1. INTRODUCTION

A chatbot is an instant messaging account that able to provide services using instant messaging frameworks with the aim of providing conversational services to users in an efficient and friendly manner. In technological view it is a set of queries, that contains predefined questions and answers leveraging Natural Language Processing (NLP). It is a python library which generates automated user responses. Chatbot is trained with a data set, from which it predicts the properties of the new data. At first it receives the user request, analyses the request and complies the response. Chatbots are different from other applications, because they do not have last seen, status, posts.

Main purpose of this chatbot is to reduce man power and to satisfy the frequently asked queries by the user. Chatbots can be attached to the web browser with different operating systems and also be used as an android application. Recent days, chatbots use machine learning algorithm to analyze and respond to the user. This analysis helps the bot to respond to each user query in a perfect manner. Sometimes, the response will not be so accurate for certain queries. This issue can be solved by moderate dataset, which can be trained to the bot. This paper primarily focuses on the development of user friendly chatbot using python. This has a practical deployment in SCSVMV website.

* 1. PROBLEM STATEMENT

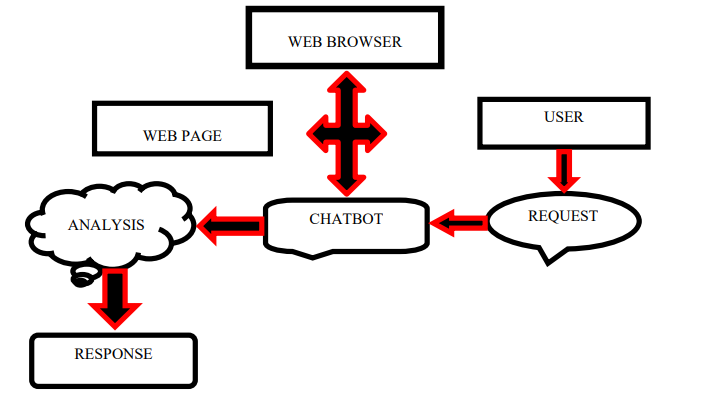
Chatbot to perform operations like greet the user, tell the current date time and day, predict the weather forecast of the city selected of users choice, text to speech operation.

* 1. OBJECTIVES
* Develop a basic chatbot that can interact with users and respond to their queries.
* Implement greetings functionality to greet the user with random responses.
* Provide the chatbot with a name and allow users to inquire about its name.
* Enable the chatbot to provide the current date and time upon user request.
* Allow the chatbot to inform the user about the current day of the week.
* Handle user expressions of gratitude and respond with appropriate acknowledgments.
* Incorporate a well-being check to allow users to ask how the chatbot is doing and respond positively.
* Integrate weather forecasting functionality to provide weather information for a specified city.
* Implement a text-to-speech feature using the pyttsx3 library to enable the chatbot to speak.
* Allow users to engage in a conversation with the chatbot, listening to its spoken responses.
* Provide the chatbot with the ability to generate random responses for different types of user inputs.
* Handle unrecognized user inputs and prompt the user to try again.

1. METHODOLOGY

The main purpose of this chatbot is to respond to user queries without man power. User can work with the chatbot in any web browser. The chatbot receives the request sent by the user, analyses it and responds to the user in return. This analysis is done, Figure 2: General Architecture of the Propose System using the machine learning algorithm. The queries are predefined with a particular tag for each set. This tag is nothing but a keyword that helps the chatbot to analyze the user request. After the analysis, the chatbot responds to the user with required reply. If the user request is not clear to the chatbot, the response will be a default message defined by the developer. Almost all the queries from the user will be clearly responded, only few are exceptional cases.

The overview of the general chatbot operations is as shown in Figure 1.1.



**FIGURE 1.1. OVERVIEW OF THE CHATBOT OPERATION.**

2.1 MODULES USED IN THIS PROJECT:

# a)**Python - Random Module:**

The random module is a built-in module to generate the pseudo-random variables. It can be used perform some action randomly such as to get a random number, selecting a random elements from a list, shuffle elements randomly, etc.

Syntax: import random

b)Python-Date time module:

In Pyhton, date and time are not a data type of their own,but a module named datetime can be imported to work with the date as well as time.Python Datetime module comes built into python,so there is no need to install it externally.

Syntax: import datetime

Print(datetime.datetime.now())

c) . Python -requests module:

Requests library is one of the integral part for making HTTP requests to a specified URL. When one makes request to a URL, it returns a response. Python requests provides inbuilt functionalities for managing both the request and the response.

Syntax:

import requests

requests.ger(url,params={key:value},args)

d) Pyhton pyttsx module:

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. An application invokes the pyttsx3.init() factory function to get a reference to a pyttsx3. Engine instance. it is a very easy to use tool which converts the entered text into speech. The pyttsx3 module supports two voices first is female and the second is male which is provided by “sapi5” for windows. It supports three TTS engines.

Syntax: import pyttsx3

All these above modules have to be installed in the command prompt of our system and then use it in our program.

3. IMPLEMENTATION

3.1 SOURCE CODE

import random

import datetime

import requests

import pyttsx3

hello=["hi","you there?","hello",]

what=["what can you do?","what are your features?"]

bye=["bye","i am going","nice talking to you"]

name=["what is your name?","do you have any name?","how to call you?"]

date=["what is todays date?","date?","what is date today?"]

time=["what time is it now?","exact time now?","time?","tell me time now"]

day=["what day is it today?","which day is today?","tell me todays date","day?"]

thanks=["thanks","thank you","ok thank you"]

fine=["how are you?","how are you doing?"]

ans=["i am fine","fine","good","i am good","i am also fine","all good"]

weather=["weather forecast","weather info","weather report"]

talk=["can you talk?","talk"]

print("\*\*\*\*\*\*\*\*CHAT BOT WELCOMES YOU\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \n")

print("BOT:: how may i help you? \n")

while (1):

a=input('you :')

if a.lower() in hello:

botans=["hello boss!","namaste :)","chat bot welcomes you :)","greetings!!"]

print('BOT:: '+random.choice(botans)+'\n')

elif a.lower() in bye:

botans=["see you soon!","byee take care","come back soon","nice talking to you :)"]

print('BOT:: '+random.choice(botans)+'\n')

print("\n")

break

elif a.lower() in name:

botans=["you can call me joker","my name is joker","i am known as joker"]

print('BOT:: '+random.choice(botans)+'\n')

elif a.lower() in date:

print("BOT:: ")

print( datetime.datetime.now())

elif a.lower()in time:

print("BOT:: ")

print(datetime.datetime.now())

elif a.lower()in day:

print("BOT:: today is ")

x = datetime.datetime.now()

print(x.strftime("%A")+"\n")

elif a.lower() in thanks:

botans=["you are welcome!","no problem anytime :)","my pleasure :)"]

print('BOT:: '+random.choice(botans)+'\n')

elif a.lower() in fine:

botans=["i am fine :) ","i am always happy!!"]

print('BOT:: '+random.choice(botans)+'\n')

print("BOT:: how are you? ")

elif a.lower() in ans:

botans=[" thats awesome ! how can i help you??","great! how can i help you?"]

print('BOT:: '+random.choice(botans)+'\n')

elif a.lower() in weather:

city\_name = input("BOT:: Enter the name of the City : ")

print("BOT:: sure! hold on \n")

def Gen\_report(C):

url = 'https://wttr.in/{}'.format(C)

try:

data = requests.get(url)

T = data.text

except:

T = "Error Occurred"

print(T)

Gen\_report(city\_name)

elif a.lower()in what:

botans=["i can tell date, time, day. You can also see weather forecast of any city .I also have special feature of text to speech"]

print('BOT:: '+random.choice(botans)+'\n')

elif a.lower() in talk:

bot=pyttsx3.init()

voices = bot.getProperty('voices')

bot.setProperty('voice', voices[1].id)

bot.say("now i am learning to speak...what do you want me to say?")

bot.runAndWait()

answer=input("what do you want me to say?::")

bot.say(answer)

bot.runAndWait()

while(1):

bot.say("do you want to hear more from me? press 1 for yes else 0 for no")

bot.runAndWait()

yes=int(input("BOT:: press 1 for yes else 0 for no:: "))

if yes==1:

answer=input("what else you want me to say?::")

bot.say(answer)

bot.runAndWait()

elif yes==0:

print("It was nice talking to you!")

bot.say("It was nice talking to you!")

bot.runAndWait()

print("choose different task..")

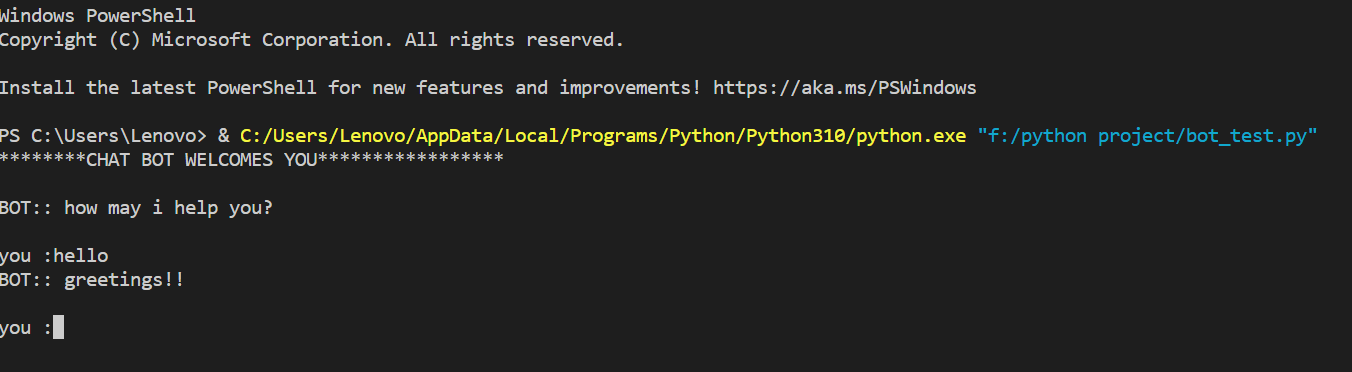
break

else:

print('sorry! i didnt get you try again :( '+'\n')

4. OUTPUTS

Screenshots of the interface are shown below in figures:



**FIGURE 5.1. PROGRAM STARTS AND BOT GREETS THE USER.**

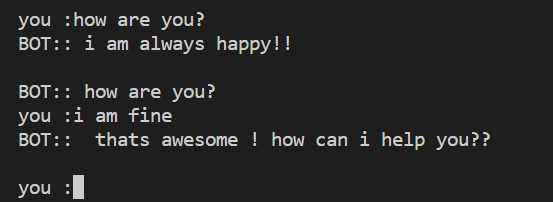
****

FIGURE 5.2. FORMAL INTERACTION WITH THE BOT.

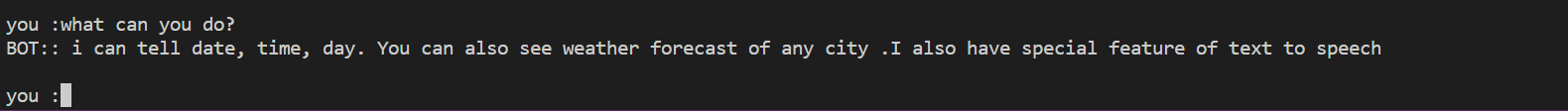


FIGURE 5.3. DIFFERENT TASKS BOT CAN PERFORM.

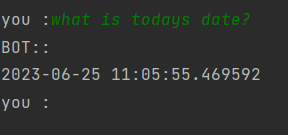


FIGURE 5.4. SHOWING CURRENT DATE AND TIME.

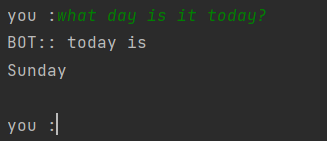


FIGURE 5.5. SHOWING CURRENT DAY.

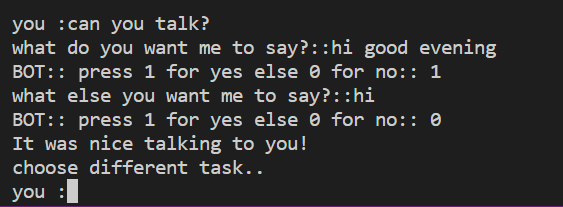


FIGURE 5.6. PERFORMING TEXT TO SPEECH OPERATION.

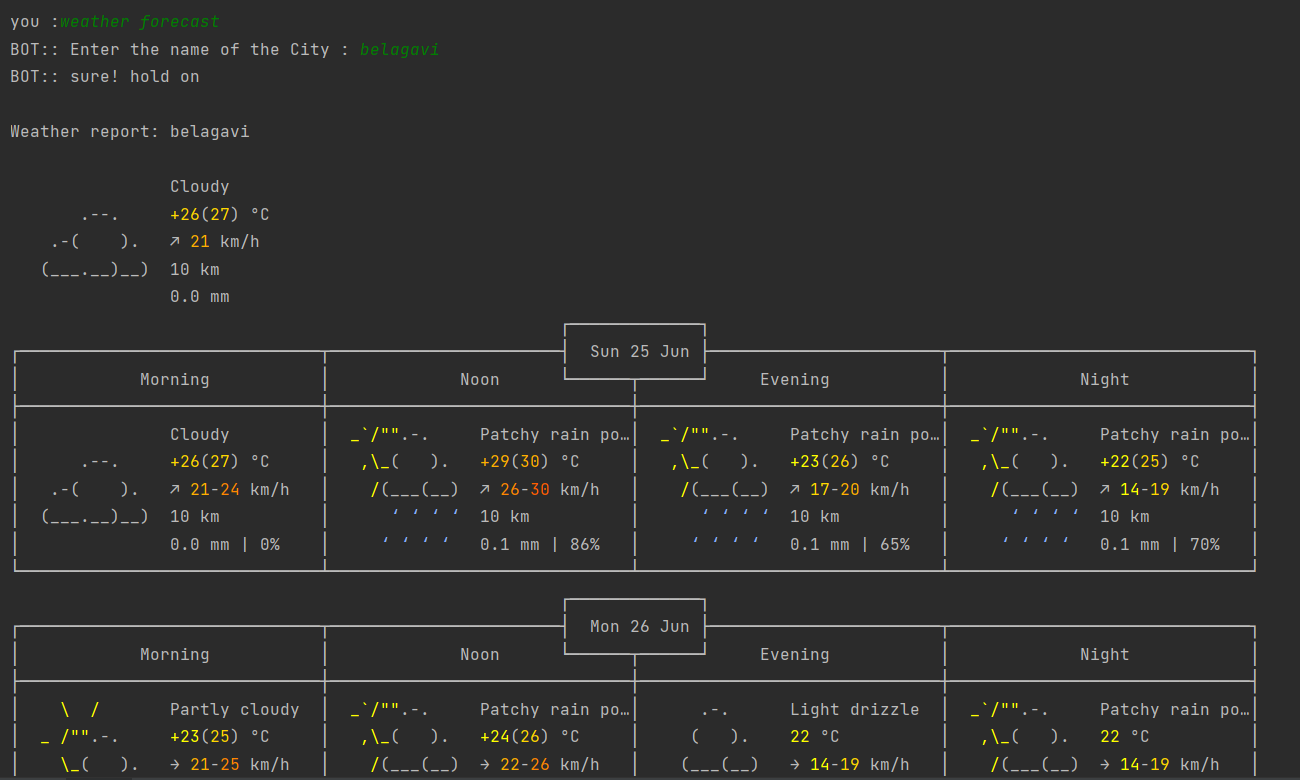


FIGURE 5.7. OUTPUT FOR WEATHER FORECAST OPERATION.

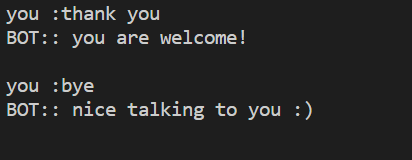


FIGURE 5.8. TERMINATION OF THE PROGRAM.

5. CONCLUSION

In this project, we have introduced a chatbot that is able to interact with user. This chatbot can answer queries in the textual user input. For this purpose, random() module has been used. The chatbot can answer only those questions which he has the answer in the list created for a particular question and selects one answer randomly and gives it as output. So, to increase the knowledge of the chatbot, we can add the Sports, News, Government and a lot more using the requests module and get the result from the internet. The next step towards building chatbots involves helping people to facilitate their work and interact with computers using natural language or using their set of rules.

6. REFERENCES

[1] **“Recruitment Chatbots”, International Research Journal of Engineering and Technology (IRJET), vol. 5, Issue: 08, Aug 2018[1]. Authors: Akash Balachandar, Anusha D Kulkarni.**