Project

ReportOn

“CHATBOT”

Submitted By

Mr. Rahul Balu Barkade

MCA II, SEM III

Savitribai Phule Pune University

In partial fulfillment of the requirement of the award of degree of

Master in Computer Applications

Jayawantrao Sawant College of Engineering,

Hadapsar, Pune 28.

[2022-2023]

CERTIFICATE

This is to certify that **Mr. Rahul Balu Barkade** a bonafide student of, **Jayawantrao Sawant College of Engineering Hadapsar, Pune- 28**. Has successfully completed the project work as prescribed by the SavitribaiPhule Pune University in the partial fulfillment of the requirement of the master of computer application (MCA) program 2021-2022. The project work, titled as has been based on project work, which was carried out as **"CHATBOT"**

Prof. Anil Gaikwad — **Project Guide**

Prof. Swayam Shah — **Head of Department**

External Examiner:

Internal Examiner

Date:23/03/2023

Place: Pune

ACKNOWLEDGEMENT

I would take the opportunity to thank Principal **Dr. R. D. Kanphade**, and **JSCOE** for their support to me throughout the course.

I would like to thank **Prof. Anil Gaikwad**, Academic Coordinator, for scholarly disporsitkor, timely guidance, support and cooperation.

I would like to thank **Prof. Anil Gaikwad**, for her kind guidance, keen Interest, I (optimums encouragement and inspiration throughout the project work.

Finally, I grateful than all the staff members of JSCOE for their cooperation and support.

I also thankful to get constant encouragement, support and guidance from all Teaching and rot Teaching staff for their timely support which helped us in successfully completion of our project work.

Rahul Balu Barkade

DECLARATION

To,

The Principal,

Jayawantrao Sawant College Of

EngineeringHadapsar, Pune- 411028.

Respected Sir,

Mr. Rahul Balu Barkade Hereby declare that the project entitled **"CHATBOT".** developed and submitted under the guidance of **Prof. Anil Gaikwad** my original work.

The system Presented here is my own work and not has been duplicated from any other sources.

Date:23/03/2023

Place: Pune

You're sincerely

Rahul Balu Barkade

INDEX

1.1 Abstract:

A chatbot is an artificial intelligence computer program which performs communication using audio and video system. A person can ask any questions and chatbot will answer accordingly. Nowadays a chatbot is highly popular and takes speed as a computer communication application. Chatbot system is in trend, thus it is being used on many websites. With the chatbot, one doesn't have to wait to talk to the customer helpline, they don't even have to search for shopping through Websites. A chatbot is used in many areas like order food, product suggestions, customer support, weather, personal finance assistance, scheduled a meeting, search and track flights, send money, and many more

1.2)Introduction

A chat bot is automated software program that interacts with humans. A chat bot is merely computer program that fundamentally simulates humans conversion. Now days, Software Industry are using chatbots improve customer. Example, Eliza chatBot, proProfs chatBot, Mitsuku-pandora bot,botsify etc.

Chat has become the center of focus in this current era, thus the bots are being utilized to deliver information engagingly and conveniently. A chatbot is standout amongst the most progressive and promising tools of communication among people and machines. Famous chatbots like Whatsapp chatbot, EVA by HDFC and many more are in trend. These are very much helpful, but in this era of enhancing technology, day by day technology gets updated, and accordingly, user expectations also increase. A user wants more automation in the chatbot. Although every system is not perfect there is always a flaw in the system, so as in the chatbot there are some problems that the user has experienced while using a chatbot. Chatbot can be described as an answering system where a system will be able to answer questions or statements submitted by users.

1.3) Project Scope

This project can be easily implemented. We can add new features as and when we require. User request and response of chatbot is possible in this project. There is flexibility in all the modules. There are many features which could be added to this project for making this project more productive.

In this Chatbot create interface between user and machine. The usage chatbot is provide good knowledge of what should be done or not.

2.Literature Survey

Sr.no	Title of the paper	Year of Publication	What you studied from this paper /Seed Idea
1	Chatbot in python	2020(google scholar)	In this we can add voice assistant.
2	A Platform for Human-Chatbot Interaction Using Python	2016(IEEE)	request and response of user and chatbot
3	Ai Chatbots: Transforming the Digital World	2020(google scholar)	creation of chatbot and implementing real world
4	Bank chatBot using python	2017(IEEE)	In this, it allow financial institutions to talk to millions of customers.

3.Problem Statement

- chatbot just produced only one output instead of multiple outputs/results, the basic process flow is the same where each time an input is entered, the new search will be done.
- The quantity and quality of training data is critical to the performance.
- We want to update the list of the chatbot improve the interaction.
- Some time it can’t recognize the voice of user.
- Sometimes it can become difficult to work with the chatbot, and we have to set our expectations accordingly.

4.SYSTEM STUDY AND ANALYSIS:

1. Existing System:

There are many this type of systems available in the **Internet**. In this part of the report we will briefly describe the existing systems and undermine the features of the proposed system. These features are the improvements or advantages of proposed system over the existing systems.

Drawbacks of Existing System:

- Chatbot sound too mechanical in existing system.
- In existing system it responses the wrong words.
- Chatbot ban only handle basic questions.
- They are difficult to create earlier**.**
- They require constant modification.
- Use the way of their language.
- No recommendation or prediction.

4.2 Proposed System

In this chatbot is answering your question fluently. Chatbot will provide you the data.

In Proposed system chatbot is highly automated and is very easy to use, which will make interaction easy to understand. In Proposed system will also provide you all the data you want to know.

- ✓ Understandable information about the customer.
- ✓ The modern need is there for Bot Building for growth of Business to make progress.
- ✓ We can add our query in the chatbot.
- ✓ Also we suggest the what should be better to system.
- ✓ Bots, for the most part, operate on a network. Bots that can communicate with one another will use internet-based services like IRC.

4.3 Feasibility Study

An important outcome of preliminary investigation is the determination that the system requested is feasible. A feasibility study is carried out to select the best system that meets the performance requirements. A feasibility study is both necessary and prudent to evaluate the feasibility of the project at the earliest possible time.

Three key considerations involved in the feasibility analysis are.

- **ECONOMICAL FEASIBLITY**
- **OPERTINAL FEASIBLITY**
- **TECHNICAL FEASIBLITY**

ECONOMICAL FEASIBILITY

Economic feasibility is the cost and logistical outlook for this project. Thus, the developed system as well within the budget and this was achieved because most of the technologies used are freely available. The economical study analyses data to determine whether the cost ultimately profitable to the user. This project requires the historical data. So it is not a difficult task to any user to analyze the patterns or predicting the sales. Due to this. it is economically feasible.

OPERATIONAL FEASIBILITY :

Assessing operational feasibility is to gain an understanding of whether the proposed system is to solve the User problems, or take advantage of the opportunities or not. Is important to understand how the new systems will fetch into the current day-to-day operations of the organization. Operational feasibility studies are generally utilized to Process, Evaluation, Implementation, and Resistance. Python also enables developers to roll out programs and get prototypes running, making the development process much faster. Once a project is on its way to becoming an analytical tool or application, it can be ported to more sophisticated languages.

TECHNICAL FEASIBILITY :

Technical feasibility is one of the first studies that must be conducted after the project has been identified. Any system developed must not have a high demand on the available technical resources. This leads to high demands on the available technical resources. This lead to high demands being placed on the client. This application has been developed with python, where it provides more general approach to data science. It is a general purpose language with a readable syntax. A panda is the Python Data Analysis Library, used for everything from importing data from Excel spreadsheets to processing sets for time-series analysis.

5 PROJECT REQUIREMENTS

5.1 Hardware Specification:

Processor	:	Intel Core Duo and above
RAM	:	2.0 GB or More
Hard Disk	:	500 GB or more

5.2 Software Specification:

1. Operating System : Windows 7 or Above
2. Programming Language : Python
3. Tools : VS Code.
4. Front-End Technology : HTML ,CSS,JS

6 SYSTEM DESIGN:

Module list :

- Chatterbot
- Numpy
- Nltk
- Random
- Neural Network

1. Chatterbot :

ChatterBot is a Python library that makes it easy to generate automated responses to a user's input. ChatterBot uses a selection of machine learning algorithms to produce different types of responses. This makes it easy for developers to create chat bots and automate conversations with users.

Syntax : " pip install chatterbot".

2. Ppttsx3 :

pyttsx3 is a text-to-speech conversion library in Python. Unlike alternative libraries, it works offline, and is compatible with both Python 2 and 3.

Syntax : " pip install numpy

3. Numpy :

NumPy is a Python library used for working with arrays.

Syntax : " pip install numpy".

4. NTLK :

Natural Language Processing (NLP) is a process of manipulating or understanding the text or speech by any software or machine. An analogy is that humans interact and understand each other's views and respond with the appropriate answer. In NLP, this interaction, understanding, and response are made by a computer instead of a human.

5. Flask:

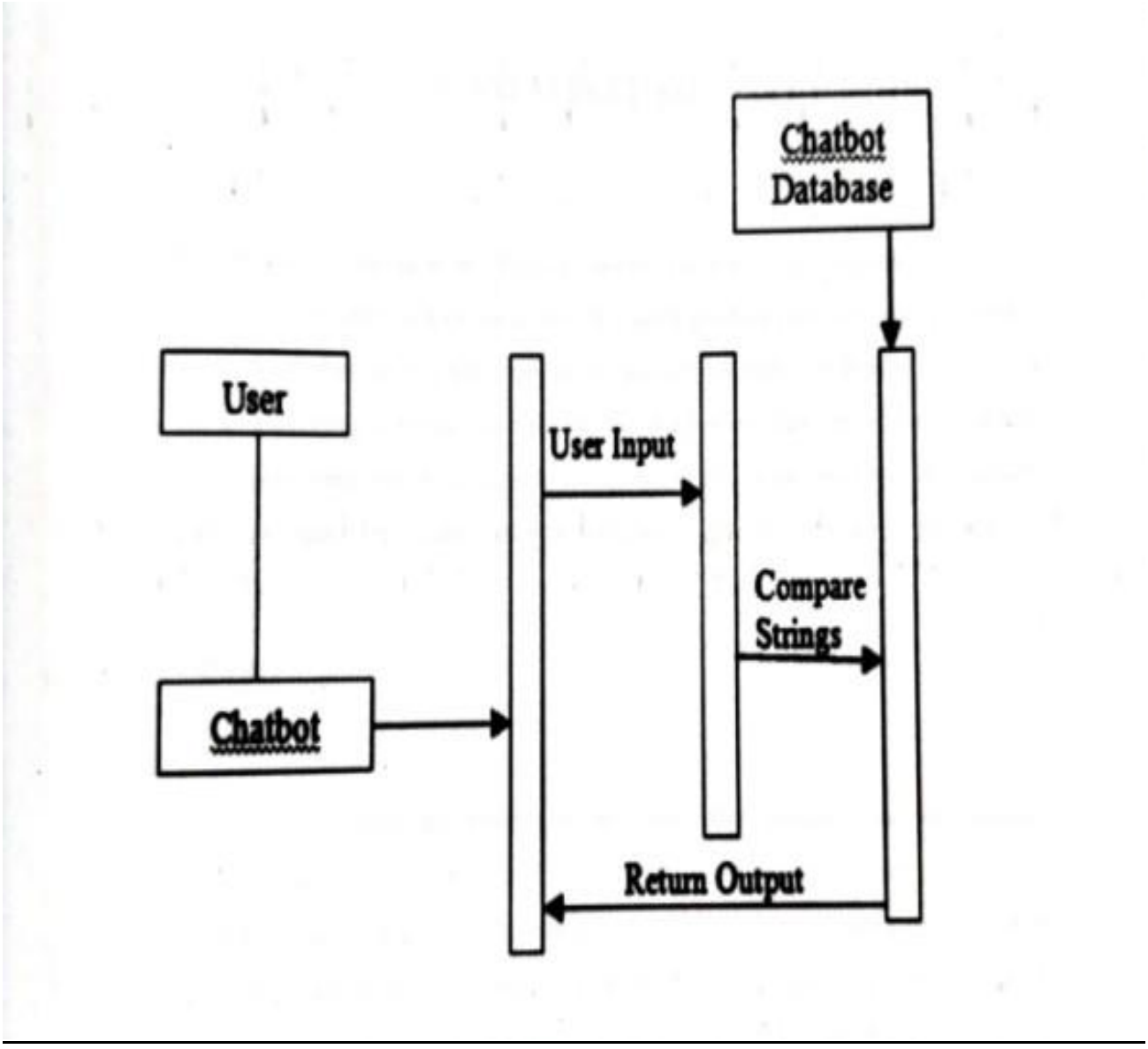
Flask provides configuration and conventions, with sensible defaults, to get started. This section of the documentation explains the different parts of the Flask framework and how they can be used, customized, and extended.

Syntax :

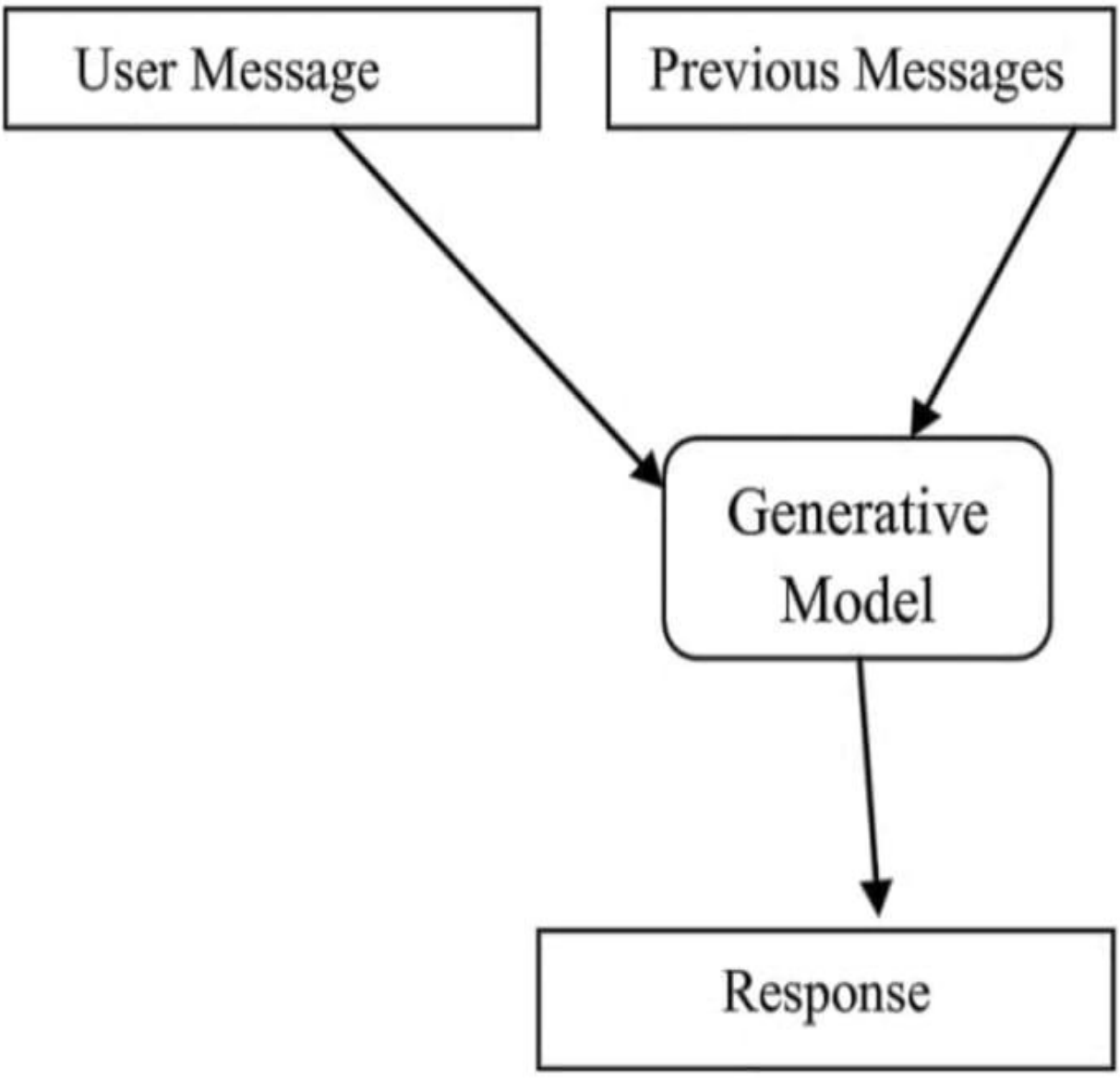
pip install flask

5.2 System Design

DATA FLOW DIAGRAM :



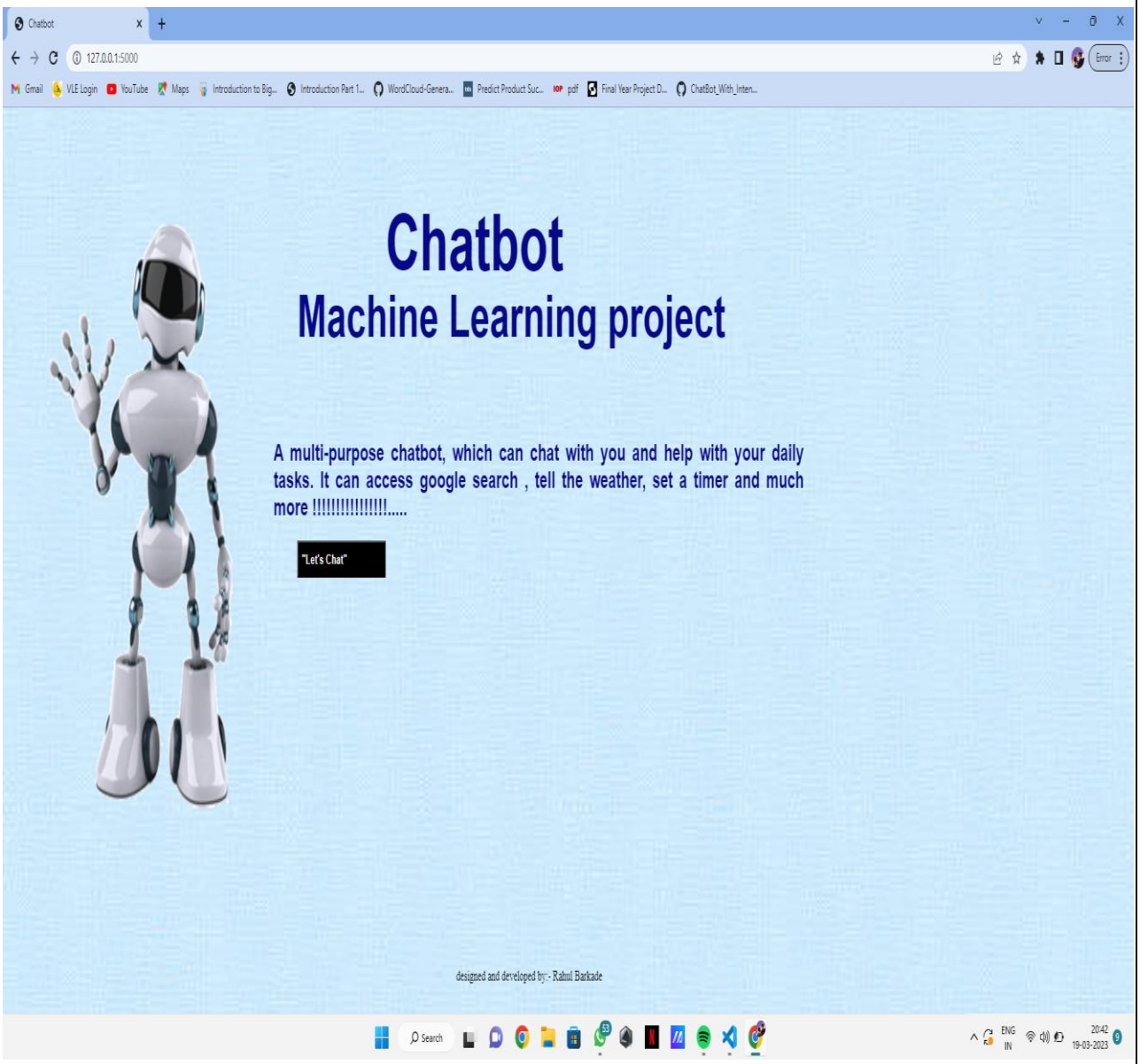
BLOCK DIAGRAM :

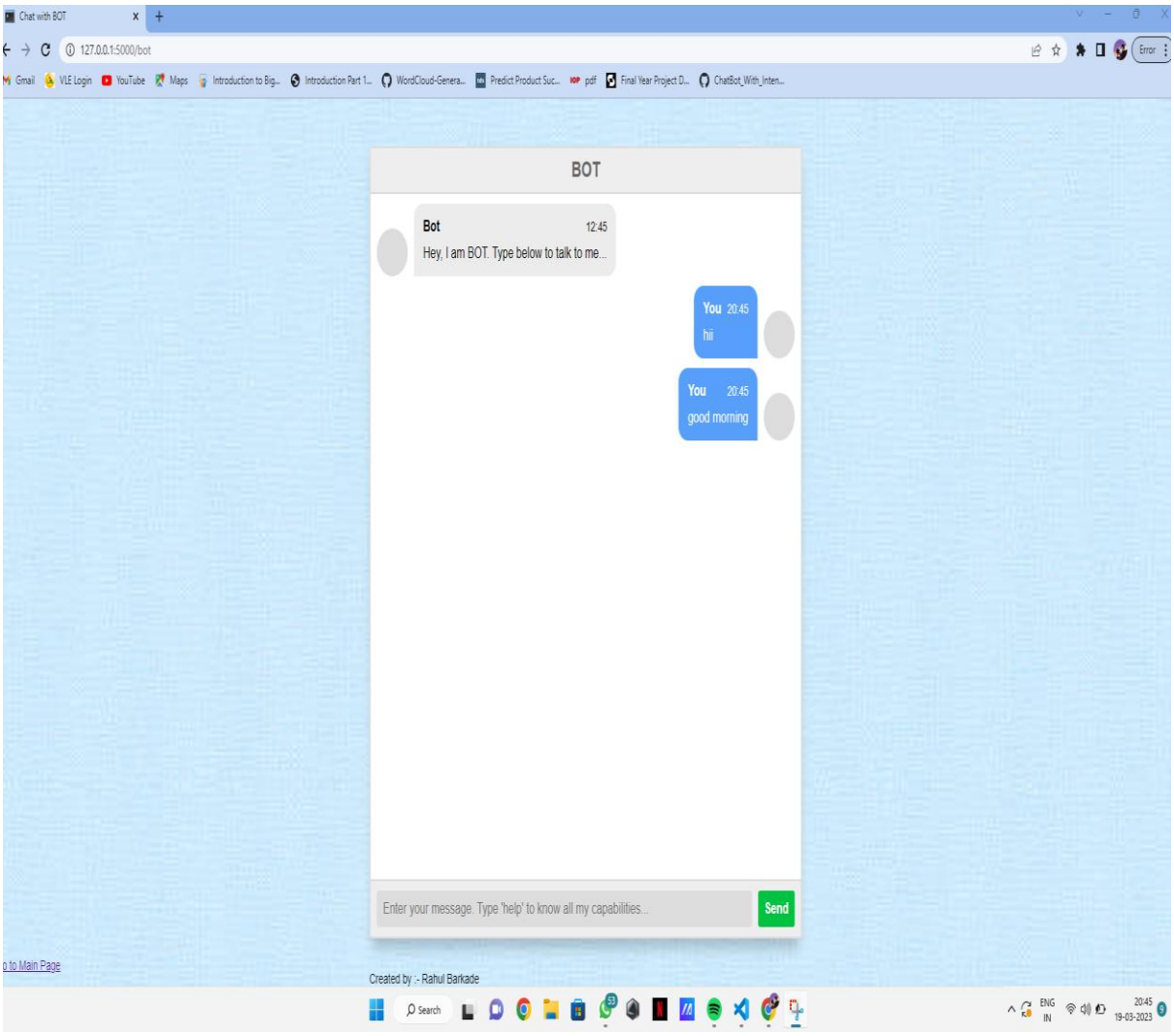


Activity Diagram :

6. SCREEN DESIGN :

1. Interface OF chatbot :




Chat with BOT
127.0.0.1:5000/bot
BOT
Bot 12:45
Hey, I am BOT. Type below to talk to me...
You 20:45
hii
You 20:45
good morning
Enter your message. Type 'help' to know all my capabilities...
Send
Created by :- Rahul Barkade

4 CODE REVIEW

6.1 bot _code.py

```
import keras
import nltk
import pickle
import json
import numpy as np
from keras.models import Sequential
from keras.layers import Dense,Dropout,Activation
import random
import datetime
#from googlesearch import *
import webbrowser
import requests
#from pycricbuzz import Cricbuzz
#import billboard
import time
#from pygame import mixer
#import COVID19Py
import nltk
nltk.download('punkt')

nltk.download('wordnet')

from nltk.stem import WordNetLemmatizer
lemmatizer=WordNetLemmatizer()
from keras.models import load_model
from flask import Flask
from flask import *




words=[]
classes=[]
documents=[]
ignore=['?','!',',',"'s"]

data_file=open('C:/Users/Shreyash/Pictures/Rahul_data/Chatbot-master/Chatbot-
master1/chatbot_codes/intents.json').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']))

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

words=[lemmatizer.lemmatize(word.lower()) for word in words if word not in
ignore]
words=sorted(list(set(words)))
classes=sorted(list(set(classes)))
pickle.dump(words,open('C:/Users/Shreyash/Pictures/Rahul_data/Chatbot-
master/Chatbot-master1/chatbot_codes/words.pkl','wb'))
pickle.dump(classes,open('C:/Users/Shreyash/Pictures/Rahul_data/Chatbot-
master/Chatbot-master1/chatbot_codes/classes.pkl','wb'))

#training data
training=[]
output_empty=[0]*len(classes)

for doc in documents:
    bag=[]
    pattern=doc[0]
    pattern=[ lemmatizer.lemmatize(word.lower()) for word in pattern ]

    for word in words:
        if word in pattern:
            bag.append(1)
        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)
X_train=list(training[:,0])
y_train=list(training[:,1])

#Model
model=Sequential()
model.add(Dense(128,activation='relu',input_shape=(len(X_train[0]),)))
model.add(Dropout(0.5))
model.add(Dense(64,activation='relu'))
model.add(Dense(64,activation='relu'))
```

```
model.add(Dropout(0.5))
model.add(Dense(len(y_train[0]),activation='softmax'))

adam=keras.optimizers.Adam(0.001)
model.compile(optimizer=adam,loss='categorical_crossentropy',metrics=['accurac
y'])
#model.fit(np.array(X_train),np.array(y_train),epochs=200,batch_size=10,verbos
e=1)
weights=model.fit(np.array(X_train),np.array(y_train),epochs=200,batch_size=10
,verbose=1)
model.save('C:/Users/Shreyash/Downloads/Chatbot-master/Chatbot-
master1/chatbot_codes/mymodel.h5',weights)


model = load_model('C:/Users/Shreyash/Pictures/Rahul_data/Chatbot-
master/Chatbot-master1/chatbot_codes/mymodel.h5')
intents = json.loads(open('C:/Users/Shreyash/Pictures/Rahul_data/Chatbot-
master/Chatbot-master1/chatbot_codes/intents.json').read())
words = pickle.load(open('C:/Users/Shreyash/Pictures/Rahul_data/Chatbot-
master/Chatbot-master1/chatbot_codes/words.pkl','rb'))
classes = pickle.load(open('C:/Users/Shreyash/Pictures/Rahul_data/Chatbot-
master/Chatbot-master1/chatbot_codes/classes.pkl','rb'))

app= Flask(__name__)
@app.route("/")
def home():
    #C:\Users\Shreyash\Pictures\Rahul_data\templates\main.html

    return render_template("main.html")
#Predict
@app.route("/bot",methods = ['POST'])
def bot():
    #query=request.args.get("user_resp")

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

    def create_bow(sentence,words):
        sentence_words=clean_up(sentence)
        bag=list(np.zeros(len(words)))

        for s in sentence_words:
            for i,w in enumerate(words):
                if w == s:
                    bag[i] = 1
        return np.array(bag)
```

```
    def predict_class(sentence,model):
        p=create_bow(sentence,words)
        res=model.predict(np.array([p]))[0]
        threshold=0.8
        results=[[i,r] for i,r in enumerate(res) if r>threshold]
        results.sort(key=lambda x: x[1],reverse=True)
        return_list=[]

        for result in results:

return_list.append({'intent':classes[result[0]],'prob':str(result[1])})
        return return_list

    def get_response(return_list,intents_json):

        if len(return_list)==0:
            tag='noanswer'
        else:
            tag=return_list[0]['intent']
        if tag=='datetime':
            print(time.strftime("%A"))
            print (time.strftime("%d %B %Y"))
            print (time.strftime("%H:%M:%S"))

        if tag=='google':
            query=input('Enter query...')
            chrome_path = r'C:\Program Files
(x86)\Google\Chrome\Application\chrome.exe %s'
            for url in search(query, tld="co.in", num=1, stop = 1, pause = 2):
                webbrowser.open("https://google.com/search?q=%s" % query)
        if tag=='weather':
            api_key='987f44e8c16780be8c85e25a409ed07b'
            base_url = "http://api.openweathermap.org/data/2.5/weather?"
            city_name = input("Enter city name : ")
            complete_url = base_url + "appid=" + api_key + "&q=" + city_name
            response = requests.get(complete_url)
            x=response.json()
            print('Present temp.: ',round(x['main']['temp']-273,2),'celcius ')
            print('Feels Like:: ',round(x['main']['feels_like']-
273,2),'celcius ')
            print(x['weather'][0]['main'])

        if tag=='news':
            main_url = " http://newsapi.org/v2/top-
headlines?country=in&apiKey=bc88c2e1ddd440d1be2cb0788d027ae2"
            open_news_page = requests.get(main_url).json()
            article = open_news_page["articles"]
```

```
            results = []

            for ar in article:
                results.append([ar["title"],ar["url"]])

            for i in range(10):
                print(i + 1, results[i][0])
                print(results[i][1],'\n')

        if tag=='cricket':
            c = Cricbuzz()
            matches = c.matches()
            for match in matches:
                print(match['srs'],' ',match['mnum'],' ',match['status'])

        if tag=='song':
            chart=billboard.ChartData('hot-100')
            print('The top 10 songs at the moment are:')
            for i in range(10):
                song=chart[i]
                print(song.title,'- ',song.artist)
        if tag=='timer':
            mixer.init()
            x=input('Minutes to timer..')
            time.sleep(float(x)*60)
            mixer.music.load('Handbell-ringing-sound-effect.mp3')
            mixer.music.play()

        if tag=='covid19':

            covid19=COVID19Py.COVID19(data_source='jhu')
            country=input('Enter Location...')

            if country.lower()=='world':
                latest_world=covid19.getLatest()
                print('Confirmed:',latest_world['confirmed'],'
Deaths:',latest_world['deaths'])

            else:

                latest=covid19.getLocations()

                latest_conf=[]
                latest_deaths=[]
                for i in range(len(latest)):

                    if latest[i]['country'].lower()== country.lower():
                        latest_conf.append(latest[i]['latest']['confirmed'])
```

```
                        latest_deaths.append(latest[i]['latest']['deaths'])
                latest_conf=np.array(latest_conf)
                latest_deaths=np.array(latest_deaths)
                print('Confirmed: ',np.sum(latest_conf),'Deaths:
',np.sum(latest_deaths))

        list_of_intents= intents_json['intents']
        for i in list_of_intents:
            if tag==i['tag'] :
                result= random.choice(i['responses'])
        return result

    def response(text):
        return_list=predict_class(text,model)
        response=get_response(return_list,intents)
        return response

    while(1):
        x=input("Enter your message here")
        #x="hi good morning"
        output=response(x)
        output= str(output)
        # if x.lower() in ['bye','goodbye','get lost','see you']:
        #     break
        return render_template("index.html")
if __name__ == '__main__':
   #app.run()
   app.run(host='127.0.0.1', port=5000, debug=True)

# #Self learning
# print('Help me Learn?')
# tag=input('Please enter general category of your question  ')
# flag=-1
# for i in range(len(intents['intents'])):
#     if tag.lower() in intents['intents'][i]['tag']:
#         intents['intents'][i]['patterns'].append(input('Enter your message:
'))
#         intents['intents'][i]['responses'].append(input('Enter expected
reply: '))
#         flag=1

# if flag==-1:

#     intents['intents'].append (
#         {'tag':tag,
#          'patterns': [input('Please enter your message')],
#          'responses': [input('Enter expected reply')]})
```

```
# with open('C:/Users/Shreyash/Pictures/Rahul_data/Chatbot-master/Chatbot-
master/chatbot_codes/intents.json','w') as outfile:
#     outfile.write(json.dumps(intents,indent=4))
```

6.2 Main.html

```
<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <title>Chatbot</title>
<link rel="icon" href="https://image.flaticon.com/icons/svg/2597/2597330.svg">
<style media="screen">

/*body{
  background-image:
url(https://s3.envato.com/files/229501069/Preview_Image_Di.jpg);
  background-repeat: no-repeat;
  background-size: cover;
  background-blend-mode: lighten;
}*/
body{
  background-image: url(https://media.istockphoto.com/photos/light-blue-
background-with-pattern-picture-
id624409380?k=6&m=624409380&s=612x612&w=0&h=XYHFBLtqTp3EMuKDtJMzTW5wTzkFbalhfA
rb14U6mWI=);
  background-repeat: no-repeat;
  background-size: cover;
}
#chat{
   position: fixed;
   left: 815px;
   font-size: 18px;
   bottom: 150px;
   color: white;
   background-color: black;
   height: 50px;
   width: 140px;
   border-radius: 12px;
}
h1{
  position: fixed;
  left: 650px;
  top: 40px;
  font-size: 80px;
  color: #08088A;
  font-family:sans-serif  ;
```

```
}

h2{
  position: fixed;
  left: 500px;
  top: 130px;
  font-size: 60px;
  color: #08088A;
  font-family:sans-serif;
}

form button {
  position: fixed;
  left: 500px;
  width: 150px;
  height: 38px;
  margin-top: 20px;
  border-color: black;
  background-color:  black;
  font-family: sans-serif;
  font-weight: bold;
  text-align: justify;
  color: white;
}
.boxed{

  width: 900px;
  height: 300px;
  position: fixed;
  left: 460px;
  top: 340px;
font-size: 24px;
color: #08088A;
font-family: sans-serif;
font-weight: bold;
text-align: justify;

}

</style>
  </head>
  <body>

    <br><br><br>

  <!--<img src="https://media.istockphoto.com/photos/light-blue-background-
with-pattern-picture-
id624409380?k=6&m=624409380&s=612x612&w=0&h=XYHFBLtqTp3EMuKDtJMzTW5wTzkFbalhfA
```

```
rb14U6mWI=" style=" height: 770px; width: 1550px; opacity: 0.7; position:
fixed; left: 0px; top: 0px;"> -->
    <img src="https://i.gifer.com/3q63.gif" height="600" width="320"
style="position: fixed; left: 70px;


top: 120px;">

    <h1 style=" f"><b>Chatbot</b></h1>
    <h2><b>Machine Learning project</b></h2>


    <div class="boxed">
  A multi-purpose chatbot, which can chat with you and help with your daily
tasks. It can access google search , tell the weather, set a timer and much
more !!!!!!!!!!!!!!!.....

  <form action="{{ url_for('bot')}}" method="post">
    <button type="submit">
    "Let's Chat"
  </button>
   </form>
</div>


<div>
<p style="position: fixed;bottom: 0;left: 770px;font-size: 14px; color:
black;">designed and developed by:- Rahul
Barkade</a><br>          &nb
sp;   </p>
</div>
  </body>
</html>
```

6.3 Index.html

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <title>Chat with BOT</title>
  <link rel="icon"
href="https://image.flaticon.com/icons/svg/2597/2597330.svg">
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <link rel="stylesheet" href="{{ url_for('static',
filename='styles/style.css') }}">
  <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></scrip
t>
    <style>
    :root {
      --body-bg: linear-gradient(135deg, #f5f7fa 0%, #c3cfe2 100%);
      --msger-bg: #fff;
      --border: 2px solid #ddd;
      --left-msg-bg: #ececec;
      --right-msg-bg: #579ffb;
    }

    html {
      box-sizing: border-box;
    }

    *,
    *:before,
    *:after {
      margin: 0;
      padding: 0;
      box-sizing: inherit;
    }

    body {
    background-image: url(https://media.istockphoto.com/photos/light-blue-
background-with-pattern-picture-
id624409380?k=6&m=624409380&s=612x612&w=0&h=XYHFBLtqTp3EMuKDtJMzTW5wTzkFbalhfA
rb14U6mWI=);
    background-repeat: no-repeat;
```

```
  background-size: cover;
  backface-visibility: hidden;
    display:flex;
    justify-content: center;
    align-items: center;
    height: 100vh;

    font-family: Helvetica, sans-serif;
  }

  .msger {
    display: flex;
    flex-flow: column wrap;
    justify-content: space-between;
    width: 100%;
    max-width: 700px;
    margin: 25px 10px;
    height: calc(100% - 100px);
    border: var(--border);
    border-radius: 5px;
    background: var(--msger-bg);
    box-shadow: 0 15px 15px -5px rgba(0, 0, 0, 0.2);

  }

  .msger-header {
    /* display: flex; */
    font-size: medium;
    justify-content: space-between;
    padding: 10px;
    text-align: center;
    border-bottom: var(--border);
    background: #eee;
    color: #666;
  }

  .msger-chat {
    flex: 1;
    overflow-y: auto;
    padding: 10px;
  }
  .msger-chat::-webkit-scrollbar {
    width: 6px;
  }
  .msger-chat::-webkit-scrollbar-track {
    background: #ddd;
  }
  .msger-chat::-webkit-scrollbar-thumb {
```

```
    background: #bdbdbd;
  }
  .msg {
    display: flex;
    align-items: flex-end;
    margin-bottom: 10px;
  }

  .msg-img {
    width: 50px;
    height: 50px;
    margin-right: 10px;
    background: #ddd;
    background-repeat: no-repeat;
    background-position: center;
    background-size: cover;
    border-radius: 50%;
  }
  .msg-bubble {
    max-width: 450px;
    padding: 15px;
    border-radius: 15px;
    background: var(--left-msg-bg);
  }
  .msg-info {
    display: flex;
    justify-content: space-between;
    align-items: center;
    margin-bottom: 10px;
  }
  .msg-info-name {
    margin-right: 10px;
    font-weight: bold;
  }
  .msg-info-time {
    font-size: 0.85em;
  }

  .left-msg .msg-bubble {
    border-bottom-left-radius: 0;
  }

  .right-msg {
    flex-direction: row-reverse;
  }
  .right-msg .msg-bubble {
    background: var(--right-msg-bg);
    color: #fff;
```

```
      border-bottom-right-radius: 0;
    }
    .right-msg .msg-img {
      margin: 0 0 0 10px;
    }

    .msger-inputarea {
      display: flex;
      padding: 10px;
      border-top: var(--border);
      background: #eee;
    }
    .msger-inputarea * {
      padding: 10px;
      border: none;
      border-radius: 3px;
      font-size: 1em;
    }
    .msger-input {
      flex: 1;
      background: #ddd;
    }
    .msger-send-btn {
      margin-left: 10px;
      background: rgb(0, 196, 65);
      color: #fff;
      font-weight: bold;
      cursor: pointer;
      transition: background 0.23s;
    }
    .msger-send-btn:hover {
      background: rgb(0, 180, 50);
    }

    .msger-chat {
      background-color: white;
      /*background-image:
url("https://th.bing.com/th/id/OIP.igHaNytA8mbd1lbiEeJ8MAHaEK?w=296&h=166&c=7&
o=5&dpr=1.25&pid=1.7");*/
    background-repeat: no-repeat;
    background-size: cover;
    }
</style>
</head>

<body>


  <!-- partial:index.partial.html -->
```

```
  <section class="msger">
    <header class="msger-header">
      <div class="msger-header-title">
        <i></i> <p style="font-size:22px;"><b>BOT </b><p><i></i>
      </div>
    </header>

    <main class="msger-chat">
      <div class="msg left-msg">
        <div class="msg-img" style="background-image:
url(https://image.flaticon.com/icons/svg/2597/2597330.svg)"></div>

        <div class="msg-bubble">
          <div class="msg-info">
            <div class="msg-info-name">Bot</div>
            <div class="msg-info-time">12:45</div>
          </div>

          <div class="msg-text">
            Hey, I am BOT. Type below to talk to me...
          </div>
        </div>
      </div>

    </main>

    <form class="msger-inputarea" action="" method="post">

    <input type="text" class="msger-input" id="textInput" placeholder="Enter
your message. Type 'help' to know all my capabilities..." >


       <button type="submit" class="msger-send-btn">Send</button>

    </form>
  </section>



  <!-- partial -->
  <script
src='https://use.fontawesome.com/releases/v5.0.13/js/all.js'></script>
  <script>

  function play() {
  var audio = new Audio('Handbell-ringing-sound-effect.mp3');
  audio.play();
  }
```

```
    const msgerForm = get(".msger-inputarea");
    const msgerInput = get(".msger-input");
    const msgerChat = get(".msger-chat");


    // Icons made by Freepik from www.flaticon.com
    const BOT_IMG = "https://image.flaticon.com/icons/svg/2597/2597330.svg";
    const PERSON_IMG =
"https://image.flaticon.com/icons/svg/3048/3048122.svg";
    const BOT_NAME = "bot";
    const PERSON_NAME = "You";

    msgerForm.addEventListener("submit", event => {
      event.preventDefault();

      const msgText = msgerInput.value;
    temp=msgText.split(':')
      if((temp[0].toLowerCase()).trim()=='google')
      {search='https://www.google.com/search?q=' + temp[1].trim()
      window.open(search)}
      if (!msgText) return;
      if((temp[0].toLowerCase()).trim()=='set a timer')
      {
      play()}


      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-bubble">
    <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) {

      // Bot Response
      $.get("/bot", { msg: rawText }).done(function (data) {
        console.log(rawText);
        console.log(data);
        msgText = data;
        msgText=msgText.split("\n")
        var i;
        for(i=0;i<msgText.length;i++)
        {
          appendMessage(BOT_NAME, BOT_IMG, "left", msgText[i]);
      }
      });

    }


    // 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>
  <p style="position: fixed;bottom: 0;left: 5px; bottom: 14px;font-size:
14px;"><a href="/">Go to Main Page</a></p>

  <p style="position: fixed;bottom: 0;left: 610px;font-size: 14px;">Created by
:- Rahul Barkade </p>

</body>

</html>
```

7 Advantages

- Provide Understandable information about the queries.
- Time and Cost saving.
- Chatbots have 24/7 working Availability.
- Understand the Customer behaviour.
- Simple and Easy UI for user.
- Instant answers to queries.
- Chatbot provide long time saving.
- They are used for increased the sales of many companies.
- They are used to customer satisfaction.
- They are used for easy request and response.

8 Conclusion: -

- As per the study, different types of algorithms and techniques are used for creating the chatbot.
- In this project, we have introduced a chatbot that is able to interact with users.
- This chatbot can answer queries in the textual user input.
- And also by voice we can ask queries to chatbot.

9 Reference

O https://www.javatpoint.com/java-tutorial

- https://chatterbot.readthedocs.io/
- https://www.geeksforgeeks.org/chat-bot-in-python-with-chatterbot-module/
- https://pypi.org/project/ChatterBot
- Anjana Tiha, "Intelligent Chatbot using Deep Learning", University of Memphis Spring 2018, 04/26/2018.
- https://scholar.google.com/scholar?hl=en&as_sdt=0%2C5&q=ai+chatbot+using+python&btnG=
- Bayan Abu Shawar and Eric Atwell, 2007 "Chatbots: Are they Really Useful?"
- LDV Forum - GLDV Journal for Computational Linguistics and Language Technology.