AI E-Health Chatbot

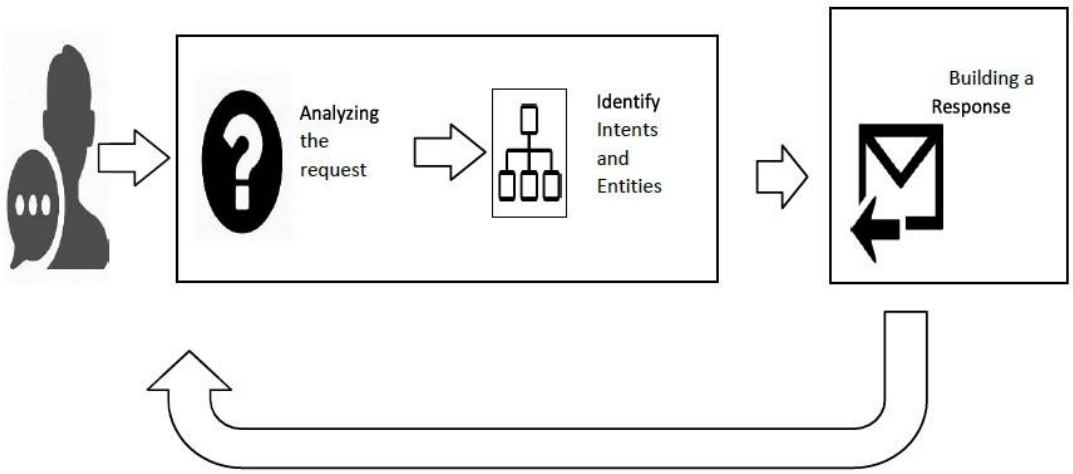
Gowtham Kumar.B

121810302019 – 4B2

Table of Contents:

Introduction

This project is focusing on creating a chatbot to be used by people to get their queries responded easily from the online website. A chatbot is a program which can do real conversations with textual and/or auditory methods. Using Artificial Intelligence (AI), chatbots can simulate human conversations. There are two categories of chatbots. One category is command based chatbots where chatbots rely on a databank of replies and heuristics. The user must be very specific while asking the questions so that the bot can answer. Hence, these bots can answer limited set of questions and cannot perform function outside of the code. The other category is chatbots based on AI or machine learning algorithms, these bots can answer ambiguous questions which means the user do not have to be specific while asking questions. Thus, these bots create replies for the user's queries using Natural Language Processing (NLP).



This is how a Chatbot works.

The above diagram shows how a chatbot works. Whenever a user asks any query, the bot will first analyse the request, then identifies intents and entities, builds a response and sends it back to the user. Now, intents mean intention of the query and entity means details of that query.

AI-powered chatbots are motivated by the need of traditional websites to provide a chat facility where a bot is required to be able to chat with user and solve queries. When live agent can handle only two to three operations at a time, chatbots can operate without an upper limit which really scales up the operations. Also, if any business is receiving lots of queries, having a chatbot on a website takes off the load from support team. Having a chatbot clearly improves the response rate compared to human support team. In addition, since millennial prefer live chats over a phone call, they find a chatbot, which provide a highly interactive marketing platform, very attractive. Furthermore, a chatbot can automate the repetitive tasks. There can be some scenarios where a business receives same queries in a day for many times and support team must respond to each query repetitively. Lastly, the most important advantage of having a chatbot is that it is available 24/7. No matter what time it is, a user can get a query solved. All these advantages of a chatbot

constitute the motivation to implement a Smart AI-EH sector Chatbot.

In order to understand the requirement of a chatbot, consider an example of Amazon Shopping App. In this app, when a customer buys an item, he/she does not have any information about how to return the item. To get this information, the customer must call and wait to talk to customer representative for a long time. However, this whole process is tedious for a customer. Hence, Amazon created a chatbot to answer simple queries of customers.

Similarly, the E-Health Chatbot is designed to help people to get their queries solved on a fingertip.

Examples of Chatbot :

Here are some existing systems:

- ELIZA is the primary chatbot created by joseph Weinbaum utilizing a keyword coordinating strategy. The thought was to persuade the client info and look for certain keywords, if a catchphrase was discovered then the appropriate response was recovered. In the event that a catchphrase is not present then ELIZA would proceed, as per indicated principles, to get more data from the client to keep the discussion going.
- Jabberwocky is one of the earliest attempts at designing an AI through human interaction. It was mainly a form of

entertainment. It aimed to move from a text-based system to wholly voice operated system.

- ALICE is developed by Richard Wallace in 1995. It utilizes design coordinating and stores the data in Artificial Intelligence Mark-up Language records. An AIML record is like a XML document that is created to store design information for chatbots. Code 2 gives a case of a discussion between a human and ALICE.
- Watson, built by IBM is a question answering (QA) computing system designed to apply advanced natural language processing, information retrieval, knowledge representation, automated reasoning, and machine learning technologies to the field of open domain question answering.
- Alexa is a voice service inhabiting the Amazon Echo device. Alexa uses natural language processing algorithms for voice interaction. She uses these algorithms to receive, recognize and respond to voice commands.

Problem Introduction:

In the pandemic situation people are unable to go the hospital or meet the doctor in person.

So in order to establish a virtual communication between the doctor and the patients a chat application or called chatbot is built.

Proposed Method :

The need for e-health inquiry system arises due to various reasons which include:

The slow nature of hospital website, an outsider would not know where to search for a particular piece of information, difficult for the person outside hospital's domain to extract information. The smart solution for all the drawbacks lends to the need of the system. The smart e-health inquiry system will provide the response by summarizing the query and then output answers, it also provides selective information what the user wants. A e-health system will dispense all answers relating to domains such as precautions, medication and symptoms of a disease.

The major features of the chatbot are:

- Any disease related queries could be answered through it.

- People can get to know about the precautions and know in detail about the effects of the particular disease.
- People can fetch particulars about any disease.

E-Health inquiry system will act as a fast, standard and informative widget to enhance health website's user experience and bestow users with righteous information. The bot will analyze user's queries and understand users' message and then reply accordingly. It uses AI & NLP. This way users' time and efforts will be saved and he/she will be equipped with effective answers.

The objectives of this application are :

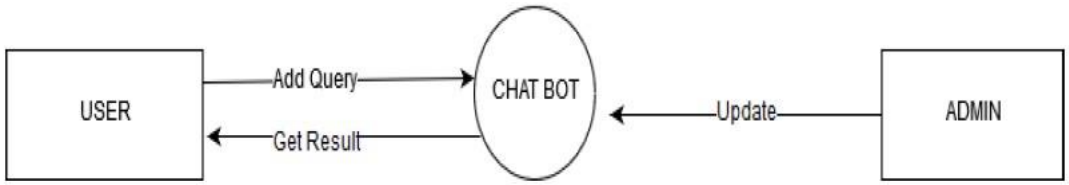
- To analyse users queries and understand users message.
- To provide an answer to the query of the user very effectively.
- To save the time of the user since she/he does not have to personally go to the hospital or medical centre for inquiry.
- This system will help people to be updated about the new diseases evolving.

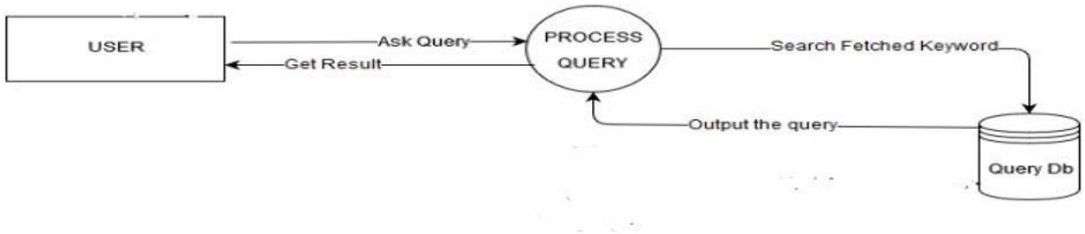
The system will reply using an effective GUI which implies that as if a real person is talking to the user.

System Design :

Data Flow Diagram / Flow Diagram :

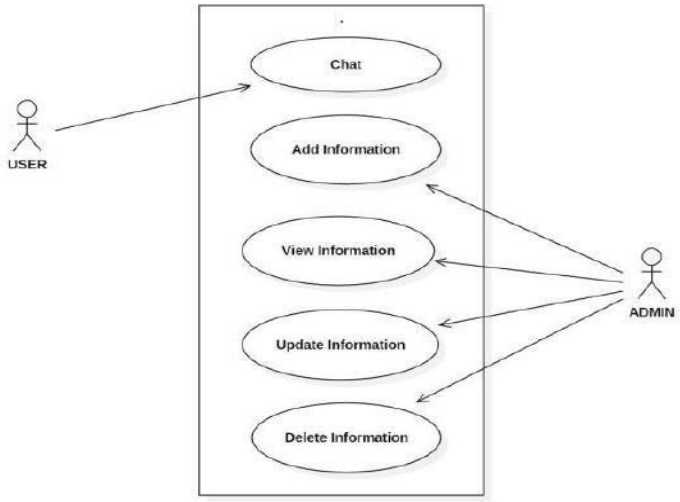
The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of the input data to the system, various processing carried out on these data, and the output data is generated by the system.



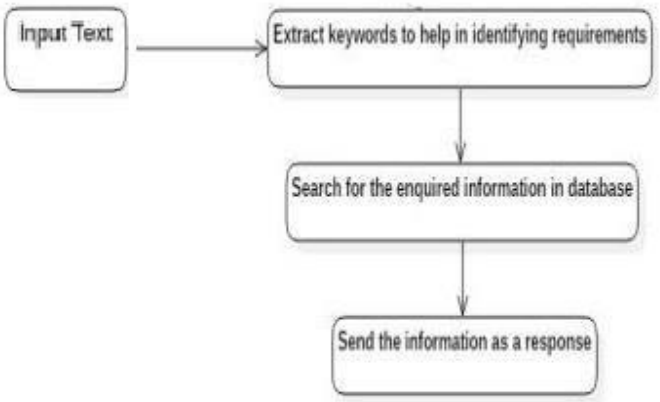


Use Case Diagram :

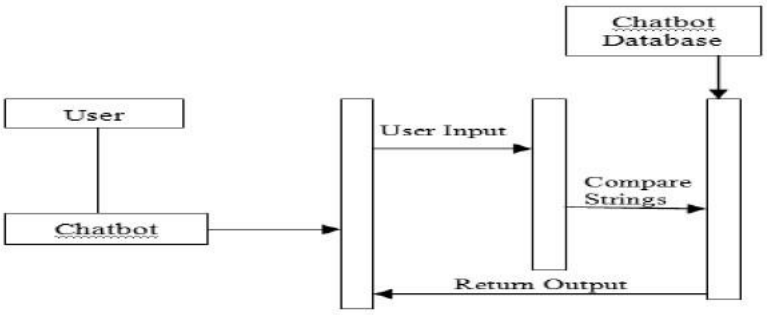
(User and admin roles)



Activity Diagram :



Sequence Diagram :



Code :

Chatbot.py :

chatbot.py

```python
import nltk
nltk.download('punkt')
nltk.download('wordnet')
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
from tensorflow.keras.optimizers import SGD
import random

words=[]
classes = []
documents = []
ignore_words = ['?', '!']
data_file = open('job_intents.json', encoding='utf-8').read()
intents = json.loads(data_file)


for intent in intents['intents']:
    for pattern in intent['patterns']:

        w = nltk.word_tokenize(pattern)
        words.extend(w)

        documents.append((w, intent['tag']))

```

chatbot.py

```python
        documents.append((w, intent['tag']))


        if intent['tag'] not in classes:
            classes.append(intent['tag'])

words = [lemmatizer.lemmatize(w.lower()) for w in words if w not in ignore_words]
words = sorted(list(set(words)))

classes = sorted(list(set(classes)))

print (len(documents), "documents")

print (len(classes), "classes", classes)

print (len(words), "unique lemmatized words", words)


pickle.dump(words,open('words.pkl','wb'))
pickle.dump(classes,open('classes.pkl','wb'))


training = []
output_empty = [0] * len(classes)
for doc in documents:

    bag = []

    pattern_words = doc[0]
    pattern_words = [lemmatizer.lemmatize(word.lower()) for word in pattern_words]
```

chatbot.py

```
    for w in words:
        bag.append(1) if w in pattern_words else bag.append(0)

    output_row = list(output_empty)
    output_row[classes.index(doc[1])] = 1

    training.append([bag, output_row])

random.shuffle(training)
training = np.array(training)

train_x = list(training[:,0])
train_y = list(training[:,1])
print("Training data created")

model = Sequential()
model.add(Dense(128, input_shape=(len(train_x[0]),), activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(64, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(len(train_y[0]), activation='softmax'))

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

hist = model.fit(np.array(train_x), np.array(train_y), epochs=200, batch_size=5, verbose=1)
model.save('chatbot_model.h5', hist)

print("model created")
```

Processor.py :

processor.py

```python
import nltk
from nltk.stem import WordNetLemmatizer
lemmatizer = WordNetLemmatizer()
import pickle
import numpy as np

from keras.models import load_model
model = load_model('chatbot_model.h5')
import json
import random
intents = json.loads(open('job_intents.json', encoding='utf-8').read())
words = pickle.load(open('words.pkl','rb'))
classes = pickle.load(open('classes.pkl','rb'))

def clean_up_sentence(sentence):
    sentence_words = nltk.word_tokenize(sentence)
    sentence_words = [lemmatizer.lemmatize(word.lower()) for word in sentence_words]
    return sentence_words

def bow(sentence, words, show_details=True):

    sentence_words = clean_up_sentence(sentence)
    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))
```

processor.py

```python
    return(np.array(bag))

def predict_class(sentence, model):
    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]
    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

def getResponse(ints, intents_json):
    tag = ints[0]['intent']
    list_of_intents = intents_json['intents']
    for i in list_of_intents:
        if(i['tag']== tag):
            result = random.choice(i['responses'])
            break
        else:
            result = "I am unable to understand you, can you please elaborate"
    return result

def chatbot_response(msg):
    ints = predict_class(msg, model)
    res = getResponse(ints, intents)
    return res

```

App.py :

app.py

```python
from flask import Flask, render_template, jsonify, request
import processor


app = Flask(__name__)

app.config['SECRET_KEY'] = 'enter-a-very-secretive-key-3479373'


@app.route('/', methods=["GET", "POST"])
def index():
    return render_template('index.html', **locals())



@app.route('/chatbot', methods=["GET", "POST"])
def chatbotResponse():

    if request.method == 'POST':
        the_question = request.form['question']

        response = processor.chatbot_response(the_question)

    return jsonify({"response": response })



if __name__ == '__main__':
    app.run(host='0.0.0.0', port='8888', debug=True)

```

JSON File :

job_intents.json > [ ] intents

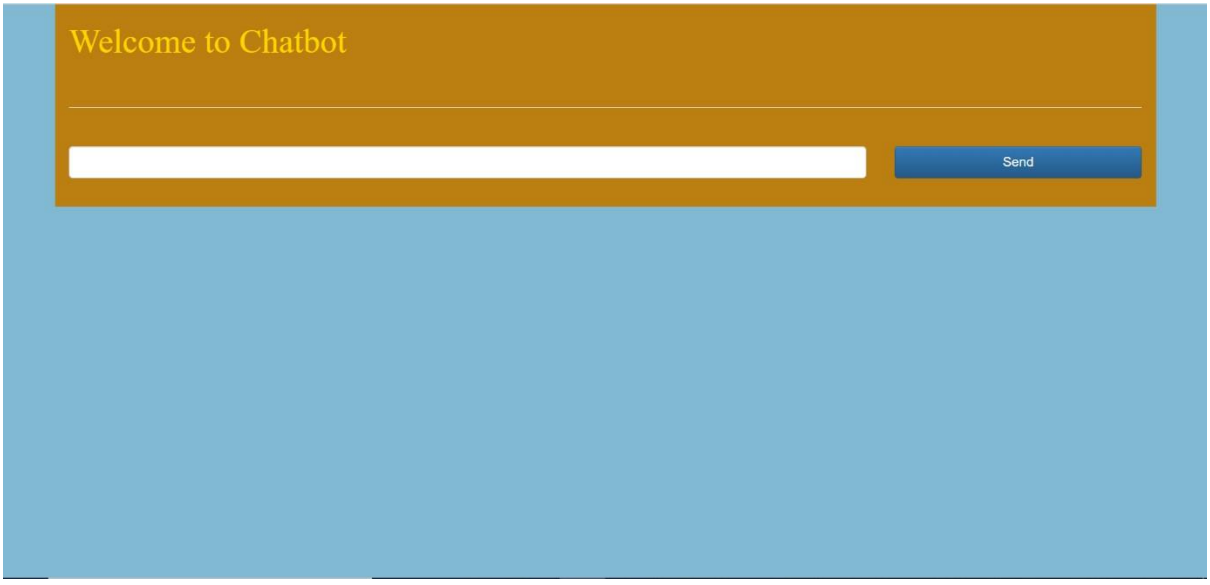
```json
{
  "intents": [
    {"tag": "noanswer",
      "patterns": ["",""],
      "responses": ["Sorry, can't understand you", "Please give me more info"]
      },
    {"tag": "greeting",
     "patterns": ["Hi there", "How are you", "Is anyone there?","Hey","Hola", "Hello", "Good day", "Hey", "Hi
     "responses": ["Hi stranger", "Yes, how can I help", "Ey watsupp"]
    },
    {"tag": "goodbye",
     "patterns": ["Bye", "See you later", "Goodbye", "Ok bye", "Bye Bye"],
     "responses": ["See you!", "Have a nice day", "Sure Bye"]
    },
    {"tag": "thanks",
     "patterns": ["Thanks", "Thank you", "That's helpful", "Awesome, thanks", "Thanks for helping me"],
     "responses": ["Happy to help!", "Any time!", "My pleasure", "You are welcome"]
    },
     {"tag": "name",
      "patterns": ["What is your name", "Whats your name", "Tell me your name", "Who are you", "Ungubani", "T
      "responses": ["I am Bot", "You can call be Bot"]
      },
      {"tag": "job",
        "patterns": ["What is your job here?", "What do you do?", "What can you help me with?", "Can i know w
        "responses": ["My work is to answer your queries related to health like about different diseases."]
       },
       {"tag": "covid",
        "patterns": ["What is covid?", "Tell me about covid", "covid", "i want to know about covid", "What do
        "responses": ["Doxycycline 100mg bd 5days ,Ivermectin 12mg od 3days, Vit b complex, Vit c 500mg tid,
       },
```

Index.html :

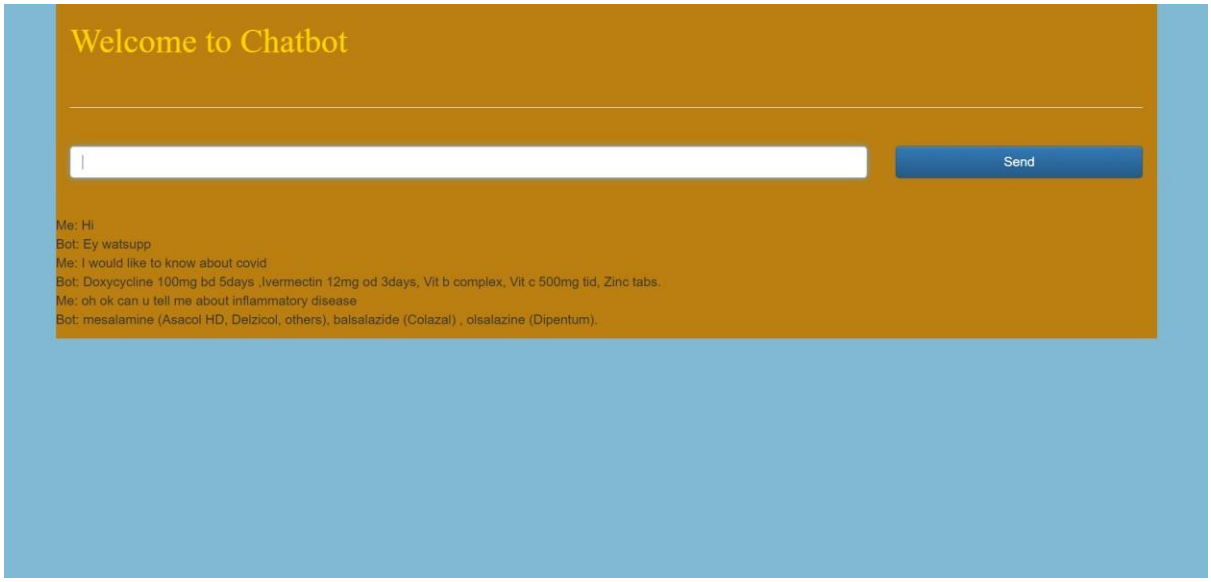
```
<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">

    <meta charset="utf-8">
    <meta http-equiv="x-ua-compatible" content="ie=edge">
    <meta name="description" content="">
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <title>Chatbot</title>

    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css" integ
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap-theme.min.css"

  </head>
  <body style="background-color: rgb(129, 185, 211);">
    <div class="container" style="background-color: rgba(187, 126, 14, 0.993);">
      <h1 style="font-family: Brush Script;color: gold;">Welcome to Chatbot</h1>
      <br>
      <hr>
      <br>

        <div class="row">
          <div class="col-lg-9">
            <input class="form-control" type="text" name="question" id="question">
          </div>
          <div class="col-lg-3">
            <button class="btn btn-primary btn-block" id="submit-button">Send</button>
          </div>
        </div>
        <br>
        <div class="row">
          <div class="col">
            <p id="response"></p>
          </div>
        </div>

    </div>

    <script src="/static/jquery.min.js" ></script>

    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js" integrity="sha384-Tc5IQ


    <script>


      jQuery(document).ready(function() {

        $("#submit-button").click(function(e) {
          e.preventDefault();


          $.ajax({
              type: "POST",
              url: "/chatbot",
              data: {
                  question: $("#question").val()
              },
```

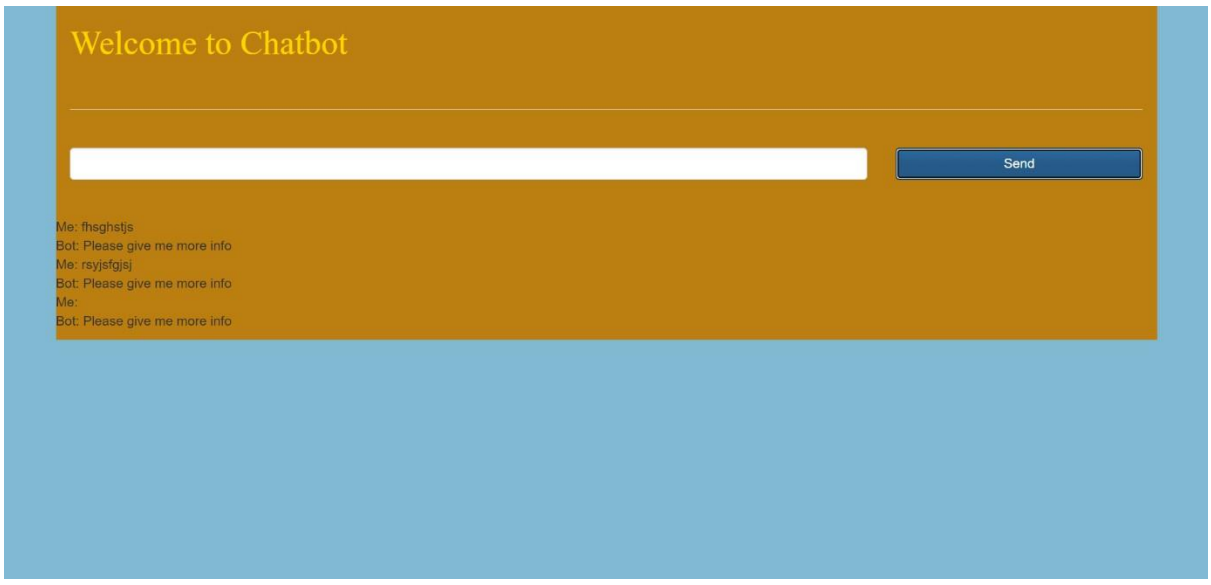
```
                question: $("#question").val()
            },
            success: function(result) {
              $("#response").append("<br>Me: "+$("#question").val()+ "<br> Bot: "+result.response);
              $("#question").val("")
            },
            error: function(result) {
                alert('error');
            }
        });


    });

  });

  </script>

</body>
</html>

```

End User View:




Welcome to Chatbot
Send
Me: Hi
Bot: Ey watsupp
Me: I would like to know about covid
Bot: Doxycycline 100mg bd 5days ,Ivermectin 12mg od 3days, Vit b complex, Vit c 500mg tid, Zinc tabs.


Welcome to Chatbot
Send
Me: Hi
Bot: Ey watsupp
Me: I would like to know about covid
Bot: Doxycycline 100mg bd 5days ,Ivermectin 12mg od 3days, Vit b complex, Vit c 500mg tid, Zinc tabs.
Me: oh ok can u tell me about inflammatory disease
Bot: mesalamine (Asacol HD, Delzicol, others), balsalazide (Colazal) , olsalazine (Dipentum).

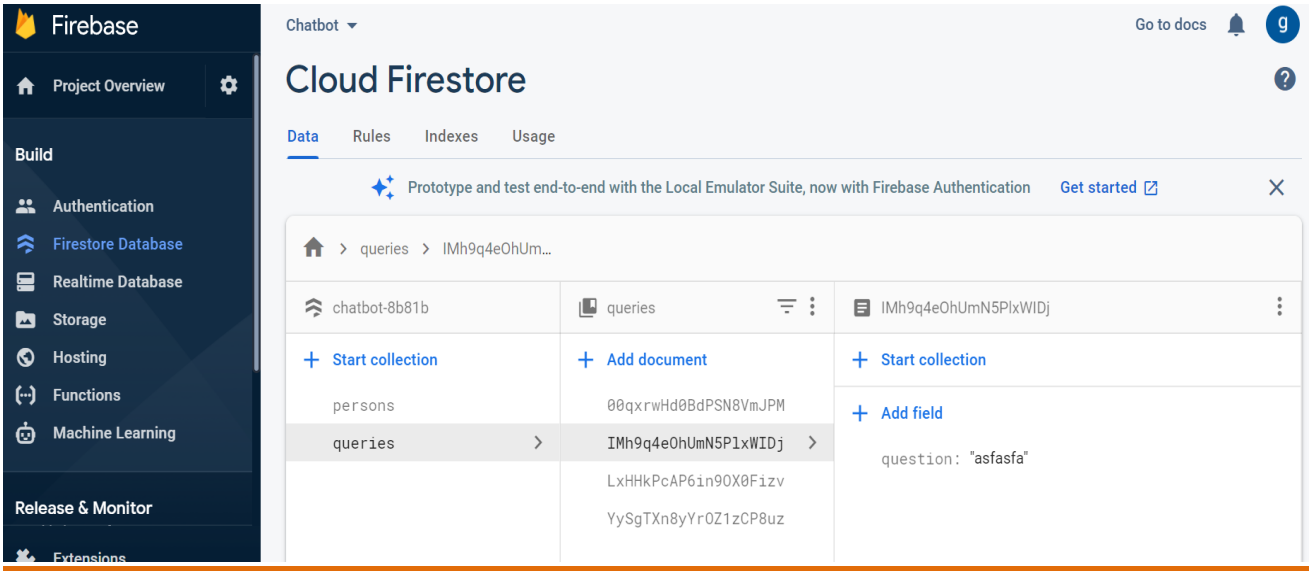

Welcome to Chatbot
Send
Me: fhsghstjs
Bot: Please give me more info
Me: rsyjsfgjsj
Bot: Please give me more info
Me:
Bot: Please give me more info

Firebase Database is connected:

The firebase database is connected to the chatbot. And only the queries which are unanswered or when the bot

is unable to predicts output then those user's queries will be stored into the database.

When the admin comes online, he/she will be viewing the queries and will be updating the data set.



Input Design:

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple.

The input is designed in such a way so that it provides security and ease of use with retaining the privacy.

Objectives :

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

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

3. When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are

provided as when needed so that the user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow.

Output Design :

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

Objectives :

1. Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.
2. Select methods for presenting information.
3. Create document, report, or other formats that contain information produced by the system.

Project Modules :

NLTK : The Natural Language Toolkit (NLTK) is a platform used for building Python programs that work with human language data for applying in statistical natural language processing (NLP). It contains text processing libraries for tokenization, parsing, classification, stemming, tagging and semantic reasoning.

Keras Sequential Model consisting of 3 layers is used.

Numpy is used for creating numpy arrays which will be sent as input to the model.

Random function is used to select a random response from the given intent's responses.

The tokenization and the lemmatizer functions are used for dividing the sentences into words and storing them in lists which are later converted into numpy arrays as numpy arrays are faster than python lists.

JSON module is used as the predefined data sets are present in a json file which has to be imported.

Conclusion:

The main objectives of the project were to develop an algorithm that will be used to identify answers related to user submitted questions. To develop a database were all the related data will be stored and to develop a web interface.

The web interface developed had two parts, one for simple users and one for the administrator.

A database was developed, which stores information about questions, answers, keywords.

Future Enhancements :

Instead of AIML based bot, other algorithms can be implemented. We can include voice-based queries. The users will have to give voice input and the system will take the input and extracts the text in the message and returns the required output in the form of text. And one more thing which could be introduced is the language translator, where people who know different languages can also access the chatbot for any queries. Also, after successful execution of chatbot in health domain, we can implement it in other domains like colleges, forensic, sports, etc. It will be beneficial in all the fields as without spending much time, we are accessing the relevant information and that too without any sorting.