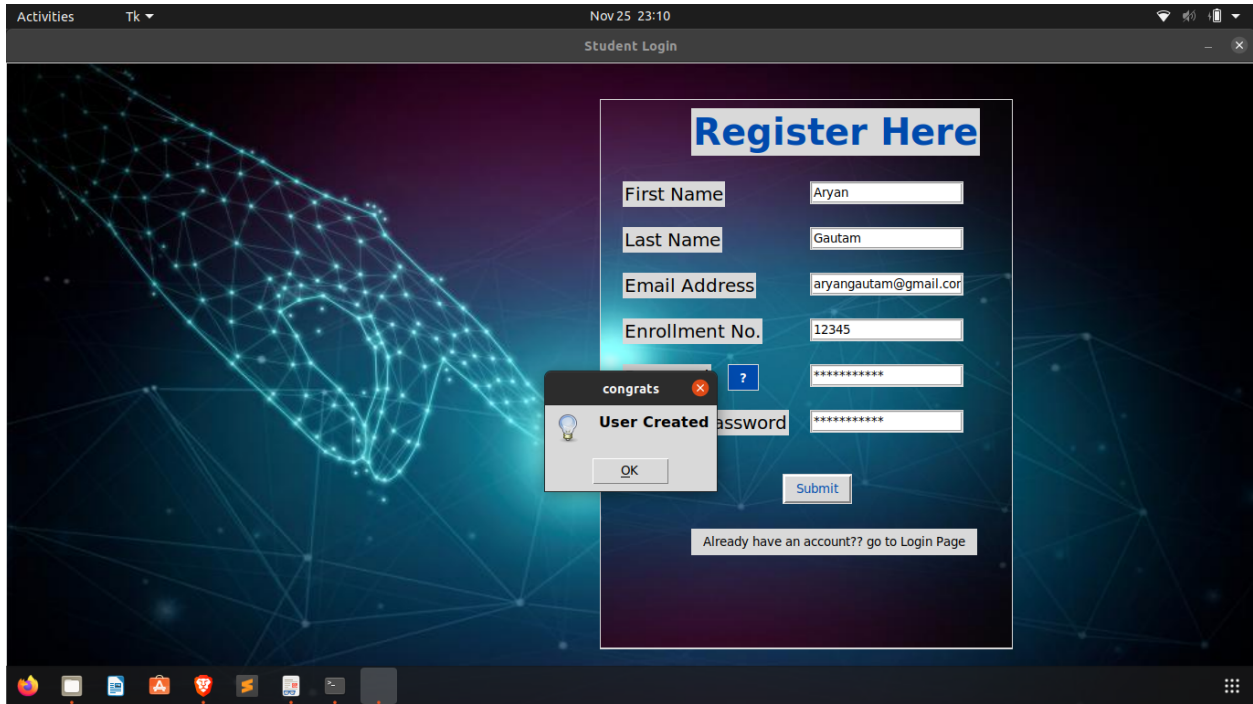


fig-3.1 screenshot of user registration

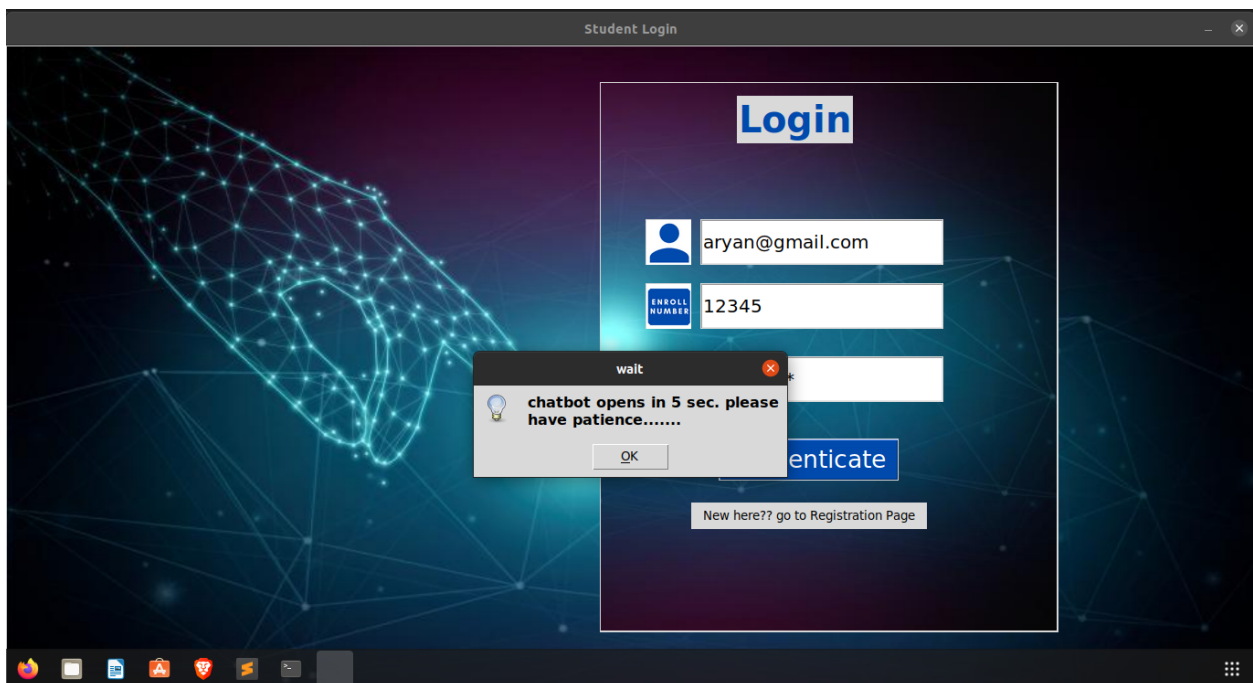


fig-3.2 screenshot of login authentication



fig-3.3 screenshot of chatbot of sample input 1



fig-3.4 screenshot of chatbot of sample input 2



fig-3.5 screenshot of chatbot of sample input 3

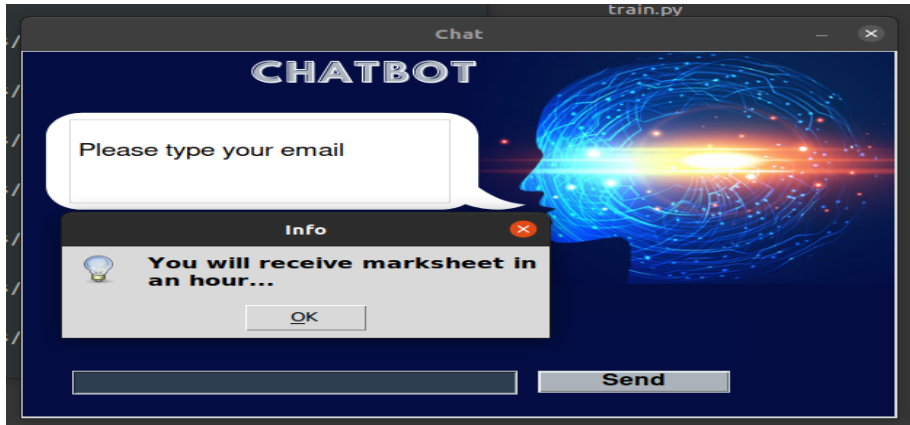


fig-3.6 screenshot of chatbot of sample input4

CONCLUSION AND FUTURE SCOPE

5.1 CONCLUSION

After implementing chatbot assistant we came to the conclusion that making an AI chatbot is useful in many ways like if we make a rule based chatbot it takes some fixed set of input and reply accordingly and we may not have the options to ask questions we can only select input from the list. But in the case of AI based chatbots we can train the input and responses according to their tag and when the user asks a question it checks from the trained model and if the sentence has a high frequency of matching from input it replies accordingly. There are also possibilities that some questions may not have the answers but these inputs can be recorded and data will be updated from time to time and every time the application will run the model will first train the data from the updated json file and then run.

5.2 FUTURE SCOPE

Chat assistants use the technique of tensorflow to train the model. These techniques can be used to solve different known and unknown problems. Chat assistants use the technique of tensorflow to train the model. These techniques can be used to solve different known and unknown problems. It is a rising trend and it will help to increase the effectiveness of business by providing the best experience at low cost.

TAKE AWAY FROM THE PROJECT

6.1 REFERENCE

- www.google.com
- www.tutorialspoint.com
- www.stackoverflow.com
- www.geeksforgeeks.org
- www.tensorflow.org
- <https://machinelearningknowledge.ai/category/natural-language-processing/>