# CHATBOT FOR HOSPITAL MANAGEMENT

***A Major Project Report Submitted in the partial fulfillment of the requirements for the award of the degree of***

## BACHELOR OF TECHNOLOGY
## IN
## INFORMATION TECHNOLOGY

Submitted by


| | |
|---|---|
| Ms. A. LAHARI | (18071A1203) |
| Ms. B. APARNA | (18071A1208) |
| Ms. N. CHANDANA | (18071A1243) |
| Mr. T. RAJESHWAR RAO | (18071A1253) |




**DEPARTMENT OF INFORMATION TECHNOLOGY**

**VALLURUPALLI NAGESWARA RAO VIGNANA JYOTHI**

**INSTITUTE OF ENGINEERING AND TECHNOLOGY**

An Autonomous Institute, NAAC Accredited with 'A++' Grade NBA Accredited for CE, EEE, ME, ECE, CSE, EIE, IT B. Tech Courses Approved by AICTE, New Delhi, Affiliated to JNTUH Recognized as "College with Potential for Excellence" by UGC ISO 9001:2015 Certified, QS I GUAGE Diamond Rated

Vignana Jyothi Nagar, Pragathi Nagar, Nizampet (S.O), Hyderabad – 500 090, TS, India


JUNE 2022

# CHATBOT FOR HOSPITAL MANAGEMENT

***A Major Project Report Submitted in the partial fulfillment of the requirements for the award of the degree of***

**BACHELOR OF TECHNOLOGY**
**IN**
**INFORMATION TECHNOLOGY**

Submitted by


| | |
|---|---|
| Ms. A. LAHARI | (18071A1203) |
| Ms. B. APARNA | (18071A1208) |
| Ms. N. CHANDANA | (18071A1243) |
| Mr. T. RAJESHWAR RAO | (18071A1253) |


**Under the esteemed guidance of**


Dr. D. Srinivasa Rao
Associate Professor,
Dept. of Information Technology,
VNRVJIET




**DEPARTMENT OF INFORMATION TECHNOLOGY**

**VALLURUPALLI NAGESWARA RAO VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY**

An Autonomous Institute, NAAC Accredited with 'A++' Grade NBA Accredited for CE, EEE, ME, ECE, CSE, EIE, IT B. Tech Courses Approved by AICTE, New Delhi, Affiliated to JNTUH Recognized as "College with Potential for Excellence" by UGC ISO 9001:2015 Certified, QS I GUAGE Diamond Rated

Vignana Jyothi Nagar, Pragathi Nagar, Nizampet (S.O), Hyderabad – 500 090, TS, India


June 2022

# VALLURUPALLI NAGESWARA RAO VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

An Autonomous Institute, NAAC Accredited with 'A++' Grade NBA Accredited for CE, EEE, ME, ECE, CSE, EIE, IT B. Tech Courses Approved by AICTE, New Delhi, Affiliated to JNTUH Recognized as "College with Potential for Excellence" by UGC ISO 9001:2015 Certified, QS I GUAGE Diamond Rated

Vignana Jyothi Nagar, Pragathi Nagar, Nizampet (S.O), Hyderabad – 500 090, TS, India

**DEPARTMENT OF INFORMATION TECHNOLOGY**

## CERTIFICATE

This is to certify that the project work entitled **"CHATBOT FOR HOSPITAL MANAGEMENT"** is being submitted by **Miss. A. LAHARI (18071A1203), Miss. B. APARNA (18071A1208), Miss. N. CHANDANA(18071A1243) ,Mr. T. RAJESHWAR RAO (18071A1253)** in partial fulfillment for the award of Degree of Bachelor of Technology in Information Technology to the Jawaharlal NehruTechnological University, Hyderabad during the academic year **2021-22** is a record of bonafide work carried out by him/her under our guidance and supervision.

The results embodied in this report have not been submitted by the students to any other University or Institution for the award of any degree or diploma.

**Under the Guidance of:**

**Dr. D. Srinivasa Rao,**
**Associate Professor,**
**Department of IT,**
**VNRVJIET**

**Head of the Department**

**Dr. D. Srinivasa Rao,**
**Associate Professor,**
**Department of IT,**
**VNRVJIET**

# VALLURUPALLI NAGESWARA RAO VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

An Autonomous Institute, NAAC Accredited with 'A++' Grade NBA Accredited for CE, EEE, ME, ECE, CSE, EIE, IT B. Tech Courses Approved by AICTE, New Delhi, Affiliated to JNTUH Recognized as "College with Potential for Excellence" by UGC ISO 9001:2015 Certified, QS I GUAGE Diamond Rated

Vignana Jyothi Nagar, Pragathi Nagar, Nizampet (S.O), Hyderabad – 500 090, TS, India

**DEPARTMENT OF INFORMATION TECHNOLOGY**

## DECLARATION

I hereby declare that the project entitled **"CHATBOT FOR HOSPITAL MANAGEMENT"** submitted to VNR Vignana JyothiInstitute of Engineering and Technology in partial fulfillment of the requirement for the award of Bachelor of Technology in Information Technology is a bonafide report of the work carried out by us under the guidance and supervision of Dr.D.Srinivasa Rao, Associate Professor, Department of Information Technology, VNRVJIET.

To the best of my knowledge, this has not been submitted in any form to any university or institution for the Award of any degree or diploma.

Signature of the Student:

| A.LAHARI | B.APARNA | N.CHANDANA | T.RAJESHWAR |
|---|---|---|---|
| (18071A1203) | (18071A1208) | (18071A1243) | (18071A1253) |

Place:

Date:

# ACKNOWLEDGMENT

Behind every achievement lies an unfathomable sea of gratitude to those who activated it, without which it would never have come into existence. To them, we lay the words of gratitude imprinted within us.

We are indebted to our venerable principal **Dr. C. D. Naidu** for this unflinching devotion, which led us to complete this project. The support, encouragement given by him, and his motivation led us to complete this project.

We express our thanks to our internal guide **Dr. D. Srinivasa Rao** for having provided us with a lot of facilities to undertake the project work and guide us to complete the project.

We express our sincere thanks to our faculty of the Department of **Information Technology** and the remaining members of our college **VNR VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY** who extended their valuable support in helping us to complete the project in time.

1. Miss.A. Lahari (18071A1203) _____
2. Miss.B. Aparna (18071A1208) _____
3. Miss.N. Chandana (18071A1243) _____
4. Mr.T. Rajeshwar rao (18071A1253) _____

# CHATBOT FOR HOSPITAL MANAGEMENT

***A Major Project Report Submitted in the partial fulfillment of the requirements for the award of the degree of***

## BACHELOR OF TECHNOLOGY
## IN
## INFORMATION TECHNOLOGY

Submitted by

Ms. A. LAHARI (18071A1203)

**Under the esteemed guidance of**

Dr. D. Srinivasa Rao,
Associate Professor,
Information Technology,
VNRVJIET


## DEPARTMENT OF INFORMATION TECHNOLOGY

**VALLURUPALLI NAGESWARA RAO VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY**

An Autonomous Institute, NAAC Accredited with 'A++' Grade NBA Accredited for CE, EEE, ME, ECE, CSE, EIE, IT B. Tech Courses Approved by AICTE, New Delhi, Affiliated to JNTUH Recognized as "College with Potential for Excellence" by UGC ISO 9001:2015 Certified, QS I GUAGE Diamond Rated

Vignana Jyothi Nagar, Pragathi Nagar, Nizampet (S.O), Hyderabad – 500 090, TS, India

June 2022

# CHATBOT FOR HOSPITAL MANAGEMENT

***A Major Project Report Submitted in the partial fulfillment of the requirements for the award of the degree of***

**BACHELOR OF TECHNOLOGY**
**IN**
**INFORMATION TECHNOLOGY**

Submitted by

Ms. B. APARNA (18071A1208)

**Under the esteemed guidance of**

Dr. D. Srinivasa Rao,
Associate Professor,
Information Technology,
VNRVJIET


**DEPARTMENT OF INFORMATION TECHNOLOGY**

**VALLURUPALLI NAGESWARA RAO VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY**

An Autonomous Institute, NAAC Accredited with 'A++' Grade NBA Accredited for CE, EEE, ME, ECE, CSE, EIE, IT B. Tech Courses Approved by AICTE, New Delhi, Affiliated to JNTUH Recognized as "College with Potential for Excellence" by UGC ISO 9001:2015 Certified, QS I GUAGE Diamond Rated

Vignana Jyothi Nagar, Pragathi Nagar, Nizampet (S.O), Hyderabad – 500 090, TS, India

June 2022

# CHATBOT FOR HOSPITAL MANAGEMENT

***A Major Project Report Submitted in the partial fulfillment of the requirements for the award of the degree of***

**BACHELOR OF TECHNOLOGY**
**IN**
**INFORMATION TECHNOLOGY**

Submitted by

Ms. N. CHANDANA (18071A1243)

**Under the esteemed guidance of**

Dr. D. Srinivasa Rao,
Associate Professor,
Information Technology,
VNRVJIET


**DEPARTMENT OF INFORMATION TECHNOLOGY**

**VALLURUPALLI NAGESWARA RAO VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY**

An Autonomous Institute, NAAC Accredited with 'A++' Grade NBA Accredited for CE, EEE, ME, ECE, CSE, EIE, IT B. Tech Courses Approved by AICTE, New Delhi, Affiliated to JNTUH Recognized as "College with Potential for Excellence" by UGC ISO 9001:2015 Certified, QS I GUAGE Diamond Rated

Vignana Jyothi Nagar, Pragathi Nagar, Nizampet (S.O), Hyderabad – 500 090, TS, India

June 2022

# CHATBOT FOR HOSPITAL MANAGEMENT

***A Major Project Report Submitted in the partial fulfillment of the requirements for the award of the degree of***

## BACHELOR OF TECHNOLOGY
## IN
## INFORMATION TECHNOLOGY

Submitted by

Mr. T. RAJESHWAR RAO (18071A1253)

**Under the esteemed guidance of**

Dr. D. Srinivasa Rao,
Associate Professor,
Information Technology,
VNRVJIET


## DEPARTMENT OF INFORMATION TECHNOLOGY

**VALLURUPALLI NAGESWARA RAO VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY**

An Autonomous Institute, NAAC Accredited with 'A++' Grade NBA Accredited for CE, EEE, ME, ECE, CSE, EIE, IT B. Tech Courses Approved by AICTE, New Delhi, Affiliated to JNTUH Recognized as "College with Potential for Excellence" by UGC ISO 9001:2015 Certified, QS I GUAGE Diamond Rated

Vignana Jyothi Nagar, Pragathi Nagar, Nizampet (S.O), Hyderabad – 500 090, TS, India

June 2022

# ABSTRACT

The healthcare sector represents one of the most significant segments of the economy. This sector offers medical services and goods to everyone. A reliable healthcare system ensures a strong economy by increasing life expectancy, contributing to national growth, and reducing the burden on families.

The purpose of this project is to implement a proper healthcare management system integrating all the basic functionalities powered by an Artificial intelligence (AI) chatbot that is capable of having a very organic conversation with the user and solving their queries using a knowledge base. The knowledge base has real-time data collected in a JSON format which is pre-processed to make it ready for further processing. The bag of words model is used for further pre-processing. Further, the proposed methodology uses Deep Neural Networks to implement the chatbot which has speech recognition capabilities. The information is received and delivered in both speech and text formats.

The chatbot can provide navigation links according to the requests of a user. Furthermore, it is capable of predicting the problem by performing symptom diagnosis and recommending a doctor to be consulted and any immediate measures to be taken. In addition, it provides information regarding diagnostics beforehand.

Chatbots are a sort of intelligent conversational system that works on natural language input, which could be text, speech, or a combination of the two. They respond via conversational output and are often used for accomplishing tasks.

# LIST OF FIGURES

# INDEX

# CHAPTER 1 - INTRODUCTION

An AI chatbot is a computer program that simulates human communication. It is a piece of software that interacts with a human through written language. It is often embedded in web pages or other digital applications to answer customer inquiries without the need for human agents, thus providing affordable effortless customer service. Chatbots based on Machine Learning make an AI chatbot that is very capable of having an organic conversation with the user and answering their queries. Chatbots make use of the data given to them and using different training algorithms they can answer the queries in the best way possible.

In our proposed system we create a conversational chatbot that is integrated into a hospital website. It is trained using Machine Learning algorithms and acts as a very efficient interface between the user and the application. There is no predefined format for the users to ask their queries in, the chatbot manages to answer the query in the best possible way. Users have the flexibility to raise a query both in text and speech format.

With this chatbot, users have access to hospital information, doctor availability, diagnostics, and other related data. They are navigated to different pages according to their requests which makes it easier and faster for them to explore. They can book appointments, and identify the problem by specifying symptoms to try to know about it beforehand, doing this they can take any required precautionary measures and book an appointment with the doctor as soon as possible.

Figure1.1: Chatbot model

## 1.1 Purpose of the Project

Healthcare being one of the most important sectors in our economy is the motivation for the project. The vision is for this proposed system to function efficiently and be accessible to everyone. Integrating all basic features in one place in an application and powering it with an AI chat bot further adds new functionalities like easy navigation, access to data on doctors, diagnostics information, symptom analysis, precautionary or instant medication suggestions and appointment booking, all these in a single application. Further, considering people who can not write fluently,those with special needs and those in emergency situations, both voice and text input formats are accepted by the chatbot.

## 1.2 Existing methodologies and disadvantages

In all the existing systems, the scope is divided and they provide very few features at a time. Few chatbots only provide appointment booking functionality only and also may not include voice inputs from the users, this makes it difficult for people with special needs or those who are in urgency. Some bots provide disease diagnosis but can’t provide medication and navigation through a complex hospital website. There are smaller number of chatbots integrated to hospital website that provides all the necessary contents and features.

Health care being one of the important sectors in our economy this kind of management is not feasible,the system needs to be fast,efficient and make use of upcmoing technologies in order to improvise its design and improve its performance.

## 1.3 Proposed System

The proposed system focuses on integrating all basic features in one place in an application and powering it with an AI chat bot further adds new functionalities like easy navigation, access to data on doctors, diagnostics information, symptom analysis, precautionary or instant medication suggestions and appointment booking, all these in a single application. Further, considering people who cannot write fluently, those with special needs and those in emergency situations, both voice and text input formats are accepted by the chatbot.

We are building the website using Flask, which contains Login and registration page, dashboard of website, appointment booking and viewing pages and also the animated chatbot button at the end of every page of the website.

Speech enabled chatbots provide higher level of interactivity and usability. User can either give their input using text or speech and similarly chatbot is able to give its response by either text or voice. In our project, this process of conversion between text and speech is done by using speech_recognition and pyttsx3 python modules.

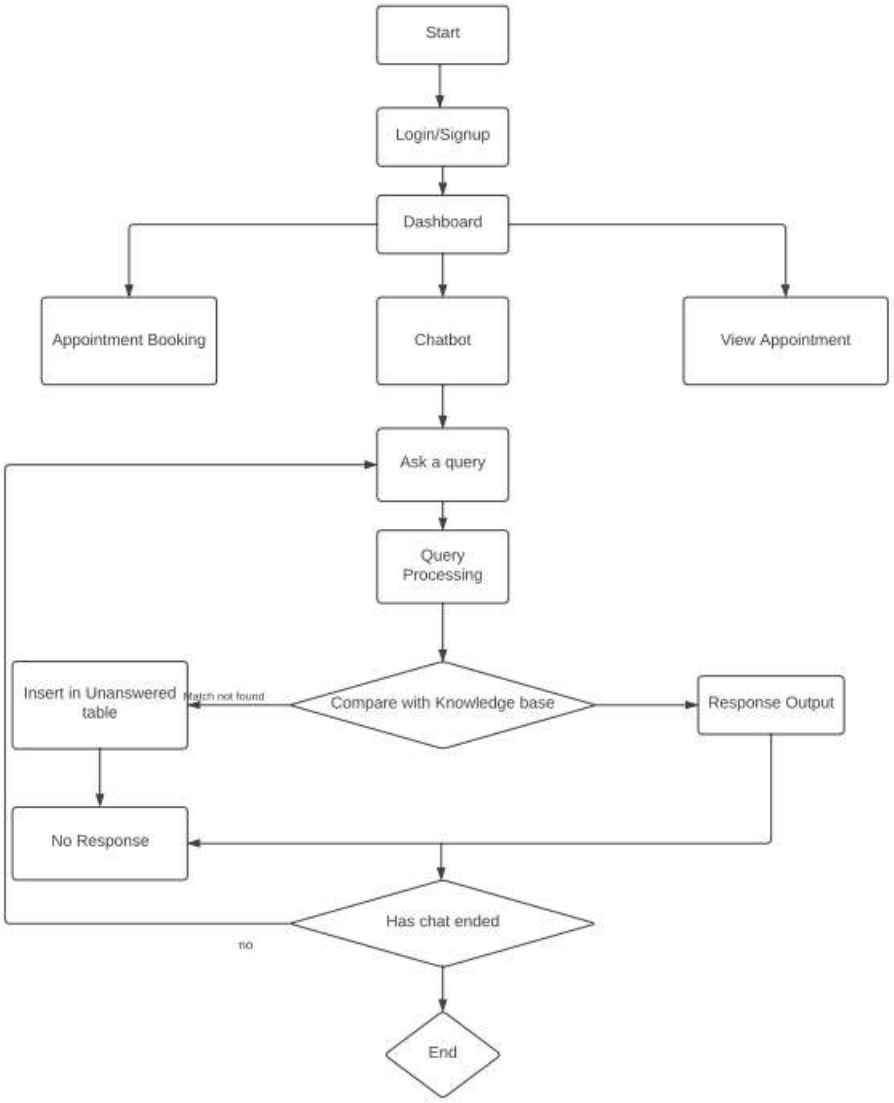


Figure1.3: Flow chart of chatbot

## 1.4 Objective

The goal of the Hospital Management Chatbot project is to employ algorithms to read user questions and answer them. The Hospital Management Chatbot project is based on algorithms that assess and interpret user questions and messages. This involves using an online application to provide answers to the user's queries. Users can ask the chabots any hospital or medical related questions. The computer examines the user's query and responds accordingly. The system answers the inquiry as if it were being asked to a person. With the use of algorithms, the computer reacts to the user’s questions. An option to raise a quesry in both voice and text formats will be provided. Users can book appointments,navigate the website as required,perform symptom diagnosis and get information related to the hospital,doctor and diagnostics available at the hospital.This application will be a one place for all queries and the chatbots acts as the medium.

## 1.5 Thesis Organization

Chapter 1: This chapter presents the basic information and introduction and the necessary technical knowledge to implement.

Chapter 2: This chapter presents the literature survey and research needed to implement the project.

Chapter 3: This chapter presents the challenges and issues that occurred during the implementation of the project.

Chapter 4: This chapter presents the software requirement analysis

Chapter 5: This chapter presents the software design with UML diagrams

Chapter 6: This chapter presents the methodology and approach

Chapter 7: This chapter presents the implementation of the methodology.

Chapter 8: This chapter presents the experimental results of the model.

Chapter 9: This chapter presents the conclusion.

Chapter 10: Future Scope of the Project

Chapter 11: References collected for the Project

Chapter 12: Frontend and Backend source code of the Project

# CHAPTER 2 - LITERATURE SURVEY

In the paper by Mamta Mittal [1], Gradient descent method, NLP (Natural Language Processing), and feed-forward neural network (FNN) are some of the algorithms used to create the chatbot.Gradient Descent (GD) is a cost-minimization technique that examines the coefficients of a function (f).It is a key optimisation approach for determining the minimal cost function.The model may be conveniently stored in memory with little noise using the GD technique. Computational linguistics—rule-based human language modeling—is combined with statistical, deep learning models and machine learning in NLP.This chatbot answers questions about hospital information, such as specialist availability, OPD hours, room registration, bed capacity, doctor availability and emergency information, among other things.The suggested chatbot acts as if it were a genuine hospital receptionist, assisting users. It offers consumers complete medical support 24 hours a day, seven days a week.

In the paper by Rohit Binu Mathew[2], KNN (K-nearest neighbor algorithm) and NLP (Natural Language Processing) algorithms are used to create the chatbot. The K-Nearest neighbor method is based on the Supervised Learning technique and is one of the most basic Machine Learning algorithms. The.K-NN method assumes that the new case/data and existing cases are comparable and places the new case in the category which is most similar to existing categories. The K-NN method maintains all existing data and uses similarity to classify new data points. This implies that when fresh data is generated, it may be quickly categorized into a well-suited category using the K-NN method. The created chatbot application is an android application in which the user may tell the chatbot about their symptoms, and the chatbot will then tell them what health measures they should take.

In the paper by Harsh Mendapara [3],the backend of the chatbot is written in Python, while the user interface is created using HTML, CSS, and JavaScript. Chatterbot, a natural processing library, is used to communicate between the user and the system. Text analysis is used to apply natural language processing. On the localhost server, the healthcare assistant's frontend interface is presented, and it is ready to address patient symptoms based on a certain ailment. The health assistant will collect certain personal information from the user at first, which will be saved in the

The database.The chatbot will next ask the user a question in which the user is expected to address health-related issues. If the patient has a high temperature, high blood pressure or low blood pressure then the chatbot will prescribe the necessary medicine

In the paper by Siddhi Pardeshi[4],Long Short-Term Memory (LSTM),Natural Language Processing (NLP) , Hybrid Emotion Inference Model (HEIM), Pattern Matching Algorithm and Naive Bayes Algorithm are some of the chatbot design techniques covered.Natural language processing allows machines to take in input, break it down, retrieve its meaning, determine suitable action, and respond to users in natural language.Long Short Term Memory (LSTM) is a Artificial Recurrent Neural Network (RNN).LSTMs are useful not just for processing single data inputs such as photos, but also for processing full sequences of data such as voice or video.The LSTM algorithm's primary tasks are handwriting identification and speech recognition.In chatbots, pattern matching became one of the most commonly utilized algorithms. The Pattern Matching Algorithm is a database that comprises questions and answers.Patterns are used to name questions, whereas templates are used to identify responses/answers.Artificial Intelligence Mark-up Language (AIML) tags make up the response to this specific query.Patterns(query) and templates(answers) are kept in a tree format. Questions are on the branches, and responses are at the nodes, thus anytime a user asks a question, the query is first searched for an answer term by term, and then the specific answer is fetched from the node.

In the paper by Lekha Athota[5], N-gram, which is a series of N words which is used to construct the chatbot application.So for example, “Final demo” is a 2-gram (a bigram), “This is a final demo” is a 4-gram, and “Good to go” is a 3-gram (trigram).The TF-IDF (term frequency-inverse document frequency) statistically examines the relevance of a word to a document in a collection of documents.This is accomplished by multiplying two metrics: the number of times a word occurs in a document and the word's inverse document frequency over a collection of documents.It's used to get the keyword out of the user query. To get the best response for the inquiry, each term is weighted down.The Web-interface is designed for users to enter their query.The programme is enhanced with security and effectiveness modifications that ensure user protection and integrity when getting answers to queries.This chatbot assists users with basic health information.When a person initially visits the website, they must register before asking the questions to chatbot.If the answer is not in the database,

the system employs an expert system to respond to the queries

## 2.1 Related Work

**Healthcare Chatbot System**

The artificial intelligence field had not previously been created. Users' problems are solved in less time since the development of chatbot systems. In the realm of healthcare, the use of automated chatbots in web applications is on the rise all around the world. Patients have a variety of disorders and need to go to the hospital for treatment. Sometimes doctors may not be available in the hospital and nursing also takes lot of time. Medical chatbots are developed to overcome this issue. The chatbots are well trained and tested on dataset. The AI-powered chatbots are quick, dependable, and precise. User provides their issue and chatbot gives response according to their query. Nowadays, in all clinics and hospitals portal chatbots are performing multitasking work. A lot of time of patient is saved and tasks are completed in minimum effort.

**Chatbot for Disease Prediction and Treatment Recommendation**

The user converses with the chatbot programme in the same manner that he or she converses with other individuals. Android application that requires the user to first login to the system after registering. The chatbot discovers the user's symptoms through this conversation. The user submits messages, and the chatbot responds with the relevant message. To ensure that this goes successfully, the chatbot will be taught with certain predefined questions and replies that the user can ask. Text processing will take place when the user submits messages. Natural language processing is used to process text (NLP). NLP allows humans to seamlessly converse with machines. NLP attempts to comprehend human natural language and classifies, analyses, and responds to it as needed. When a query is received, the chatbot tries to find a match in the dataset it has previously been trained on, which will be one of the k closest neighbours. This is accomplished with the assistance of KNN. Python offers a large library that caters to the demands of NLP. The Natural Language Tool Kit is a collection of such libraries that offers the NLP functionality.

**AI Based Healthcare Chatbot System**

The backend of the healthcare chatbot is built with Python, while the user interface is built with HTML, CSS, and JavaScript. Chatterbot, a natural processing library, is used to communicate between the user and the system.The programme runs on the localhost server, which gives relevant information based on the user's inquiries.The train.py programme is run during the training phase, and a new database is built.All of the database files in the application model's first stage are in yml format.The healthcare assistant's frontend interface is presented on the localhost server and ready to solve the patient's symptoms based on a certain ailment.Initially, the health assistant will collect certain personal information from the user, which will be saved in a database.Some of the symptoms where user inquiries are put are headache, cough, cold, and so on.For a doctor's appointment, a different data file is created. The training will assist the bot in improving answer accuracy.After executing the train.py program , all data will be loaded into the MYSQL database and a new user will be created. After building the new database, it will list all of the data files and begin training.That URL contains the chatbot web application, which may be accessed via any internet browser, including Google Chrome, Firefox, and others.

# CHAPTER 3 - ISSUES AND CHALLENGES

Even though we used a static method for this project, we encountered several problems, such as defining a project flow and establishing a model. For this project, we chose Python; however, we encountered version troubles at first, and obtaining all of the essential libraries took a lot of time. The JSON format doesn't use schema, it is a textual representation of structured data. This providestotal flexibility to represent the data the way you want while you could also create ill-shaped datavery easily. The data preparation needs special attention on structural patterns along with thecontent.

There are various methods used for dimensionality reduction of textual data. After performing pre-processing techniques for the JSON data which is prepared from scratch, there existed some unsatisfactory results for stem words. The techniques slightly identified few other non-stem words as stem words as well. Due to this, the features are becoming limited. To address this issue we useda feature engineering algorithm (bag of words).

NLP has an advanced outlook on everything. NLP takes the synonyms and extraction of the entitieswith greater ease but it fails in identifying the local language i.e. the words that have argot and require special attention. Most of the general audience are habituated to give shortcuts while texting.They use "u" in the place of "you". It consists of efficient techniques and new techniques evolve asthe necessity increases.

The python library Tkinter is very great at applying the basic library elements but fails in few advanced libraries. The speech recognition library used isn't supported by Tkinter wholly. Therefore, it is required in the installation of other supported formats as intermediate to support theservice. The response is taken and given as a text widget which in turn converts data to a string. This makes it a hard task to access the URLs specifies in the data as a response. The URLs which are interlinked need to be declared globally and accessed differently for different responses which makes it complex.

# CHAPTER 4: SOFTWARE REQUIREMENT ANALYSIS

Requirements analysis is a crucial step for determining the success of a system or software project. There are two types of requirements: those that are mandatory and those that are optional.

- Functional requirements
- External Interface requirements and
- Non-functional requirements

## 4.1 FUNCTIONAL REQUIREMENTS

### 4.1.1 Definition

These are the necessities that the end client communicates as fundamental elements that the framework ought to give. As a feature of the agreement, these functionalities should be incorporated into the framework. These are communicated or depicted as contribution to be conveyed to the framework, activity to be directed, and anticipated yield. They are the client's communicated prerequisites that, in contrast to non-utilitarian models, should be visible promptly in the finished item.

### 4.1.2 Requirements for this project

- **User Registration**

- The user needs to register if he/she is new to the application

- It is the basic action that the user needs to perform to get complete access of the website

- **User Login**

- The user needs to login if they are already existing user to the application

- If user fails to login, they can’t access the application.

- By logging into the website, all the features are made available for the user to use

- **Using the Chatbot**

- This is the most important feature of the whole website where the user can solve their health-related queries
- The user needs to enter the question or query which will be resolved by the Chatbot

## 4.2 EXTERNAL INTERFACE REQUIREMENTS

### 4.2.1 Definition

Functional requirements include external interface needs. For embedded systems, they're crucial. They also describe how your product will interact with other elements.
There are various types of interfaces for which you may have a need, including:

- User
- Hardware
- Software
- Communications

### 4.2.2 Requirements for this project

The below are the software required for the python app in this project

### Python:

Python is an interpretable significant level programming language. Python offers succinct and discernible code. While complex calculations and adaptable work processes stand behind AI and AI, Python's straightforwardness permits designers to compose dependable frameworks. Designers get to invest all their energy into taking care of a ML issue as opposed to zeroing in on the specialized subtleties of the language. It utilizes an article situated way to deal with furnish clients with a reasonable perspective on the tasks' intelligent and practical parts.

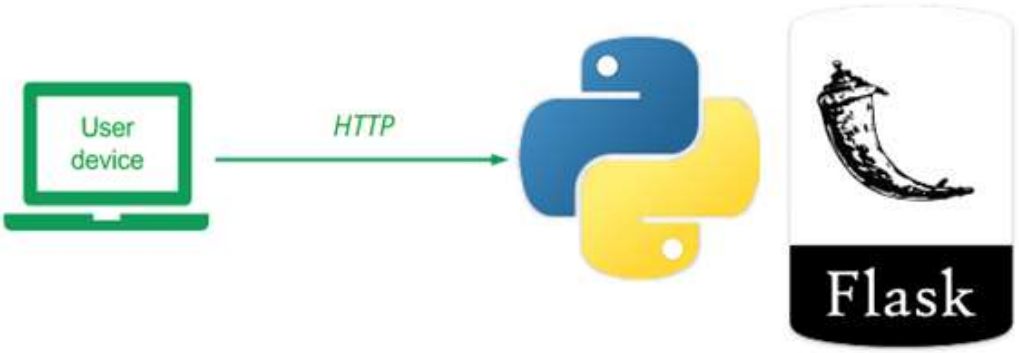


Fig 4.2.2 Requirements for nthe project

## Google Colab:

Colaboratory, or "Colab" for short, is an item from Google Research. Colab permits anyone to compose and execute inconsistent python code through the program, and is particularly appropriate to AI, information investigation and instruction. All the more in fact, Colab is a facilitated Jupyter note pad administration that requires no arrangement to use, while giving access for nothing to figuring assets including GPUs.Jupyter is the open source project on which Colab is based. Colab permits you to utilize and impart Jupyter journals to others without downloading, introduce, or run anything.

Google Colab gives lots of invigorating highlights that any advanced IDE offers, and substantially more. Probably the most interesting elements are recorded beneath.

- Intuitive instructional exercises to learn AI and brain organizations.
- Compose and execute Python 3 code without having a neighborhood arrangement.
- Execute terminal orders from the Notebook.
- Import datasets from outer sources like Kaggle.
- Save your Notebooks to Google Drive.
- Import Notebooks from Google Drive.
- Free cloud administration, GPUs and TPUs.
- Coordinate with PyTorch, Tensor Flow, Open CV.

## NLTK:

The Natural Language Toolkit (NLTK) is a stage utilized for building Python programs that work with human language information for applying in measurable normal language handling (NLP).

It contains text handling libraries for tokenization, parsing, arrangement, stemming, labeling and semantic thinking. It likewise incorporates graphical exhibitions and test informational collections as well as joined by a cook book and a book which makes sense of the standards behind the hidden language handling undertakings that NLTK supports.NLTK incorporates in excess of 50 corpora and lexical sources, for example, the Penn Treebank Corpus, Open Multilingual Wordnet, Problem Report Corpus, and Lin's Dependency Thesaurus.

## Google Speech Recognition API:

Google has an incredible Speech Recognition API. This API changes over spoken message (receiver) into composed message (Python strings), momentarily Speech to Text. You can just talk in a mouthpiece and Google API will make an interpretation of this into composed text. The API has superb outcomes for English language.

A discourse acknowledgment API offloads the rationale, to such an extent that you can just send a web solicitation to the API, which then returns the message that was perceived. You can do this from Python code straightforwardly, yet your content will require web access in the background.

The sound is recorded utilizing the discourse acknowledgment module, the module will remember for top of the program. Besides we send the record discourse to the Google discourse acknowledgment API which will then, at that point, return the output.r.recognize_google(audio) returns a string.

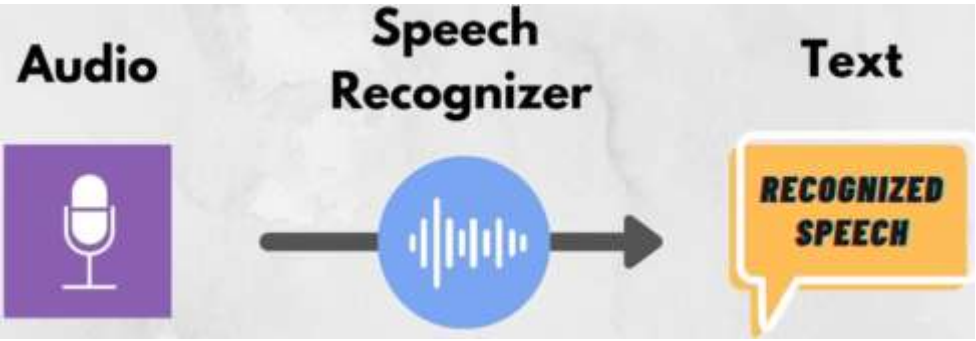


Fig 4.2.2 Audio to Text conversion Representation

And the libraries used for the python app in this project are

I. Numpy
II. Pandas
III. Keras
IV. Tensor flow

## 4.3 NON-FUNCTIONAL REQUIREMENTS

### 4.3.1 Definition

These are the quality requirements that the system must meet in order to fulfil the project contract. The importance of these aspects, as well as the amount to which they are implemented, varies from project to project.

Non-behavioral requirements are another name for them. They basically deal with issues like:

a. Maintainability
b. Performance
c. Security
d. Reusability
e. Flexibility
f. Reliability
g. Portability

h. Scalability

### 4.3.2 Requirements for this project

1.User Interface:

a. Throughout all functionalities and for all clients, the framework should keep a simple to-utilize interface.

b. All ordinarily utilized programs, for example, Internet Explorer, Firefox, Google Chrome, and Safari, ought to be viable with the client's UI.

2.Scalability:

a. The framework should have the option to scale as per the amount of clients.

3.Security:

a. The regulatory framework ought to be shielded from unapproved access.

b. The data set ought to shield from assaults and unapproved access.

c. The point of interaction ought to be shielded from assaults.

d. All passwords ought to be put away as a solid hash of the head secret word.

4.Portability:

a. The framework ought to run on different working frameworks that help the Java language. b. The framework ought to run on an assortment of equipment.

5.Maintainability:

a. The framework ought to be not difficult to keep up with.

b. There ought to be a reasonable partition of HTML and connection point code.

c. There ought to be a reasonable partition between the point of interaction and the business rationale code.

d. There ought to be an unmistakable detachment between the information access protests that map the data set and the business rationale code.

6.Exception taking care of:

a. Exceptions ought to be accounted for actually to the client assuming they happen.

7.Ethics:

a. The framework will not store or cycle any data about its clients.

b. The framework should have the option to scale as per the amount of clients.

8.Security:

a. The managerial framework ought to be shielded from unapproved access.

b. The information base ought to shield from assaults and unapproved access.

c. The connection point ought to be safeguarded from assaults.

d. All passwords ought to be put away as a solid hash of the executive secret phrase.

9.Portability:

a. The framework ought to run on various working frameworks that help the Java language. b. The framework ought to run on an assortment of equipment.

10.Maintainability:

a. The framework ought to be not difficult to keep up with.

b. There ought to be a reasonable partition of HTML and connection point code.

c. There ought to be a reasonable partition between the connection point and the business rationale code.

d. There ought to be an unmistakable partition between the information access protests that map the data set and the business rationale code.

11.Exception dealing with:

a. Exceptions ought to be accounted for actually to the client assuming they happen.

12.Ethics:

a. The framework will not store or cycle any data about its users.The framework should have the option to scale as per the amount of clients.

13.Security:

a. The authoritative framework ought to be safeguarded from unapproved access.

b. The information base ought to shield from assaults and unapproved access.

c. The connection point ought to be safeguarded from assaults.

d. All passwords ought to be put away as a solid hash of the manager secret key.

14.Portability:

a. The framework ought to run on various working frameworks that help the Java language.

b. The framework ought to run on an assortment of equipment.

15.Maintainability:

a. The framework ought to be not difficult to keep up with.

b. There ought to be a reasonable partition of HTML and point of interaction code.

c. There ought to be an unmistakable detachment between the point of interaction and the business rationale code.

d. There ought to be a reasonable detachment between the information access protests that map the data set and the business rationale code.

16.Exception dealing with:

a. Exceptions ought to be accounted for successfully to the client assuming they happen.

17.Ethics:

a. The framework will not store or cycle any data about its client.

# CHAPTER 5: SOFTWARE DESIGN

## 5.1: SYSTEM ARCHITECTURE

The below is the architecture of the system design.

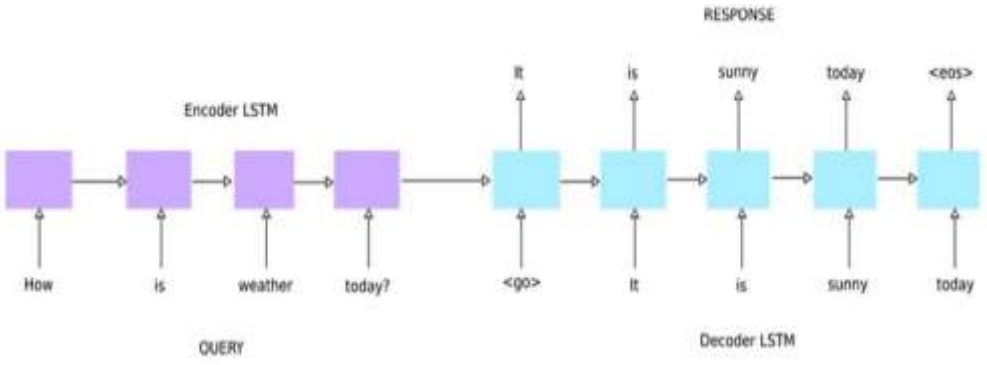


Fig 5.1.1 System Architecture

## 5.2: UML Diagrams

UML diagram is a visually portrayed framework, including its essential participants, jobs, activities, relics, or classes, that are utilized to all the more likely to grasp, update, make due, or report framework data utilizing the UML (Unified Modeling Language). Whether or not it is created previously or later execution, there are various sorts of UML charts, every one of which fills a particular need (as a feature of documentation).

## 5.2.1 Class Diagram

A class diagram is essentially the blueprint of the framework we are building. We can show different things in the class graph by addressing the framework's classes, connections among various articles and the operations among them and furthermore the traits.

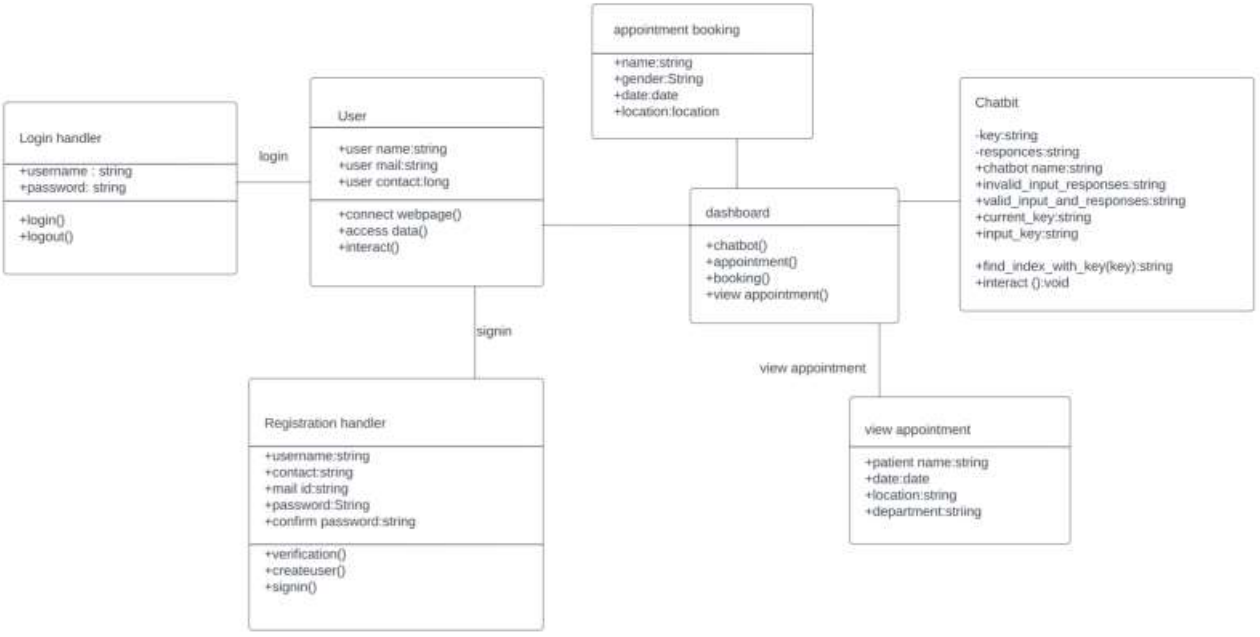


Fig 5.2.1.1 Class Diagram

### 5.2.2 Use Case Diagram

A Use case diagram is a UML conduct or dynamic chart. Actors and use cases are utilized to display the working of a framework being used in case outlines. A bunch of undertakings, administrations and capacities that the framework should do are alluded to as utilize cases. A "system" in this sense alludes to something that is being created or worked, as a site. The "actors" are people or things that carry out specific roles inside the framework. The accompanying addresses the utilization case outline of the proposed framework.

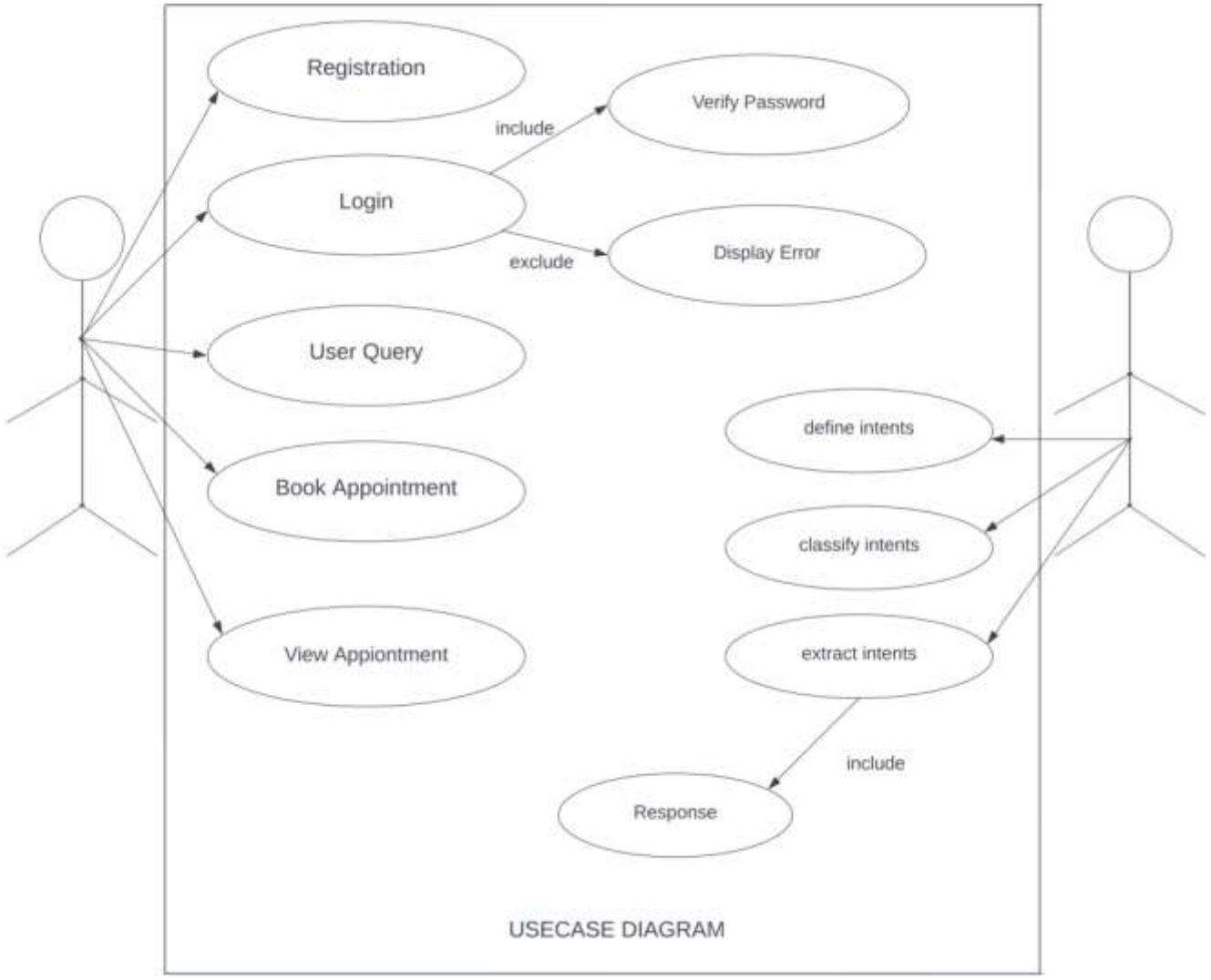


Fig 5.2.2.1 Use Case Diagram of Model

### 5.2.3 Activity Diagram

The activity diagram, which is utilized to depict dynamic parts of the framework, is another significant behavioral diagram in the UML graph. The progression of data starting with one movement and then onto the next is portrayed in a movement outline, which is a more muddled variation of a flow chart. Activity diagrams portray how activities are gathered to deliver help at different degrees of deliberation.

The following represents the activity diagram of the chatbot:

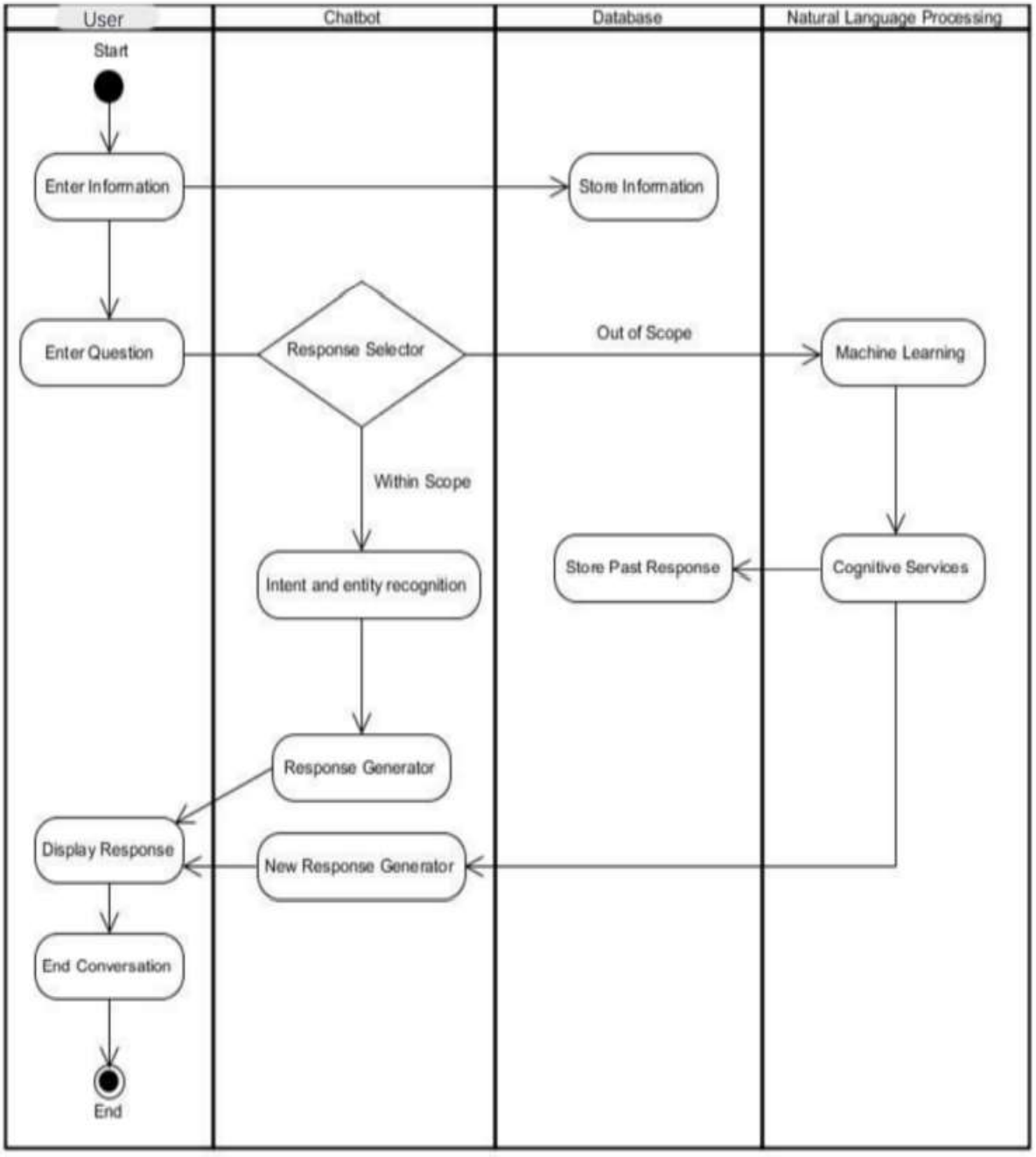


Fig 5.2.3.1 Activity Diagram of Chatbot

The following represents the Activity diagram for the login:

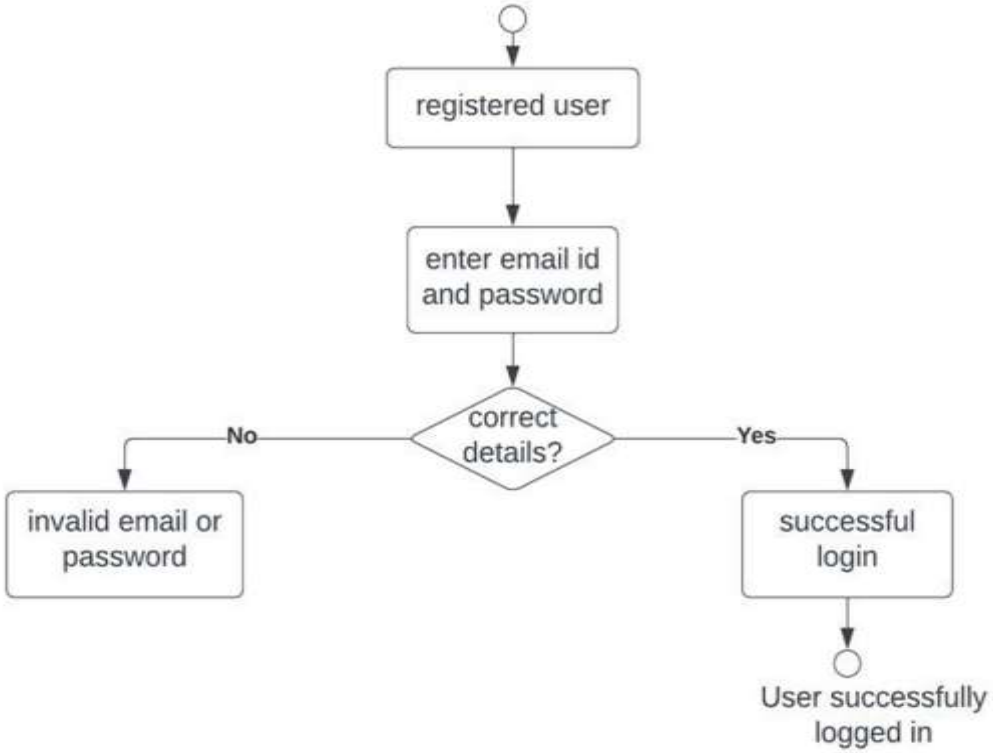


Fig 5.2.3.2 Activity Diagram of login

The following represents the Activity diagram for the Registration:

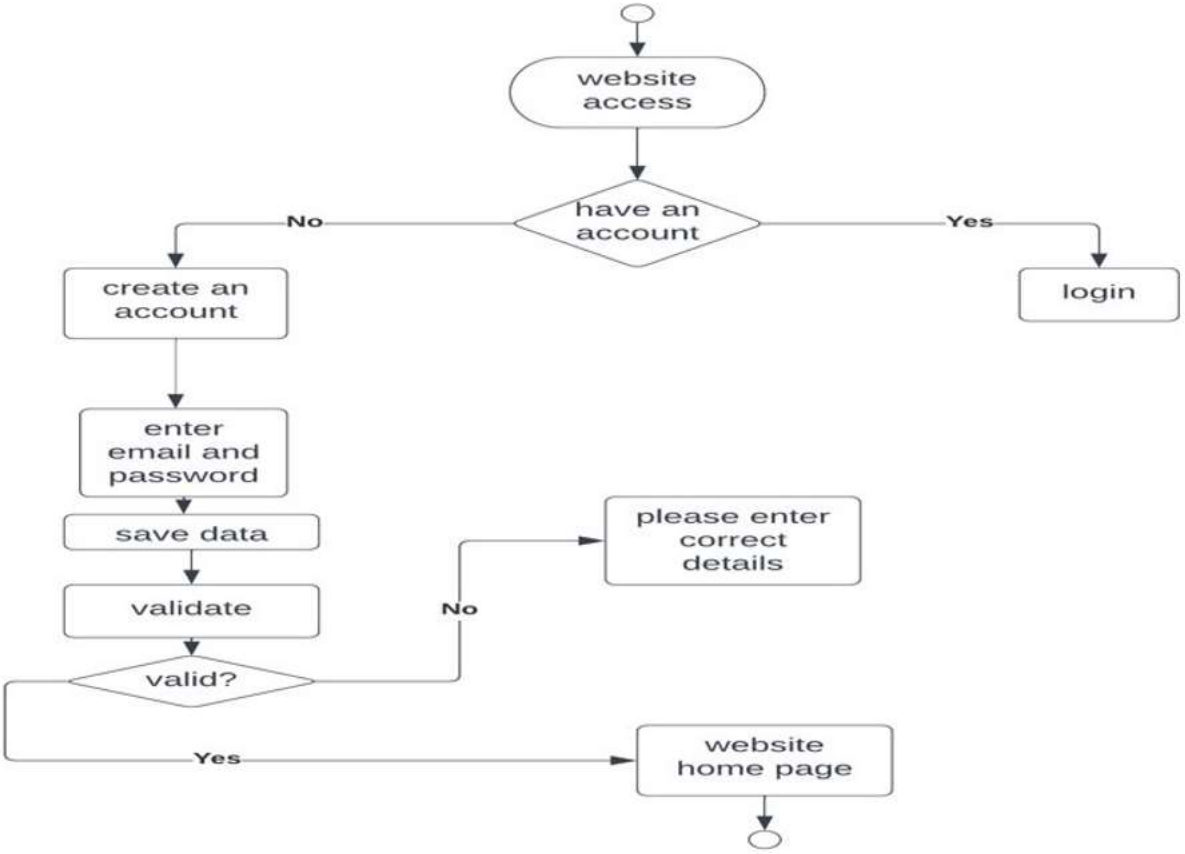


Fig 5.2.3.3 Activity Diagram of Registration

### 5.2.4 Sequence Diagram

A sequence diagram, otherwise called a system sequence diagram (SSD), shows process collaborations in time order in the domain of computer programming. It portrays the cycles that are locked in, as well as the messages that are passed between them to achieve the functionality.

The following represents the sequence diagram of the Chatbot:

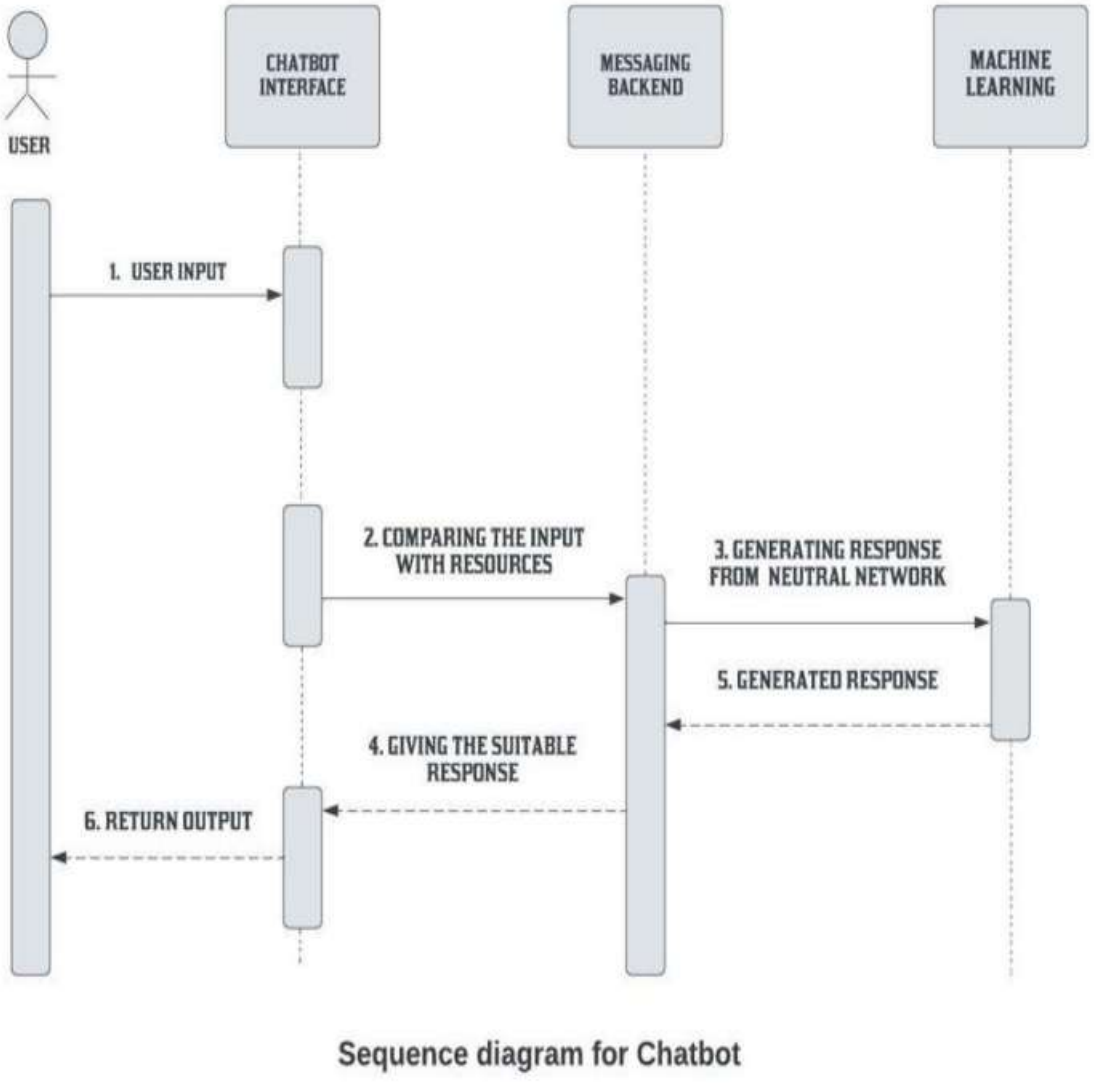


Fig 5.2.4.1 Sequence Diagram of Chatbot

The following represents the Sequence diagram for the login:

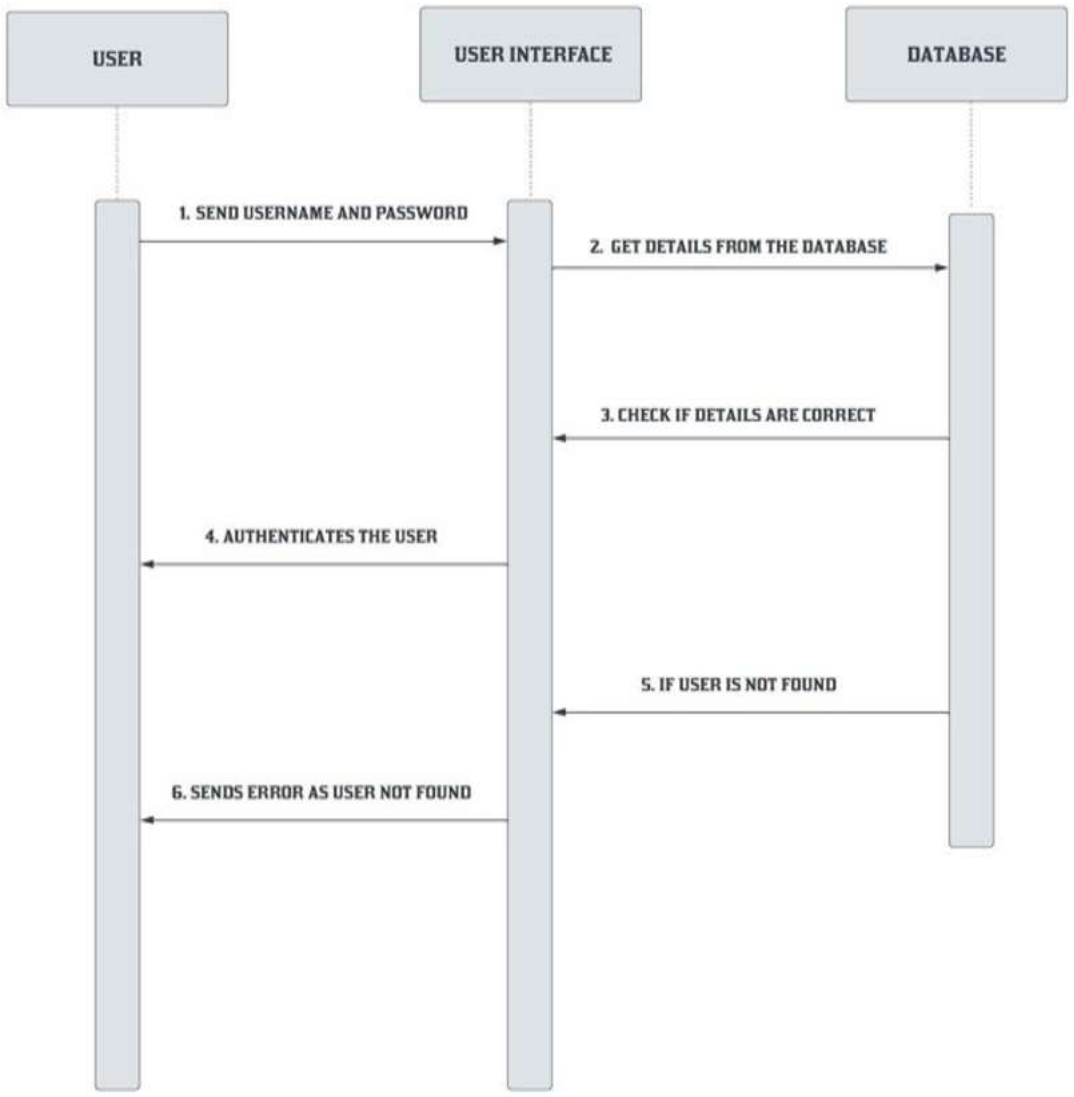


Fig 5.2.4.2 Sequence Diagram of login

The following represents the sequence diagram for the registration:

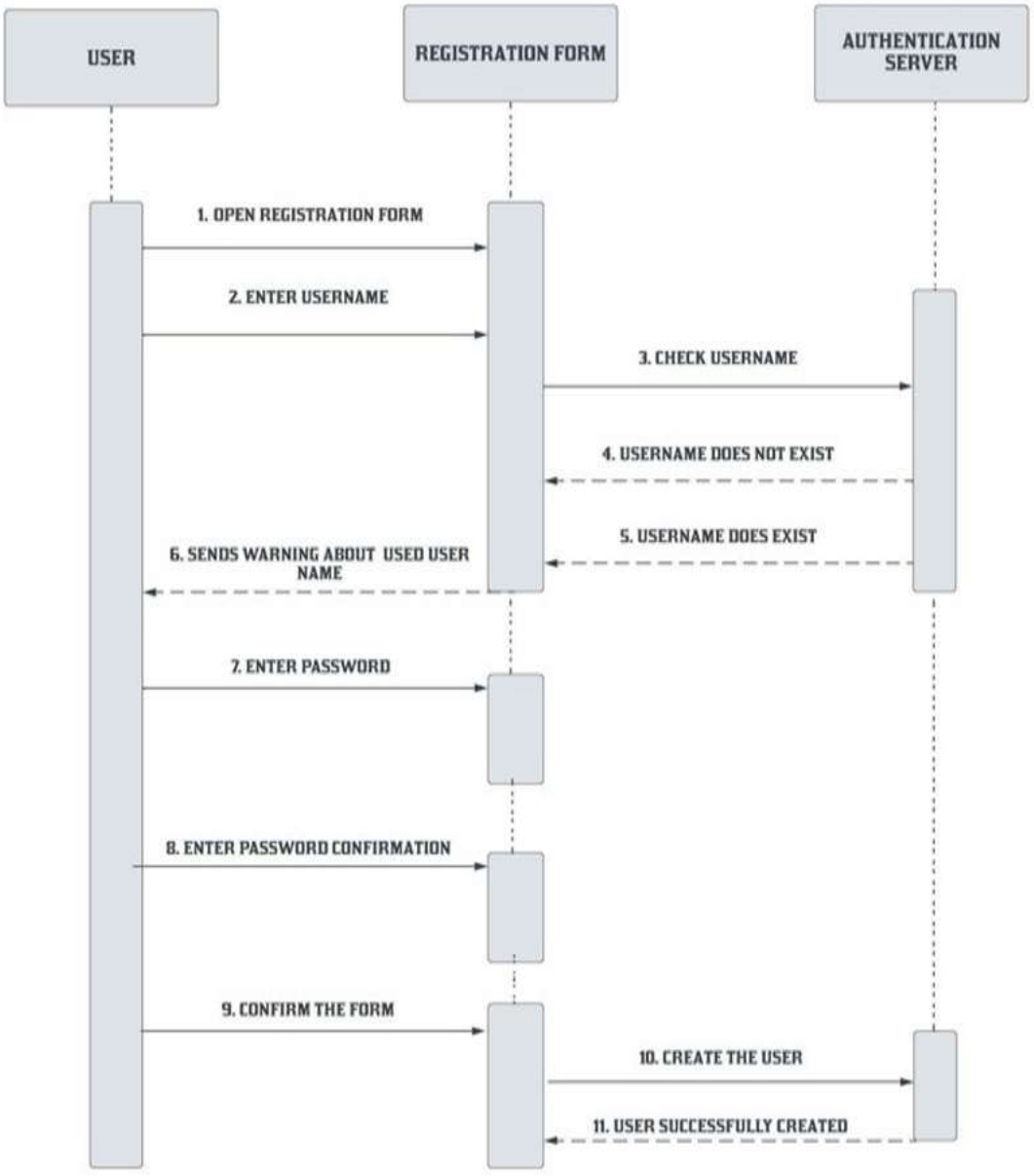


Fig 5.2.4.3 Sequence Diagram of Registration

### 5.2.5 Object Diagram

Object diagrams can be depicted as an instance of class diagrams. In this manner, these graphs are all the more near genuine situations where we carry out a framework. Object diagrams are a bunch of objects and their relationship is very much like class diagrams. They likewise address the static perspective on the system. The utilization of object diagrams is similar to class diagrams yet they are utilized to fabricate a model of a framework according to a practical perspective.

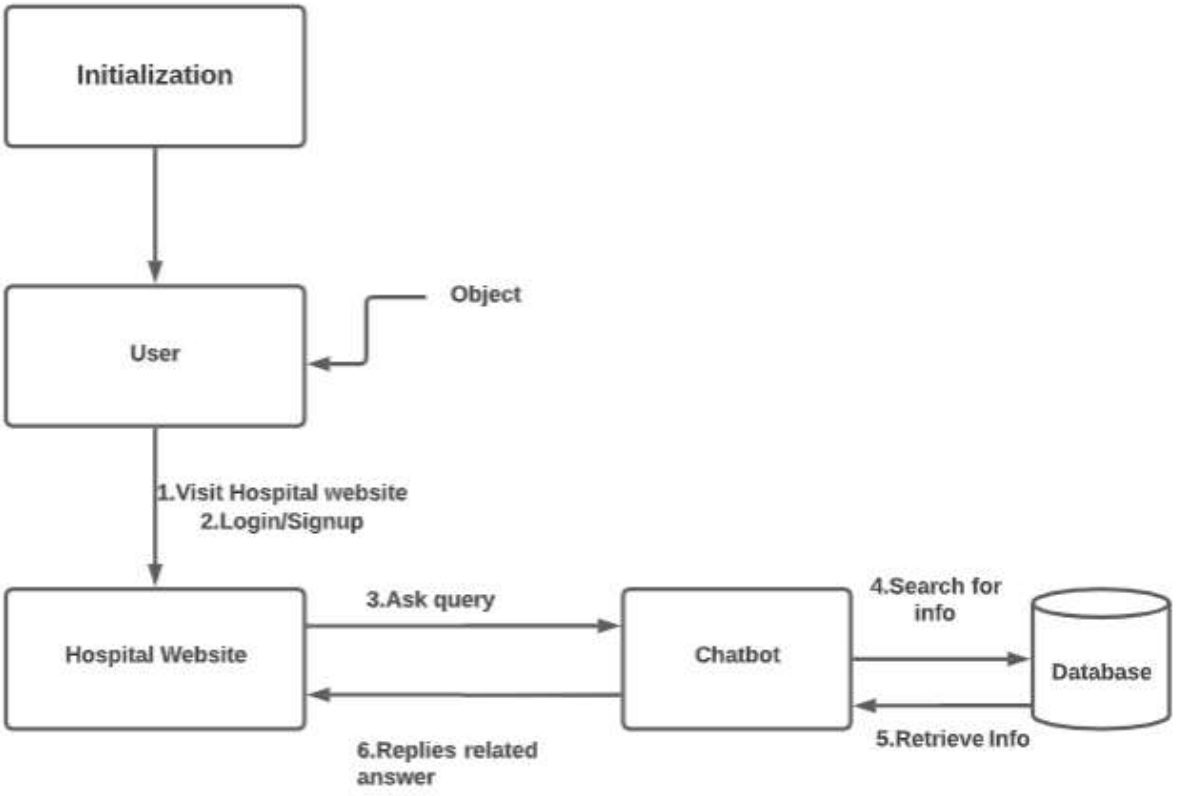


fig 5.2.5.1 Object Diagram of Chatbot

### 5.2.6 Component Diagram

Component diagrams address a set of components and their connections. These components comprise classes, points of interaction, or collaborations. Component diagrams address the execution perspective on a framework.

During the design stage, programming curios (classes, interfaces, and so forth) of a framework are organized in different groups relying on their relationship. Presently, these gatherings are known as components.

At long last, it very well may be said Component diagrams are utilized to visualize the execution.

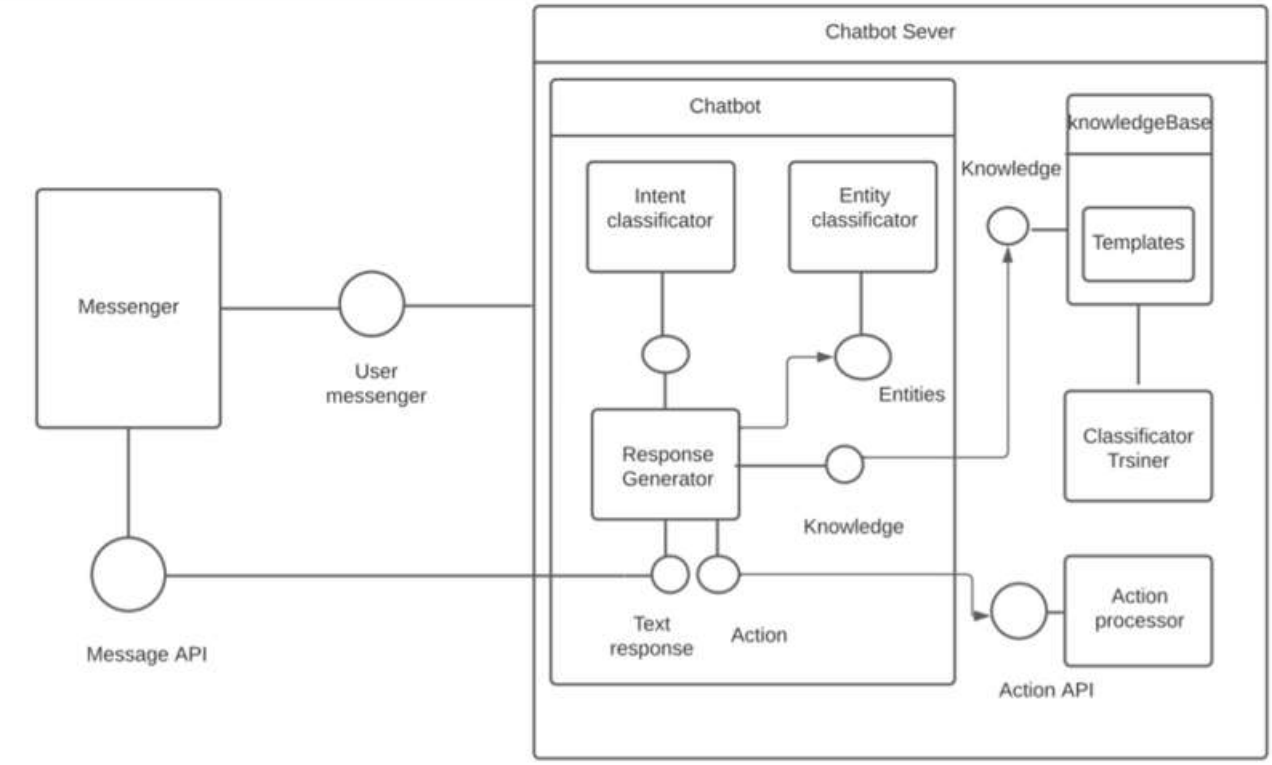


Fig 5.2.6.1 Component Diagram of Chatbot

# CHAPTER 6 - METHODOLOGY

The proposed system focuses on integrating all basic features in one place in an application and powering it with an AI chat bot further adds new functionalities like easy navigation, access to data on doctors, diagnostics information, symptom analysis, precautionary or instant medication suggestions and appointment booking, all these in a single application. Further, considering people who cannot write fluently, those with special needs and those in emergency situations, both voice and text input formats are accepted by the chatbot.

We are building the website using Flask, which contains Login and registration page, dashboard of website, appointment booking and viewing pages and also the animated chatbot button at the end of every page of the website.

Speech enabled chatbots provide higher level of interactivity and usability. User can either give their input using text or speech and similarly chatbot is able to give its response by either text or voice. In our project, this process of conversion between text and speech is done by using speech_recognition and pyttsx3 python modules.

- Voice Input by User (Speech to Text):

Using systems inbuilt microphone live audio input can be transcribed using Google's Web Speech API (recognize_google()).By using adjust for ambient noise function we can set the engine to listen to ambient noise for some time period(here 2 seconds) and adjust energy threshold accordingly. If speech Recognizer unable to detect the speech correctly, respective error messages will be given as response.

- Voice Output by Bot (Text to Speech):

Pyttsx3 is a Text to Speech Conversion Python Library. Using pyttsx3.init() an engine instance will be created for which we can set various properties like voice rate, volume level and also voices (male or female).We can directly pass the text that need to be converted to voice to this engine and output will be voice saying the text accordingly.

User gives a question to interact with the chatbot. After that, a model developed with LSTM analyses the user query. LSTMs are a kind of recurrent neural network that, as opposed to just passing its result into the following section of the network, plays out a progression of math tasks to work on its memory.

Step 1: The three information sources enter the LSTM and are directed to the forget or learn entryways. Long term information is shipped off the forget entryway, where some of it gets lost (the irrelated parts). The learn gate receives the short-term information and "E." This gate determines what information will be gathered.

Step 2: Data that goes through the forget entryway (it isn't neglected; failed to remember data stays at the door) and the learn entryway (it is learned) will be shipped off the remember entryway (which makes new long-term memory) and the utilization entryway (which updates momentary memory).

Sequential model is created with 3 layers, The first layer has 160 neurons and the second layer has 80 neurons. Both the first and second layer utilizes relu activation function. rectifiedLinear activation function (ReLU) is a direct capacity that will yield the data clearly expecting positive some other way it will give yield as nothing. For the last layer, number of neurons will be same as the number of intents of predicted output with a softmax activation function.

Now the model has been created, Stochastic Gradient Descent optimization technique is used to find the minimum possible cost function. Starting from a fundamental worth, Gradient Descent runs iteratively to find the best potential gains of the limits to find the base conceivable worth of the given cost function.SGD is liked as it is not difficult to carry out and is efficient.

The training data is fit to the model and using 1000 epochs best accuracy of 94.85 and minimum loss of 8.15 is obtained.

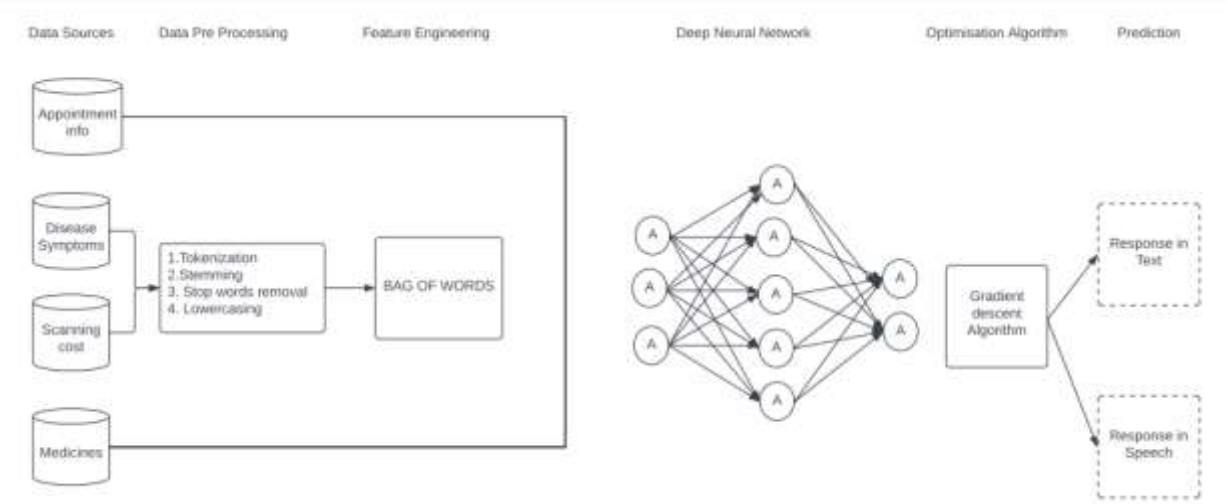


Figure 6.1: work flow of the mode

# CHAPTER 7 – IMPLEMENTATION

## 7.1 Data Preparation

Data preparation is the first step of the project. In data collection phase we collected records about diseases and their respective symptoms, medicines for the diseases, tests and scanning costs and some general greeting functions and navigation routings. All the collected data is organized in the JSON format with tags, patterns and responses fields.

Medical data is collected and coordinated in JSON format as it bears the cost of the adaptability to portray the information. The data preparation wants distinctive interest on structural patterns along with the content. A JSON refers to the JavaScript Object Notation format used to save easy objects and information structures. Usually, JSON documents are backup files, which are used to take backup of data that restored to the application when needed.


Fig 7.1.1: JSON Intents

The dataset consists of 146 documents,41 classes and unique 184 lemmatized words

```
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data]   Unzipping corpora/wordnet.zip.
146 documents
41 classes ['Asthma_symptoms', 'Consultation', 'Depression_symptoms', 'Diabetes_sympto
184 uniqe lemmatized words ["'s", '(', ')', ',', ';', 'a', 'ache', 'age', 'am', 'and',
```

## 7.2 Data Pre-Processing

Normal Language Processing (NLP) is a part of Data Science that manages Text information. Aside from mathematical information, Text information is accessible by and large which is utilized to examine and take care of business issues. Yet, prior to involving the information for investigation or expectation, it means quite a bit to handle the information.

To set up the text information for the model structure we perform text preprocessing. It is the absolute initial step of NLP projects. A portion of the preprocessing steps are:

- Tokenization
- Stemming
- Removing Stop words
- Lemmatization

**Tokenization:** It is the method of breaking a given word into smaller words called ‘tokens’. The preprocessing of a large text in NLP starts from Tokenization level. It can be done at various levels like words, character and sub word level. We have used NLTK tokenization, which is the opensource python library for NLP,it contains various levels of tokenization based on user preference like word_toenize , sent_tokenize , Treebank Word tokenizer..etc.

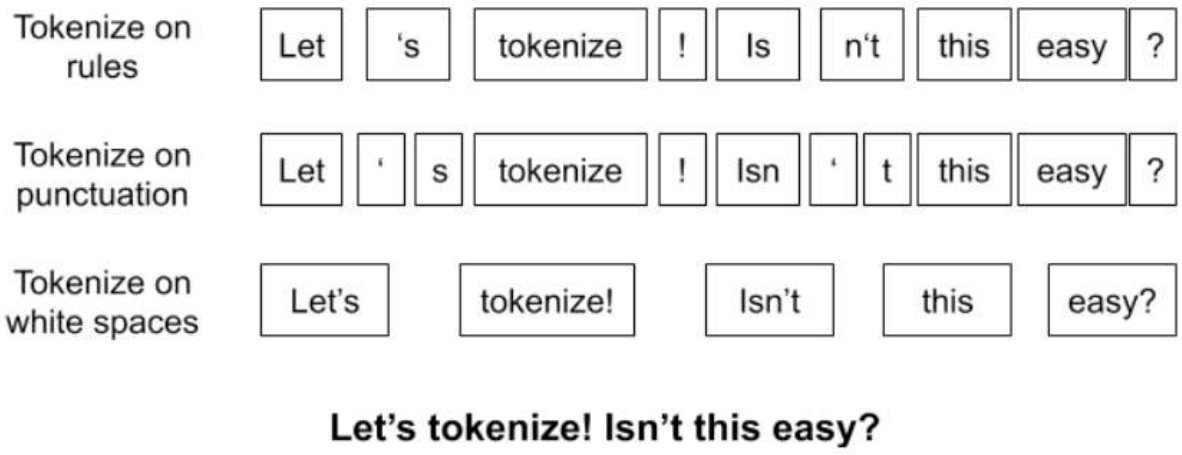


Fig 7.2.1 : Tokenization

**Stemming:** Stemming algorithms work by removing the end or the start of the word, considering a rundown of normal prefixes and postfixes that can be tracked down in a curved word. This unpredictable cutting can find success in certain events, however not dependably, and for that reason, we confirm that this approach presents a few restrictions.

| Form | Suffix | Stem |
|---|---|---|
| studies | -es | studi |
| studying | -ing | study |
| niñas | -as | niñ |
| niñez | -ez | niñ |

Fig 7.2.2: Stemming

**Lemmatization:** It is the process of reducing a given word into its canonical form such that root word is called as 'lemma'. The technique involves gathering the curved pieces of a word in a manner that can be perceived as a solitary component. This is similar to stemming yet the root words have meaning. For example, Lemmatization extracts the base form of 'studies' to 'study' that has same meaning whereas, Stemming will remove 'es' part and converts to 'studi' which results in cutting of the ends of word which doesn't actually gives any meaning.

| Form | Morphological information | Lemma |
|---|---|---|
| studies | Third person, singular number, present tense of the verb study | study |
| studying | Gerund of the verb study | study |
| niñas | Feminine gender, plural number of the noun niño | niño |
| niñez | Singular number of the noun niñez | niñez |

Fig 7.2.3 : Lemmatization

**Stop words Removal:** Some really well-known words that give off an impression of being of little worth in aiding the choice of distributions that address a user's issues are once in a while totally eliminated from the dictionary. These are known as stop words

| Sample text with Stop Words | Without Stop Words |
|---|---|
| GeeksforGeeks – A Computer Science Portal for Geeks | GeeksforGeeks , Computer Science, Portal ,Geeks |
| Can listening be exhausting? | Listening, Exhausting |
| I like reading, so I read | Like, Reading, read |

Fig 7.2.4 : Removing stop words

## 7.3 Feature Engineering

A bag of words is a Natural Language Processing (NLP) strategy of text displaying. In specialized terms, we can say that it is a strategy for feature extraction with text information. This approach is a basic and adaptable approach to extricating features from records.

A bag of words is a display of text that depicts the event of words inside a report. We simply monitor word counts and negligence the linguistic details and the word order. It is known as a "bag" of words in light of the fact that any data about the request or construction of words in the report are disposed of. The model is just worried about whether realized words happen in the document, not where in the document.

Steps involved in the Bag of words technique are:

1. **Pre-processing**

For building a BoW mode, we want to pre-process the information that is being taken care of into the model. Before the stage, we have proactively done handling the JSON information. We pre-handled and changed the text over completely to lowercase, eliminated the stop words (!,.,?) and silent letters (like a)

2. **Keywords Identification:**

Now, after the pre-handling of information is being finished. We currently recognize the

keywords, which hold a vital component or importance. For instance: In the sentence, "They have a lab close to show and tell" the words (lab and show and tell) are the keywords. To recognize these, as a matter of some importance, we proclaim a variable as a word referring to protecting these words i.e., bag of words. Then we do tokenization for each sentence present in the information to words. From that point onward, for each expression of the sentence, we test against the bag of words and check whether it exists there. On the off chance that in the event that it does, we increment the count by 1, and on the off chance that it doesn't coordinate, we add it to our sack of words (word reference) and set it to consider 1.

### 3. Building the BoW model:

This is the last step and here we construct the bag of words model. In this step, we collect a vector, which would illuminate us regardless of whether an expression in each sentence is a regular word. In the event that a word in a sentence is the most happening or successive word, we set it as 1, else we set it as nothing


| | the | red | dog | cat | eats | food |
|---|---|---|---|---|---|---|
| 1. the red dog → | 1 | 1 | 1 | 0 | 0 | 0 |
| 2. cat eats dog → | 0 | 0 | 1 | 1 | 1 | 0 |
| 3. dog eats food → | 0 | 0 | 1 | 0 | 1 | 1 |
| 4. red cat eats → | 0 | 1 | 0 | 1 | 1 | 0 |

Fig 7.3.1: Bag of words example

## 7.4 Building and training the model

The sequential model is developed with 3 layers, the First layer has 160 neurons and the second layer has 80 neurons. Both the first and second layer utilizes relu initiation work. Rectified Linear activation function (ReLU) is a straight capacity that will yield the info straightforwardly assuming positive any other way it will give yield as nothing. For the last layer, the number of neurons will be equivalent to the number of intents of anticipated yield with a softmax initiation work.

```
model = Sequential()
model.add(Dense(160, input_shape=(len(train_x[0]),),activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(80, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(len(train_y[0]), activation='softmax'))
```

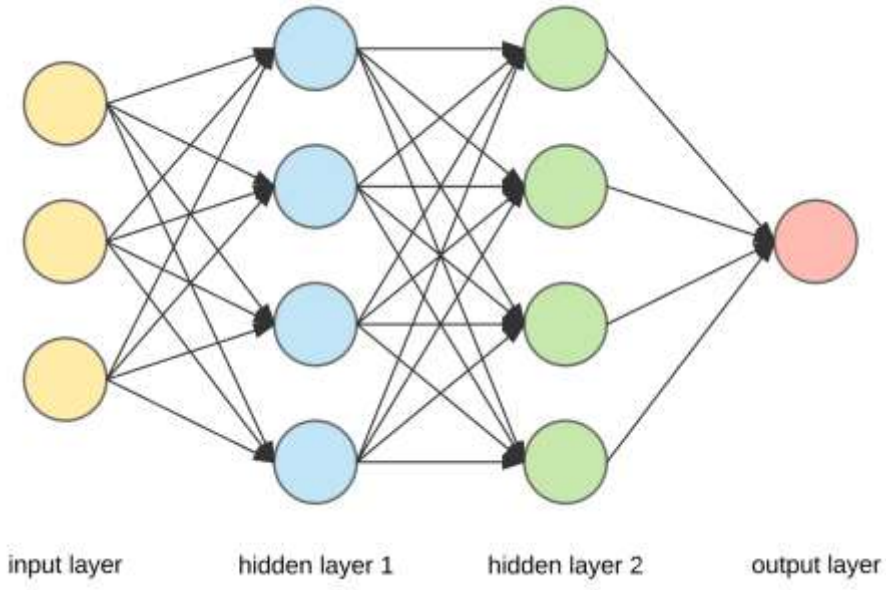


Fig 7.4.1: Neural Network with 2 hidden layers

For the developed model, the Stochastic Gradient Descent optimization strategy is utilized to track down the base conceivable cost function. Beginning from the starting value, Gradient Descent runs iteratively to track down the ideal upsides of the boundaries to find the base conceivable worth of

the given cost function. SGD is liked as it is not difficult to execute and is productive.

```
sgd = SGD(lr = 0.01, decay=1e-6, momentum=0.9, nesterov=True)
model.compile(loss='categorical_crossentropy', optimizer = sgd, metrics=['accuracy'])
```

The training data is fit to the model and using 1000 epochs best accuracy of 94.52 and minimum loss of 0.1972 is obtained.

```
hist = model.fit(np.array(train_x), np.array(train_y), epochs=1000, batch_size=5, verbose=1)
model.save('Chatbot_model.h5', hist)
print('Model created')
```

```
30/30 [==============================] - 0s 2ms/step - loss: 0.0607 - accuracy: 0.9795
Epoch 997/1000
30/30 [==============================] - 0s 1ms/step - loss: 0.1643 - accuracy: 0.9589
Epoch 998/1000
30/30 [==============================] - 0s 1ms/step - loss: 0.2175 - accuracy: 0.9521
Epoch 999/1000
30/30 [==============================] - 0s 2ms/step - loss: 0.1230 - accuracy: 0.9315
Epoch 1000/1000
30/30 [==============================] - 0s 2ms/step - loss: 0.1972 - accuracy: 0.9452
Model created
```

Fig 7.4.2 : Accuracy Ouput

## 7.5 Performance analysis

### Optimization function

Optimization is the cycle where we train the model iteratively that outcomes in a most extreme and least capacity assessment. It is one of the main peculiarities in Machine Learning to get better results. We have involved the Stochastic Gradient Descent Algorithm for the purpose of optimization. The term stochastic means arbitrariness on which the calculation depends. In stochastic gradient descent, rather than taking the entire dataset for every cycle, we arbitrarily select the batches of information. That implies we just take not many examples from the dataset.

$$w := w - \eta \nabla Q_i(w).$$

The method is first to choose the underlying boundaries w and learning rate n. Then, at that point, arbitrarily mix the information at every emphasis to arrive at an estimated least.

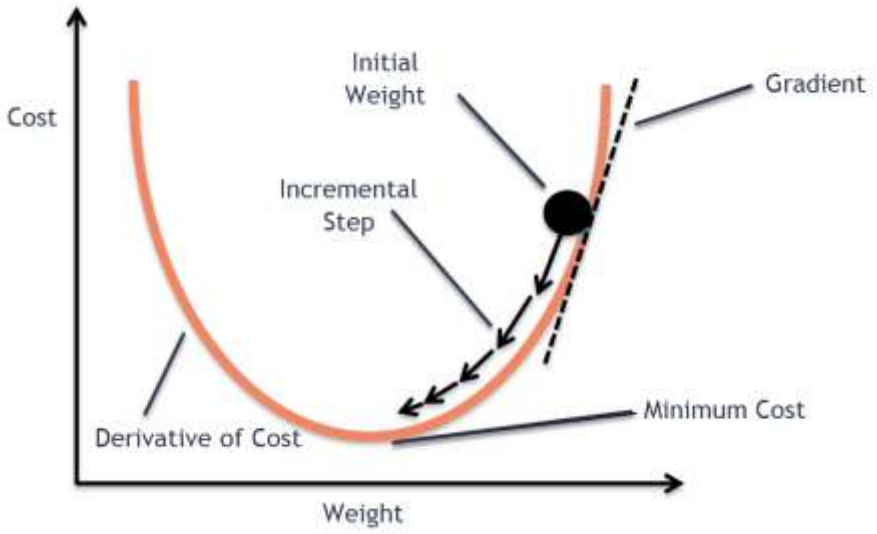


Fig 7.5.1: Gradient Descent cost vs weight graph

Since we are not utilizing the entire dataset but the clusters of it for every cycle, the way taken by the calculation is loaded with noises when contrasted with the gradient descent algorithm. In this manner, SGD utilizes a bigger number of iterations to arrive at the nearby minima. Because of an expansion in the number of cycles, the general calculation time increments. However, even in the wake of expanding the number of cycles, the calculation cost is still not exactly that of the gradient descent algorithm. So the end is in the event that the information is gigantic and computational time is a fundamental element, stochastic gradient descent ought to be liked over gradient descent algorithm.

**Loss function:**

Loss is a measure that addresses the summation of blunders in our model. It estimates how well (or awful) our model is doing. Assuming the blunders are high, the loss will be high, and that implies that the model doesn't work really well. In any case, the lower it is, the better our model works.
To calculate the loss, a loss or cost function is used. There are several different cost functions to use. Each penalizes errors in different ways, and the problem determines which one is better to use. Cross-Entropy and Mean Squared Error are the most commonly used for classification and regression problems, respectively.

**Accuracy:**

Accuracy (Precision) is clearer. It estimates how well our model predicts by contrasting the model expectations and the genuine qualities regarding rate.

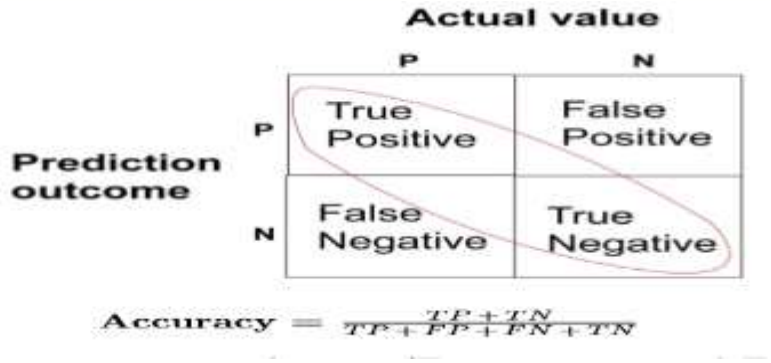


Fig 7.5.2 : Accuracy calculation

Having a low accuracy however a high misfortune would imply that the model makes large blunders in the greater part of the information. In any case, assuming both loss and accuracy are low, it implies the model makes little mistakes in the greater part of the information. Notwithstanding, on the off chance that they're both high, it makes huge mistakes in a portion of the information. Finally, if the accuracy is high and the loss is low, then the model makes small errors on just some of the data, which would be the ideal case.

For our model we have got accuracy 94.52% and loss 19.72% for 1000 epochs.

```
30/30 [==============================] - 0s 2ms/step - loss: 0.0607 - accuracy: 0.9795
Epoch 997/1000
30/30 [==============================] - 0s 1ms/step - loss: 0.1643 - accuracy: 0.9589
Epoch 998/1000
30/30 [==============================] - 0s 1ms/step - loss: 0.2175 - accuracy: 0.9521
Epoch 999/1000
30/30 [==============================] - 0s 2ms/step - loss: 0.1230 - accuracy: 0.9315
Epoch 1000/1000
30/30 [==============================] - 0s 2ms/step - loss: 0.1972 - accuracy: 0.9452
Model created
```

Fig 7.5.3 : Accuracy and Loss of Chatbot Model

# CHAPTER 8 – RESULTS

BOTAID provides multiple features which makes it a user-friendly web application. We have used flask for the front-end development and Google's speech to text API to enable users to give voice queries and convert text replies of bot to voice.

The features which of botaid are:

- User Authentication
- Chatbot
- Appointment Booking
- View booked appointments

**User Authentication**: For any user to access the website, they must login first. If user doesn't have login account, they can register themselves first and the data will be stored into mongo collection. For registration, user must specify their username, email and can set any password.

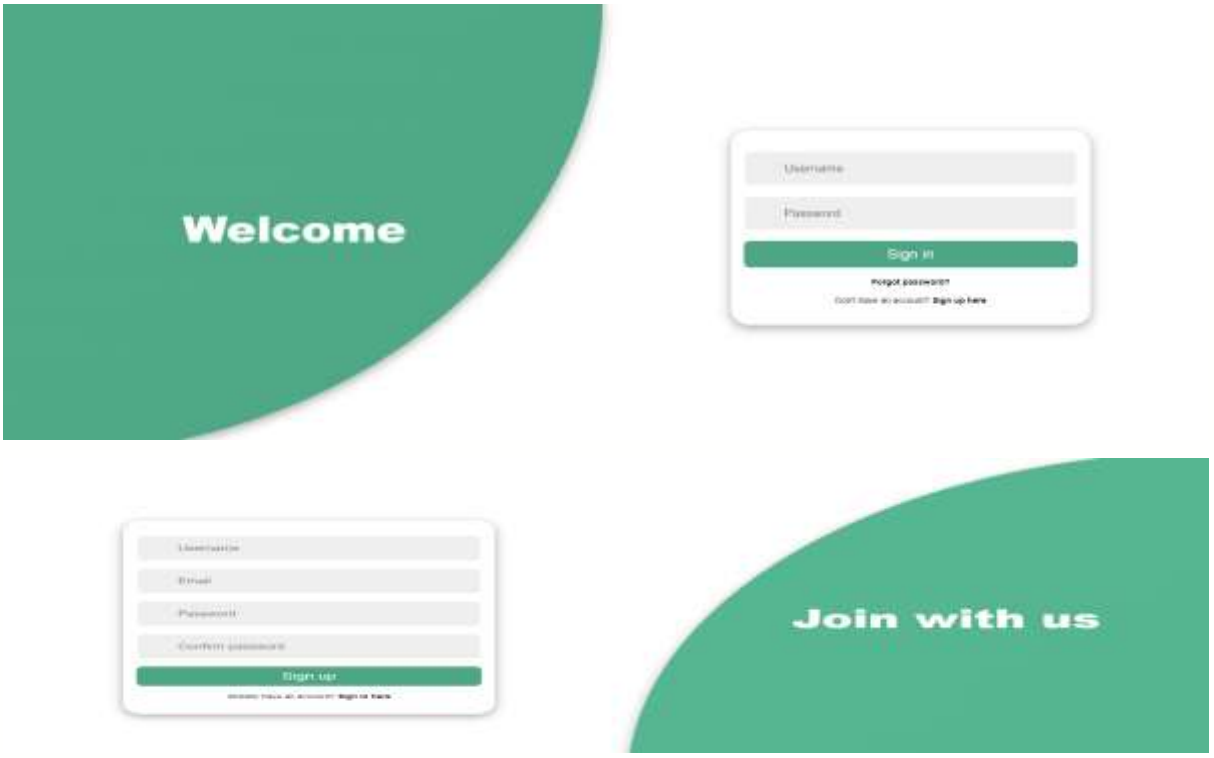


Fig 8.1 -: Login and registration webpage

In order to login user must enter their username and password, the details are then verified using mongo queries and verified users can access application. After successful login, session for that specific user will be created. Session can be used to store user's data across multiple pages of the application.

**Chatbot:** Chatbot is the major feature of our application. The aim of this chatbot is resolve the user queries in the best way possible. We have used deep neural network techniques like LSTM to classify user query and generate perfect response. The JSON dataset is sent as input to the model, which contains tags and responses that matches to particular pattern. We used NLTK to preprocess the data. To resolve ambiguity, we'll either capitalize or lowercase the user input question.

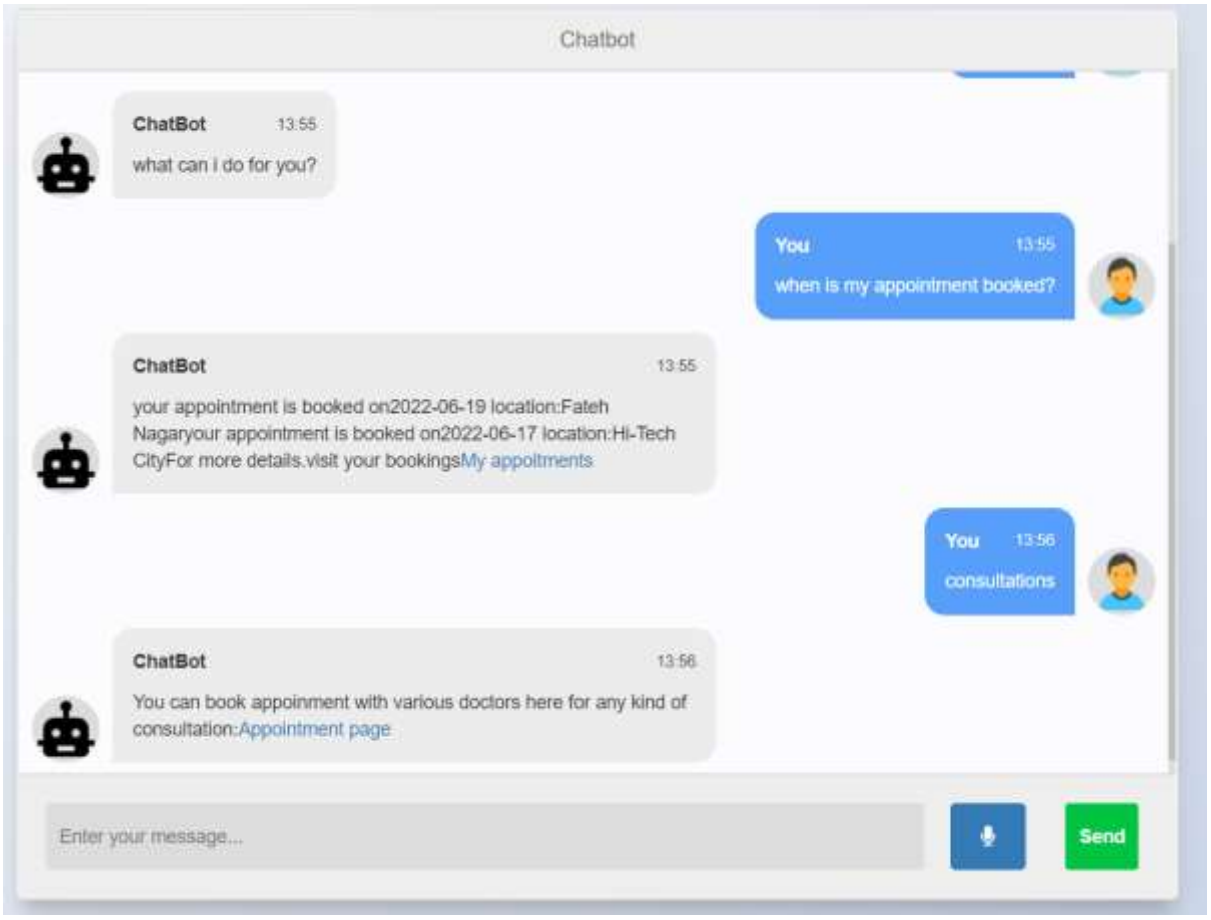


Fig 8.2 : Chatbot Interface

Whenever user requests for an appointment booking or view their booked appointments, chatbot will provide navigation links to those respective parts of the website, which when clicked page will be opened in new tab.

**Appointment Booking:** This feature provides an easy way for user to book their appointment. It can be accessed by clicking the Book an appointment card on the dashboard.

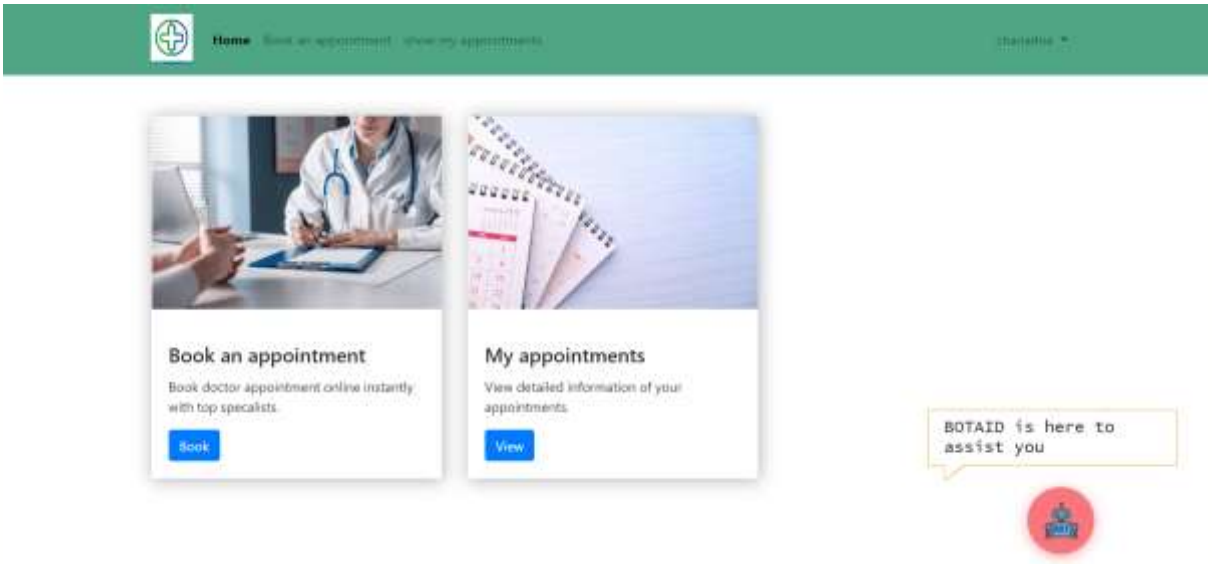


Fig 8.3 : Dashboard

A form needs to be filled to book the appointment.It contains following fields Paient name,gender,Contact number,appointment date,hospital location and department.After filling all the details and clicking submit, a new appointment record will be stored under users name in the database.

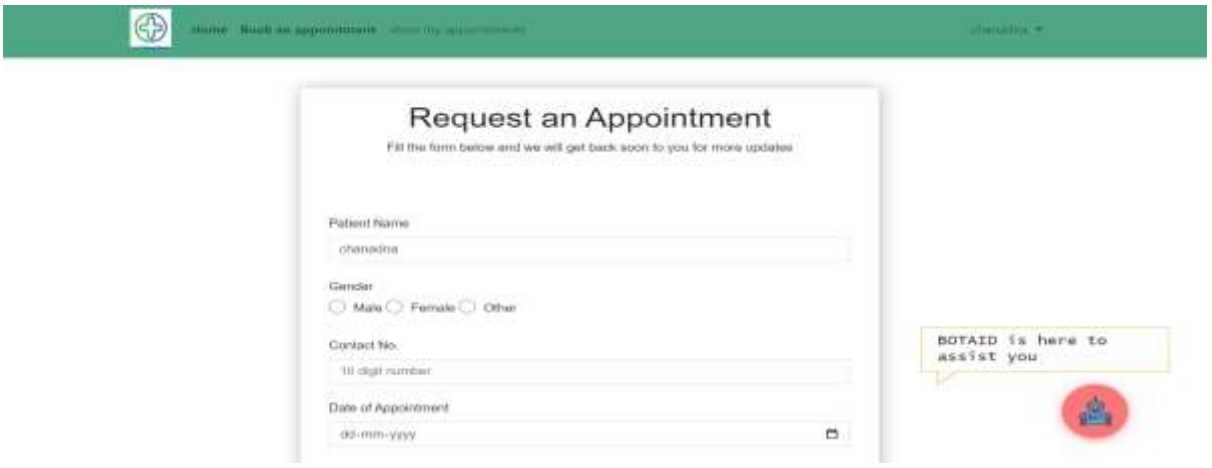


Fig 8.4 : Appointment page

**View booked appointments:** This feature is used to view the appointments booked by that particular logged in user.User's data is protected by displaying only that user's data,not the others.This is implemented using flask session.

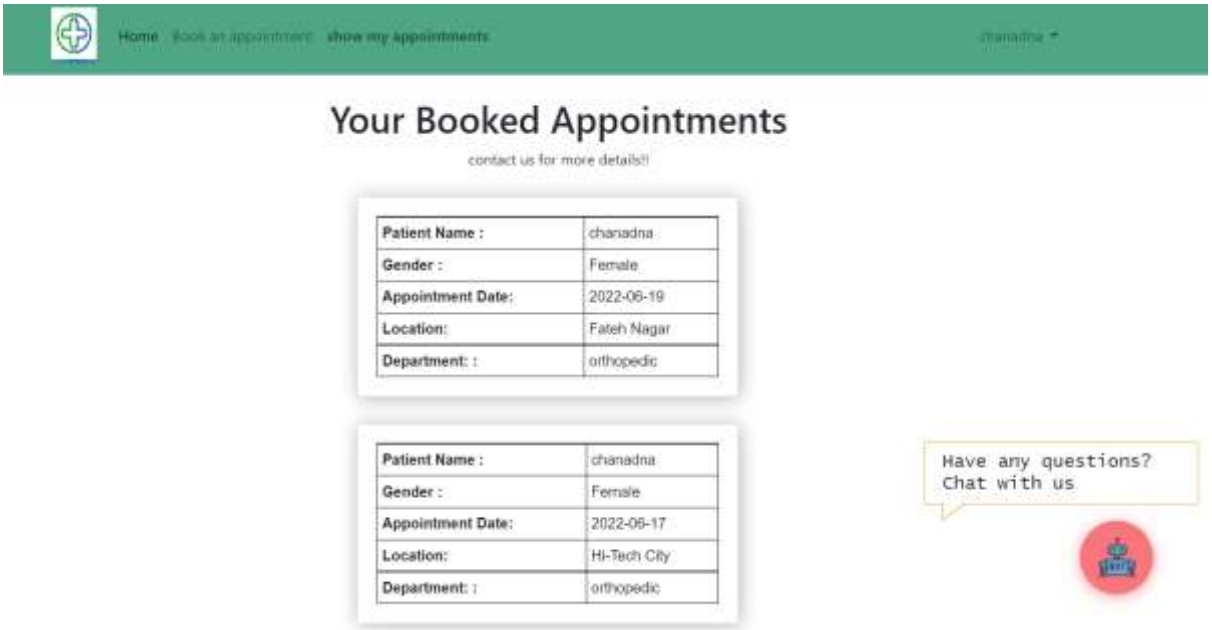


Fig 8.5 : View appointments page

# CHAPTER 9 - CONCLUSION

In the quickly developing universe of AI, consumers are requiring more mechanical assistance for all told aspects of their lives. The web gives different ways of actuating data and has drastically impacted the manner in which individuals convey. Innovation has upgraded our lives with substantially more open doors. There is a requirement for innovation that tends to the inquiries of clients all day, every day. One such innovation is a Chabot. A Chabot is only a man-made program that effectively interfaces with clients to help and tackle their questions. The services that a chatbot can convey are very particular and different as they could help from everyday fundamental questions to huge modern requirements. Chatbots can arrive at an enormous scope of the crowd and be more practical than people. Before long sooner rather than later, they will develop to be a proficient data gathering tool inside the not-so-distant future.

Our Hospital Management System chatbot is to automate repeated tasks in a user-friendly manner such that it will provide hospital employees to focus on important tasks and also to enable fast response for customer instead of waiting for employee to solve their queries as user can interact with bot anytime. Enabling Speech recognition in our chatbot also helps customers to have a simple and fast conversation. The user interactive UI provides better navigation through the website.

We have tested our application by trying various kinds of profiles. The results were satisfactory.

# CHAPTER 10 - FUTURE WORKS

There can always be a scope of improvement for any project. In this we can incorporate more department specific data to make the chatbot more diverse, as of now disease diagnosis, appointment booking, viewing user appointment details and test costs are included. More refined data will give more scope for user to resolve their queries. The application can be integrated with any hospital website to make it more reachable for users.

We can include other features like queries using images, receive SMS alerts to remind the appointment bookings and take multiple language inputs via text and speech.

Other areas where we can do some future works are:

**Sentimental Analysis:** Sentimental analysis assists with understanding the client's feelings and perspective by breaking down their question input as either by text or voice. This investigation empowers chatbots to more readily equip discussions and conveys the right reactions. Lately, sentimental analysis is likewise assuming a key part.

It contributed to developing the EQ (Emotional Quotient ) of the chatbot.

**Data Modules:** The more data a chatbot can access, the higher it performs, which in turn improves customer service. At present, the only data added is regarding the placements, clubs, facilities, and general information (ragging, timetable, and schedules).In the future, there is a scope for adding more data and making it more authentic and more accurate.

# CHAPTER 11 -REFERENCES


[1] Mittal M, Battineni G, Singh D,Nagarwal T, Yadav P. Web-based chatbot for Frequently Asked Queries (FAQ) in Hospitals. J Taibah Univ Med Sc2021;16(5):740e746

[2] Mathew, Rohit Binu; Varghese, Sandra; Joy, Sera Elsa; Alex, Swanthana Susan (2019). [IEEE 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI) - Tirunelveli, India (2019.4.23-2019.4.25)] 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI) - Chatbot for Disease Prediction and Treatment Recommendation using Machine Learning. (), 851–856. doi:10.1109/ICOEI.2019.8862707

[3] Harsh Mendapara, Suhas Digole, Manthan Thakur, Anas Dange;International Journal of Scientific Research and Engineering Development-– Volume 4 Issue 2, Mar- Apr 2021;AI Based Healthcare Chatbot System by Using Natural Language Processing

[4] Siddhi Pardeshi, Suyasha Ovhal, Pranali Shinde, Manasi Bansode, Anandkumar Birajdar;International Research Journal of Engineering and Technology (IRJET);Volume: 07 Issue: 05 | May 2020;A survey on Different Algorithms used in Chatbot

[5] Athota, Lekha; Shukla, Vinod Kumar; Pandey, Nitin; Rana, Ajay (2020). [IEEE 2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO) - Noida, India (2020.6.4-2020.6.5)] 2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO) - Chatbot for Healthcare System Using Artificial Intelligence. , (), 619–622. doi:10.1109/ICRITO48877.2020.9197833

[6] Dammavalam Srinivasa Rao, K. Lakshman Srikanth, J. Noshitha Padma Pratyusha, M. Sucharitha, M. Tejaswini and T. Ashwini, "Development of Artificial Intelligence Based Chatbot Using Deep Neural Network", In: Raju Pal and Praveen Kumar Shukla (eds), SCRS CONFERENCE PROCEEDINGS ON INTELLIGENT SYSTEMS, SCRS, India, 2021, pp. 143-151. https://doi.org/10.52458/978-93-91842-08-6-12

[7] K.W.M.C. Maduwantha, V.N. Vithana “MumCare”: An Artificial Intelligence Based Assistant”, International Journal Of Electrical And Computer Engineering Research, Vol. 1 No. 1 (2021)

[8] Rashmi Dharwadkar, Neeta A. Deshpande “A Medical Chatbot”, International Journal of Computer Trends and Technology, 2018

[9] Jitendra Chaudhary, Vaibhav Joshi, Atharv Khare, Rahul Gawali, Asmita Manna “A

Comparative Study of Medical Chatbots", International Research Journal of Engineering and Technology (IRJET), 2021

[10] A. Ait-Mlouk and L. Jiang, "KBot: A Knowledge Graph Based ChatBot for Natural Language Understanding Over Linked Data," in IEEE Access, vol. 8, pp. 149220-149230, 2020, doi: 10.1109/ACCESS.2020.3016142.

[11] Rarhi, Krishnendu & Bhattacharya, Abhishek & Mishra, Abhishek & Mandal, Krishnasis. (2017). Automated Medical Chatbot. SSRN Electronic Journal. 10.2139/ssrn.3090881.

[12] Reyner, Andrew & Tjiptomongsoguno, Wibowo & Chen, Audrey & Sanyoto, Hubert & Irwansyah, Edy & Kanigoro, Bayu. (2020). Medical Chatbot Techniques: A Review. 10.1007/978-3-030-63322-6_28.

[13] Xu, L., Sanders, L., Li, K., & Chow, J. (2021). Chatbot for Health Care and Oncology Applications Using Artificial Intelligence and Machine Learning: Systematic Review. JMIR cancer, 7(4), e27850. https://doi.org/10.2196/27850

# CHAPTER 12: SOURCE CODE

## 12.1 BACKEND CODE

### 12.1.1 Code for importing required libraries

First we need to import the required libraries for our chatbot.

```
import nltk
from nltk.stem import  WordNetLemmatizer
lemmatizer = WordNetLemmatizer()
import json
import pickle
import numpy as np
from keras.models import Sequential
from keras.layers import Dense, Activation, Dropout
import random
from keras.optimizers import gradient_descent_v2
sgd = gradient_descent_v2.SGD(...)
```

### 12.1.2 Code for loading the dataset

```
words = []
documents = []
classes = []
ignore_words = ['?', '!']

# Load json file
data_file = open('/content/intents.json').read()
intents = json.loads(data_file)
```

### 12.1.3 Code for tokenizing the sentence

```
# iterate through the patterns and tokenize sentence using nltk.word_tokenize()
nltk.download('punkt')

for intent in intents['intents']:
    for pattern in intent['patterns']:
        # tokenize each word
        w = nltk.word_tokenize(pattern)
        words.extend(w)

        # add documents in the corpus
        documents.append((w, intent['tag']))

        #create a list of classes for our tags.
        # add to our classes list
        if intent['tag'] not in classes:
            classes.append(intent['tag'])
```

### 12.1.4 Code for lemmatization

We'll now lemmatize each word and eliminate any duplicates from the list. Lemmatizing is the process of reducing a given word into its canonical form such that root word is called as ‘lemma’**.** All these lemmatized words are stored in pickle files which can be used during predication.

```
nltk.download('wordnet')

#  lemmatize each word and remove duplicate words from the list
words = [lemmatizer.lemmatize(w.lower()) for w in words if w not in ignore_words]

# sort words&classes
words = sorted(list(set(words)))
classes = sorted(list(set(classes)))

# documents = combination between patterns and intents
print(len(documents), 'documents')

# classes = intents
print(len(classes), 'classes', classes)

# words = all words, vocabulary
print(len(words), 'uniqe lemmatized words', words)

# creating a pickle file to store the Python objects which we will use while predicting.
pickle.dump(words, open('words.pkl', 'wb'))
pickle.dump(classes, open('classes.pkl', 'wb'))
```

### 12.1.5 Code for creating training and testing dataset

In this step we'll prepare the training data, which will incorporate both the input and output. The example will be our input, and the class to which our pattern design has a place will be our result. Utilizing bag of words, the text will be changed over into a vector of numbers as computer can't read the language directly.

```
# create our training data
training = []
# create an empty array for our output
output_empty = [0] * len(classes)
# training set, bag of words for each sentence
for doc in documents:
    # initialize our bag of words
    bag = []
    # list of tokenized words for the pattern
    pattern_words = doc[0]
    # lemmatize each word - create base word, in attempt to represent related words
    pattern_words = [lemmatizer.lemmatize(word.lower()) for word in pattern_words]

    # create our bag of words array with 1, if word match found in current pattern
    for w in words:
        bag.append(1) if w in pattern_words else bag.append(0)

    # output is '0' for each tag and '1' for current tag (for each pattern)
    output_row = list(output_empty)
    output_row[classes.index(doc[1])] = 1
    training.append([bag, output_row])
# shuffle our features and turn into np.array
random.shuffle(training)
training = np.array(training)
# create train and test lists. X - patterns, Y - intents
train_x = list(training[: , 0])
train_y = list(training[:, 1])

print("Training data created")
print("training data shape", training.shape)
```

### 12.1.6 Code for building the model

We currently have our training data prepared readily and will develop a three-layer neural network model. For this, we utilize the Keras consecutive API. In the wake of preparing the model for 1000 epochs, we had the option to achieve 94.52% precision. we named the model 'chatbot model.h5' and saved it.

```
# create 3 layers. First layer 128 neurons,
# second layer 64 neurons
# 3rd output layer contains number of neurons equal to number of intents to predict output intent with softmax
import matplotlib.pyplot as plt
model = Sequential()
model.add(Dense(160, input_shape=(len(train_x[0]),),activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(80, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(len(train_y[0]), activation='softmax'))


# Compile model. Stochastic gradient descent with Nesterov accelerated gradient
# gives good results for this model
sgd = SGD(lr = 0.01, decay=1e-6, momentum=0.9, nesterov=True)
model.compile(loss='categorical_crossentropy', optimizer = sgd, metrics=['accuracy'])

#fitting and saving the model
history = model.fit(np.array(train_x), np.array(train_y),epochs=1000, batch_size=5, verbose=1)
model.save('Chatbot_model.h5', history)
print('Model created')
```

## 12.2 CODE FOR INTEGRATING CHATBOT WITH WEB

```
model = load_model('./Chatbot_model.h5')
intents = json.loads(open('./intents.json').read())
words = pickle.load(open('./words.pkl', 'rb'))
classes = pickle.load(open('./classes.pkl', 'rb'))
def clean_up_sentence(sentence):
    # tokenize the pattern - split words into array
    sentence_words = nltk.word_tokenize(sentence)
    # stem each word - create short form for word
    sentence_words = [lemmatizer.lemmatize(
        word.lower()) for word in sentence_words]
    return sentence_words
# return bag of words array: 0 or 1 for each word in the bag that exists in the sentence
def bow(sentence, words, show_details=True):
    # tokenize the pattern
    sentence_words = clean_up_sentence(sentence)
    # bag of words - matrix of N words, vocabulary matrix
    bag = [0]*len(words)
    for s in sentence_words:
        for i, w in enumerate(words):
            if w == s:
                bag[i] = 1
                if show_details:
                    print("found in bag: %s" % w)
    return(np.array(bag))
def predict_class(sentence, model):
    # filter out predictions below a threshold
    p = bow(sentence, words, show_details=False)
    res = model.predict(np.array([p]))[0]
    ERROR_THRESHOLD = 0.25
    results = [[i, r] for i, r in enumerate(res) if r > ERROR_THRESHOLD]
    # sort by strength of probability
    results.sort(key=lambda x: x[1], reverse=True)
    return_list = []
    for r in results:
        return_list.append({"intent": classes[r[0]], "probability": str(r[1])})
    return return_list
```

## 12.3 CODE FOR CONNECTING MONGO AND INITIALISING SESSION TO OUR PROJECT

```
app = Flask(__name__)
app.static_folder = 'static'
app.config["SESSION_PERMANENT"] = False
app.config["SESSION_TYPE"] = "filesystem"
Session(app)
app.config["MONGO_URI"] = "mongodb://localhost:27017/chatbot"
mongodb_client = PyMongo(app)
db = mongodb_client.db
app.secret_key = "abc"
```

## 12.4 CODE FOR LOGIN AND REGISTRATION PAGE

### 12.4.1 Login and registration page Frontend

```
<body>
    <div id="container" class="container"><!-- FORM SECTION -->
        <div class="row"><!-- SIGN UP -->
            <form id="signup-form" method="POST" action="/signup" class="col align-items-center flex-col sign-up">
                <div class="form-wrapper align-items-center">
                    <div class="form sign-up">
                        <div class="input-group">
                            <i class='bx bxs-user'></i>
                            <input type="text" placeholder="Username" name="Username">
                        </div>
                        <div class="input-group">
                            <i class='bx bx-mail-send'></i>
                            <input type="email" placeholder="Email" name="Email">
                        </div>
                        <div class="input-group">
                            <i class='bx bxs-lock-alt'></i>
                            <input type="password" placeholder="Password" name="Password">
                        </div>
                        <div class="input-group">
                            <i class='bx bxs-lock-alt'></i>
                            <input type="password" placeholder="Confirm password" name="Confirmpwd">
                        </div>
                        <button> Sign up</button>
                        <p>
                            <span>Already have an account?</span>
                            <b onclick="toggle()" class="pointer">Sign in here</b>
                        </p>
                    </div>
                </div>
            </form><!-- END SIGN UP --><!-- SIGN IN -->

            <form id="login-form" method="POST" action="/login" class="col align-items-center flex-col sign-in">
                <div class="form-wrapper align-items-center">
                    <div class="form sign-in">
```

```
    <form id="login-form" method="POST" action="/login" class="col align-items-center flex-col sign-in">
        <div class="form-wrapper align-items-center">
            <div class="form sign-in">
                <div class="input-group">
                    <i class='bx bxs-user'></i>
                    <input type="text" placeholder="Username" name="Username">
                </div>
                <div class="input-group">
                    <i class='bx bxs-lock-alt'></i>
                    <input type="password" placeholder="Password" name="Password">
                </div>
                <button> Sign in</button>
                <p><b>Forgot password?</b></p>
                <p>
                    <span> Don't have an account?</span>
                    <b onclick="toggle()" class="pointer">
                        Sign up here
                    </b>
                </p>
            </div>
        </div>
        <div class="form-wrapper">
        </div>
    </form><!-- END SIGN IN -->
</div> <!-- END FORM SECTION --> <!-- CONTENT SECTION -->
<div class="row content-row"><!-- SIGN IN CONTENT -->
    <div class="col align-items-center flex-col">
        <div class="text sign-in">
            <h2>Welcome</h2>
        </div>
        <div class="img sign-in">
        </div>
    </div><!-- END SIGN IN CONTENT --><!-- SIGN UP CONTENT -->
    <div class="col align-items-center flex-col">
        <div class="img sign-up">
        </div>
```

### 12.4.2 Code for Login & Registration Backend

```
@app.route("/login",methods=['GET'])
def get_login():
    return render_template("login.html")
@app.route("/signup",methods=['POST'])
def post_signup():
    if request.method=='POST':
        data = {}
        data["Username"]=request.form['Username']
        data["Email"]=request.form['Email']
        data["Password"]=request.form['Password']
        db.userDetails.insert_one(data)
        session["name"] = data["Username"]
        return render_template('dashboard.html',value=session["name"])
@app.route("/login",methods=['POST'])
def post_login():
    if request.method=='POST':
        username=request.form['Username']
        pwd=request.form['Password']
        if username!="" and pwd!="":
            res=db.userDetails.find({'Username':username});
            result = []
            for data in res:
                result.append(data)
            if result[0]['Password']==pwd:
                flash(f"Record Saved!", "success")
                session["name"] = username
                return redirect(url_for("home"))
            else:
                return redirect(url_for("login"))
    return render_template('login.html')
@app.route("/logout")
def logout():
    session["name"] = None
    return redirect("/")
```

## 12.5 CODE FOR DASHBOARD

```
<body>
    <nav class="navbar navbar-expand-lg navbar-light " style="background-color: #4EA685;">
        <div class="container">
            <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarTogglerDemo01"
                aria-controls="navbarTogglerDemo01" aria-expanded="false" aria-label="Toggle navigation">
                <span class="navbar-toggler-icon"></span>
            </button>
            <a class="navbar-brand" href="#">
                <img src="https://thumbs.dreamstime.com/z/medical-logo-cross-logo-medical-center-logo-health-symbols-medical-logo-cross-logo-
                    width="50" height="60">
            </a>
            <div class="collapse navbar-collapse" id="navbarTogglerDemo01">
                <ul class="navbar-nav mr-auto mt-2 mt-lg-0">
                    <li class="nav-item active">
                        <a class="nav-link" href="http://127.0.0.1:5000/dashboard"><b>Home</b> <span class="sr-only">(current)</span></a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="http://127.0.0.1:5000/appointment">Book an appointment</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="http://127.0.0.1:5000/myappointments">show my appointments</a>
                    </li>
                </ul>
                <ul class="navbar-nav ">
                    <!-- PROFILE DROPDOWN - scrolling off the page to the right -->
                    {%if value!=None %}
                    <li class="nav-item dropdown">
                        <a href="#" class="nav-link dropdown-toggle" id="navDropDownLink" data-toggle="dropdown"
                            aria-haspopup="true" aria-expanded="false">
                            {{value}}
                        </a>
                        <div class="dropdown-menu" aria-labelledby="navDropDownLink">
                            <div class="dropdown-divider"></div>
                            <a class="dropdown-item" href="http://127.0.0.1:5000/logout">Logout</a>
                        </div>
                    </li>
                    ...
                    {%else %}
                    <li class="nav-item">
                        <button class="btn btn-primary"><a class="nav-link disabled"
                                href="http://127.0.0.1:5000/login">Login</a></button>
                    </li>
                    {%endif%}
                </ul>
            </div>
        </div>
    </nav>
    <br><br>
    <section>
        <div class="container">
            <div class="row">
                <div class="col-md-4">

                    <div class="card card-01">
                        <img class="card-img-top"
                            src="https://media.istockphoto.com/photos/doctor-writing-a-medical-prescription-picture-id1319031310?k=20&m=1319031
                            alt="Card image cap">
                        <div class="card-body">
                            <h4 class="card-title">Book an appointment</h4>
                            <p class="card-text">Book doctor appointment online instantly with top specalists.</p>
                            <button class="btn btn-primary"><a href="http://127.0.0.1:5000/appointment" style="color:white;">Book</a></button>
                        </div>
                    </div>
                </div>
                <div class="col-md-4">
                    <div class="card card-01">
                        <img class="card-img-top"
                            src="https://media.istockphoto.com/photos/deadline-concept-with-stack-of-calendar-on-table-with-copy-space-picture-i
                            alt="Card image cap">
                        <div class="card-body">
                            <h4 class="card-title">My appointments</h4>
                            <p class="card-text">View detailed information of your appointments.</p>
                            <button class="btn btn-primary"><a href="http://127.0.0.1:5000/myappointments" style="color:white;">View</a></butt
```

```
<script>
const typSpd = 70;
const waitTime = 500;
const text = [
  "BOTAID is here to assist you",
  "any queries? 24/7 service",
  "Have any questions?Chat with us.",
  "BOTAID is here to assist you"
]
var mi = 0;
function writeString(e, str, i) {
  e.innerHTML = e.innerHTML + str[i];
  if (e.innerHTML.length == str.length && mi != text.length)
    setTimeout(slowlyDelete, waitTime, e);
}
function deleteString(e) {
  e.innerHTML = e.innerHTML.substring(0, e.innerHTML.length - 1);
  if (e.innerHTML.length == 0)
    slowlyWrite(e, text[mi++]);
}
function slowlyDelete(e) {
  for (var i = 0; i < e.innerHTML.length; i++) {
    setTimeout(deleteString, typSpd / 2 * i, e);
  }
}
function slowlyWrite(e, str) {
  for (var i = 0; i < str.length; i++) {
    setTimeout(writeString, typSpd * i, e, str, i);
  }
}
const msg = document.querySelector(".msg-icn");
slowlyDelete(msg);
</script>
```

## 12.6 CODE FOR APPOINTMENT BOOKING PAGE

### 12.6.1 Code for Appointment booking Frontend

```
<div class="col d-flex justify-content-center">
    <div class="card">
        <header class="header">
            <h1 id="title" class="text-center">Request an Appointment</h1>
            <p id="description" class="description text-center">
                Fill the form below and we will get back soon to you for more updates
            </p>
        </header>
        <form id="appointment-form" method="POST" action="/appointment">
            <div class="form-group">
                <label id="name-label" for="name">Patient Name</label>
                <input type="text" name="name" id="name" class="form-control" value={{value}} />
            </div>
            <div class="form-group">
                <label>Gender</label>
                <div></div>
                <label> <input name="gender" value="Male" type="radio" class="input-radio" />Male</label>
                <label> <input name="gender" value="Female" type="radio" class="input-radio" />Female</label>
                <label><input name="gender" value="other" type="radio" class="input-radio" />Other</label>
            </div>
            <div class="form-group">
                <label id="number-label" for="number">Contact No.</label>
                <input name="contactnumber" id="number" class="form-control" placeholder="10 digit number"
                    required />
            </div>
            <div class="form-group">
                <label id="number-label" for="number">Date of Appointment</label>
                <input type="date" name="appointment" id="number" class="form-control" placeholder="DD/MM//YYYY" />
            </div>
            <div class="form-group">
                <label>Hospital Location</label>
                <select id="dropdown" name="location" class="form-control" required>
                    <option disabled selected value>Select Hospital Location</option>
                    <option value="Hi-Tech City">Hi-Tech City</option>
                    <option value="jKompally">Kompally</option>
                    <option value="Fateh Nagar">Fateh Nagar</option>
```

```
                <option value="Patel Nagar">Patel Nagar</option>
                <option value="Somajiguda">Somajiguda</option>
                <option value="Malakpet">Malakpet</option>
            </select>
        </div>
        <div class="form-group">
            <label>Department</label>
            <select id="dropdown" name="department" class="form-control" required>
                <option disabled selected value>Select Department</option>
                <option value="neurology">Neurology</option>
                <option value="cardiology">Cardiology</option>
                <option value="orthopedic">Orthopedic</option>
                <option value="dermatology">Dermatalogy</option>
                <option value="oncology">Oncology</option>
            </select>
        </div>
        <div class="form-group" >
            <button type="submit" id="submit" class="btn btn-success submit-button">
                Submit
            </button>
        </div>
    </form>
</div>
<div>
    <h5 class="msg-icn">So nice</h5>
    <div id="chat-circle" class="btn btn-raised">


        <!--<i class="material-icons">android</i>-->
        <a href="http://127.0.0.1:5000/chatbot"> <img class="chat-circle_robot"
            src="https://cdn.iconscout.com/icon/free/png-256/robot-86-404724.png"></a>
    </div>
```

## 12.6.2 Code for Appointment booking Backend

```
@app.route("/appointment",methods=['GET'])
def appoint():
  return render_template("appointment.html",value=session["name"])

@app.route('/appointment', methods=['POST'])
def bookappointment():

    if request.method=='POST':

        data = {}
        data["name"]=request.form['name']
        data["gender"]=request.form['gender']
        data["contactnumber"]=request.form['contactnumber']
        data["appointment"]=request.form['appointment']
        data["location"]=request.form['location']
        data["department"]=request.form['department']
        db.appointments.insert_one(data)
        flash("your appointment is booked...Check your mail for further details")
        return render_template("dashboard.html",value=session["name"])
```

## 12.7 CODE FOR VIEWING APPOINTMENT PAGE

### 12.7.1 Code for Viewing Appointment Frontend

```
<div class="col d-flex justify-content-center">
    <div class="card1" style="text-align: center;">
        <header class="header">
            <h1 id="title" class="text-center">Your Booked Appointments</h1>
            <p id="description" class="description text-center">
                contact us for more details!!
            </p>
        </header>
        {% for row in data %}
        <div class="card">
        <table style="width:100%">
            <tr>
                <th>Patient Name :</th>
                <td>{{row[0]}}</td>
            </tr>
            <tr>
                <th>Gender :</th>
                <td>{{row[1]}}</td>
            </tr>
            <tr>
                <th>Appointment Date:</th>
                <td>{{row[2]}}</td>
            </tr>
            <tr>
                <th>Location:</th>
                <td>{{row[3]}}</td>
            </tr>
            <tr>
                <th>Department: :</th>
                <td>{{row[4]}}</td>
            </tr>
        </table>
        </div>
        {% endfor %}
    </div>
    <div>
```

### 12.7.1 Code for Viewing Appointment Backend

```
@app.route("/myappointments")
def get_my_appointments():
    res=db.appointments.find({'name':session["name"]})
    result = []
    for data in res:
        result.append(data)
    print(len(result))
    l=[]
    for i in range(len(result)):
        patientname=result[i]["name"]
        gender=result[i]["gender"]
        date=result[i]["appointment"]
        location=result[i]["location"]
        department=result[i]["department"]
        app=[patientname,gender,date,location,department]
        l.append(app)
    print(l)
    return render_template('myappointments.html',value=session["name"],data=l)
```

## 12.8 CODE FOR CHATBOT PAGE

### 12.8.1 Code for Chatbot Frontend

```
<body>
  <!-- partial:index.partial.html -->
  <section class="msger">
    <header class="msger-header">
      <div class="msger-header-title">
        <i class="fas fa-robot"></i> Chatbot <i class="fas fa-robot"></i>
      </div>
    </header>
    <main class="msger-chat">
      <div class="msg left-msg">
        <div class="msg-img" style="background-image: url(https://img.icons8.com/material/48/000000/bot.png)"></div>
        <div class="msg-bubble">
          <div class="msg-info">
            <div class="msg-info-name">Chatbot</div>
          </div>
          <div class="msg-text">
            Hi, welcome to ChatBot! Go ahead and send me a message. 😍
          </div>
        </div>
      </div>
    </main>
    <div class="msger-inputarea">
      <form class="msger-inputarea" style="width: 80%;">
        <input type="text" class="msger-input" id="textInput" placeholder="Enter your message...">
      </form>
      <form class="msger-inputarea" name="execution_form" >
        <button type="submit" class="btn btn-primary"><i class="fa fa-microphone fa-xl" style="font-size: 16px;"></i> </button>
      </form>
      <form class="msger-inputarea" >
      <button type="submit" class="msger-send-btn">Send</button>
    </form>
    </div>
  </section>
  <script type="text/javascript">
    $(document).ready(function () {
      $("form[name='execution_form']").submit(function (evt) {
        evt.preventDefault();
        $.ajax({
          method: "GET",
          url: "/voice",
          contentType: "application/json;charset=utf-8",
          dataType: "json",
        }).done(data => {
          // Use the response here.
          console.log(data.message);
          document.getElementById("textInput").value = data.message
        });
      });
    });
    const msgerForm = get(".msger-inputarea");
    const msgerInput = get(".msger-input");
    const msgerChat = get(".msger-chat");
    const BOT_IMG = "https://img.icons8.com/material/48/000000/bot.png";
    const PERSON_IMG = "https://img.icons8.com/color/48/000000/person-male.png";
    const BOT_NAME = "    ChatBot";
    const PERSON_NAME = "You";
    msgerForm.addEventListener("submit", event => {
      event.preventDefault();
      const msgText = msgerInput.value;
      if (!msgText) return;
      appendMessage(PERSON_NAME, PERSON_IMG, "right", msgText);
      msgerInput.value = "";
      botResponse(msgText);
    });
    function appendMessage(name, img, side, text) {
      //   Simple solution for small apps
      const msgHTML = `
<div class="msg ${side}-msg">
  <div class="msg-img" style="background-image: url(${img})"></div>
```

```
    <div class="msg-info">
      <div class="msg-info-name">${name}</div>

      <div class="msg-info-time">${formatDate(new Date())}</div>
    </div>

    <div class="msg-text">${text}</div>
  </div>
</div>
`;
      msgerChat.insertAdjacentHTML("beforeend", msgHTML);
      msgerChat.scrollTop += 500;
    }

    function botResponse(rawText) {
      $.get("/get", { msg: rawText }).done(function (data) {
        console.log(rawText);
        console.log(data);
        const msgText = data;
        appendMessage(BOT_NAME, BOT_IMG, "left", msgText);
      });
    }
    // Utils
    function get(selector, root = document) {
      return root.querySelector(selector);
    }
    function formatDate(date) {
      const h = "0" + date.getHours();
      const m = "0" + date.getMinutes();
      return `${h.slice(-2)}:${m.slice(-2)}`;
    }

  </script>

</body>
```

## 12.8.2 Code for Chatbot Backend

```
def chatbot_response(msg):
    if msg=="show my appointment" or msg=="view my appointment" or msg=="when is my appointment booked?":
        print(session["name"])
        res=db.appointments.find({'name':session["name"]})
        result = []
        for data in res:
            result.append(data)

        l=[]
        for i in range(len(result)):
            patientname=result[i]["name"]
            gender=result[i]["gender"]
            date=result[i]["appointment"]
            location=result[i]["location"]
            department=result[i]["department"]
            app=[patientname,gender,date,location,department]
            l.append(app)

        res=""
        if(result==[]):
            res="you haven't booked any appointment yet.To book an appointment :<a href='http://127.0.0.1:5000/appointment' target='_blank'>Ap
        else:
            for i in l:
                res+="your appointment is booked on"+i[2]+"\nlocation:"+i[3]
        res+="for more details.visit your bookings<a href='http://127.0.0.1:5000/myappointments' target='_blank'>My appointments</a>"
        text_to_speech(res)
        return res
    else:
        ints = predict_class(msg, model)
        res = getResponse(ints, intents)
        text_to_speech(res)
        return res
```

## 12.9 CODE FOR SPEECH RECOGNITION

### 12.9.1 Code for Speech to Text

```
@app.route("/voice", methods=['GET'])
def sppech2text():
    try:
        with sr.Microphone() as source2:
            r.adjust_for_ambient_noise(source2, duration=0.2)
            audio2 = r.listen(source2)
            MyText = r.recognize_google(audio2)
            MyText = MyText.lower()
            # SpeakText(MyText)
            print(MyText)
            # return render_template('index.html',MyText=MyText)
            return jsonify(message=MyText)
            # return chatbot_response(MyText)
    except sr.RequestError as e:
        res = ("Could not request results; {0}".format(e))
        # return chatbot_response(res)
        print(res)
        return jsonify(message=res)
    except sr.UnknownValueError:
        res = ("Can't recognize speak again")
       # return chatbot_response(res)
        print(res)
        return jsonify(message=res)
```

### 12.9.2 Code for Text to speech

```
def text_to_speech(text):
    gender = "Male"
    voice_dict = {'Male': 0, 'Female': 1}
    code = voice_dict[gender]
    engine = pyttsx3.init()
    # Setting up voice rate
    engine.setProperty('rate', 150)
    # Setting up volume level  between 0 and 1
    engine.setProperty('volume', 0.8)
    # Change voices: 0 for male and 1 for female
    voices = engine.getProperty('voices')
    engine.setProperty('voice', voices[code].id)
    engine.say(text)
    engine.runAndWait()
def SpeakText(command):
    engine = pyttsx3.init()
    voices = engine.getProperty('voices')
    engine.setProperty('voice', voices[1].id)
    engine.say(command)
    engine.runAndWait()
```

# PLAGARISM CHECK

# Batch-A3

*by* Kumaraswamy sudha

**Submission date:** 20-Jun-2022 10:59AM (UTC+0530)
**Submission ID:** 1859928082
**File name:** MAJOR_DOC_plag_test.docx (4.7M)
**Word count:** 8514
**Character count:** 45083

Batch-A3

ORIGINALITY REPORT

| 15% | 6% | 3% | 11% |
|---|---|---|---|
| SIMILARITY INDEX | INTERNET SOURCES | PUBLICATIONS | STUDENT PAPERS |

PRIMARY SOURCES

| | | |
|---|---|---|
| 1 | Submitted to University of North Texas<br>Student Paper | 2% |
| 2 | www.ijsred.com<br>Internet Source | 1% |
| 3 | Submitted to Higher Education Commission Pakistan<br>Student Paper | 1% |
| 4 | Submitted to University of Wales Institute, Cardiff<br>Student Paper | 1% |
| 5 | Submitted to Gitam University<br>Student Paper | 1% |

| | | |
|---|---|---|
| 9 | Submitted to Kingston University<br>Student Paper | 1 % |
| 10 | Submitted to SVKM International School<br>Student Paper | 1 % |
| 11 | www.cm.com<br>Internet Source | 1 % |
| 12 | Submitted to University of Bedfordshire<br>Student Paper | 1 % |
| 13 | Submitted to University System of Georgia (USG)<br>Student Paper | <1 % |
| 14 | Submitted to Sydney Institute of Technology and Commerce<br>Student Paper | <1 % |
| 15 | Submitted to Arab Open University<br>Student Paper | <1 % |
| 16 | Submitted to Midlands State University<br>Student Paper | <1 % |

| | | |
|---|---|---|
| 20 | www.ijitee.org<br>Internet Source | <1 % |
| 21 | Submitted to Plainfield South High School<br>Student Paper | <1 % |
| 22 | Marzieh Kadivar. "Development of high performance materials based on bamboo through densification process", Universidade de Sao Paulo, Agencia USP de Gestao da Informacao Academica (AGUIA), 2020<br>Publication | <1 % |
| 23 | Submitted to Universiti Teknologi MARA<br>Student Paper | <1 % |
| 24 | Submitted to Manchester Metropolitan University<br>Student Paper | <1 % |
| 25 | Submitted to University of New Haven<br>Student Paper | <1 % |
| 26 | www.ijert.org<br>Internet Source | <1 % |

# CHATBOT FOR HOSPITAL MANAGEMENT


Submitted by

Ms. A. LAHARI (18071A1203)
Ms. B. APARNA (18071A1208)
Ms. N. CHANDANA (18071A1243)
Mr. T. RAJESHWAR RAO (18071A1253)

**Under the esteemed guidance of**

Dr. D. Srinivasa Rao
Associate Professor,
Dept. of Information Technology,
VNRVJIET