

INTERDISCIPLINARY PROJECT REPORT

at

**Sathyabama Institute of Science and Technology
(Deemed to be University)**

Submitted in partial fulfillment of the requirements for the
award of Bachelor of Engineering Degree in Computer Science and
Engineering with Data Science

By

JAYANTH TUNK

(Reg. No. 40733016)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Accredited with Grade "A" by NAAC | 12 B Status

by UGC | Approved by AICTE

**JEPPIAR NAGAR, RAJIV GANDHISALAI,
CHENNAI – 600119**

APRIL-2023



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the Bonafede work of **JAYANTH TUNK (40733016)** who carried out the project entitled "**SATHYABAMA CHATBOT**" under my supervision from Feb 2023 to April 2023.

Internal Guide Name

Ms. K. Abinaya M.E., (Ph.D.)

Head of the Department

Dr. S.Vigneshwari M.E., Ph.D.

17-04-2023

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I **Jayanth Tunk** hereby declare that the Project Report Entitled **Sathyabama Chatbot** done by me under the guidance of **Ms. K. Abinaya** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering with Data Science.

DATE: 17-04-2023



PLACE: Chennai

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to the **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala M.E., Ph.D., Dean**, School of Computing, **Dr. S. Vigneshwari M.E., Ph.D.**, Head of the Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Ms. K. Abinaya** for her valuable guidance, suggestions and constant encouragement paved the way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project

TRAINING CERTIFICATE

ABSTRACT

Voice assistants are software agents that can interpret human speech and respond via synthesized voices. Voice control is a major growing feature that change the way people can live. The voice assistant is commonly being used in smartphones and laptops. AI-based Voice assistants are the operating systems that can recognize human voice and respond via integrated voices. This chatbot is used to provide user assistance to the students and staff members. Voice assistants are software agents that can interpret human speech and respond via synthesized voices. Apple's Siri, Amazon's Alexa, Microsoft's Cortana, and Google's Assistant are the most popular voice assistants and are embedded in smartphones or dedicated home speakers. Users can ask their assistants questions, control home automation devices and media playback via voice, and manage other basic tasks such as email, to-do lists, and calendars with verbal commands. It can be **used by the students, staff, alumni** to extract information or get access to the people who can help them. This software is used to perform tasks such as quickly responding to users, informing them and to provide better support. It contains all types of information regarding the Institute like clubs, map, course details etc. Thereby, it decreases the efforts needed to access and deliver the information.

TABLE OF CONTENTS

Chapter No.	Title	Page No.
1	Introduction	3
	1.1 Smart Assistant	4
	1.2 Smart assistant working	4
	1.3 Types of chatbots	5
	1.4 Benefits of using Smart Assistant	5
	1.5 Disadvantages of Smart Assistant	6
	1.6 Python Programming	6
2	Aim & Scope of the project	9
	2.1 Aim	9
	2.2 Scope	9
3	Experimental or materials and methods	11
	3.1 Algorithm	11
	3.2 Python	11
	3.3 Modules used	11
	3.4 Artificial Neural Networks	14
	3.5 Recurrent Neural Network	15
	3.6 Architecture Diagram	18
	3.7 Flow Chart	18
4	Result and Discussion	19
	4.1 Results	19
5	CONCLUSION	20
6	REFERENCE	20

LIST OF FIGURES

Figure No.	Figure Name	Page No.
1.1	Speech Recognition	3
1.2	Satbot	5
1.3	Types of Voice Assistants	8
2.1	Speech Intake	10
3.1	RNN	16
3.2	RNN	17

LIST OF ABBREVIATIONS

Abbreviation	Expansion
RNN	Recurrent Neural Network
NLP	Natural Language Processing
ANN	Artificial Neural Network
GUI	Graphical User Interface
PIL	Python Imaging Library
Numpy	Numerical Python
Os	Operating System

CHAPTER 1

INTRODUCTION

This project is focusing on creating a chatbot to be used by students to get their queries responded easily from the college 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).



Fig 1.1

1.1 Smart assistant

A smart assistant – also known as a virtual assistant – is a form of software installed in a smart device (such as a smart speaker or a smart phone) that can perform tasks or services, or answer questions. They are essentially the link between us and smart technology software and are essential for controlling smart plugs, smart thermostats, and similar devices.

In order to communicate with your smart assistant, you will need a smart speaker for it to 'live' inside of. Popular smart assistants include Alexa, Siri and Google Assistant.

1.2 Smart assistant working

Voice control allows you to interact with your smart assistant, without lifting a finger. Whether controlling wider smart technology or searching the cloud through the internet to find exactly the information you need, this handy feature makes daily life easier and more efficient.

Though you can change from these default settings, here's how to get chatty with your smart assistant. Just follow these 'wake words' with the request you need:

Google Assistant: 'OK Google...'

Apple Siri: 'Hey Siri...'

Amazon Alexa: 'Alexa...'

Microsoft Cortana: 'Hey Cortana...'

Samsung Bixby: 'Hi Bixby...'

You can also usually wake up the assistants at the press of a button – depending on the device you are using. On an iPhone for instance, you can long-press the home button to wake up Siri.



Fig 1.2

1.3 Types of chatbots?

- Menu/button-based chatbots.
- Linguistic Based (Rule-Based Chatbots)
- Keyword recognition-based chatbots.
- Machine Learning chatbots.
- The hybrid models.
- Voice bots.

1.4 Benefits of using virtual assistant

- Time savings
- Improved turnaround of time-consuming tasks
- Reduced costs for staffing and operations
- Increased scale of operations
- Expanded reach

- Improved customer service

1.5 Disadvantages of Smart Assistant

One of the significant drawbacks of voice assistants is that they make communication less personal. There is something inherently cold about a disembodied voice at a machine dictating your messages and tasks rather than an actual person. The lack of a physical presence takes some of that warmth out.

While voice assistants help increase productivity, they also lack accountability. Email and other systems are a way to hold an employee accountable for completing tasks on time. With voice assistants like Alexa or Siri, there's no such system in place and more likely that some of the tasks may get lost without an assigned deadline date.

Moreover, voice assistants lack proper feedback mechanisms. A task could take too long to be completed, but you would not know unless you asked if it was done yet.

So, it is hard to track progress by using these digital assistants. The lack of oversight can lead to disorganized processes throughout your business.

1.6 Python Programming

Python is a popular programming language. It was created by Guido van Rossum, and released in 1991.

It is used for:

- web development (server-side)
- software development
- mathematics
- system scripting

What can Python do?

- Python can be used on a server to create web applications.
- Python can be used alongside software to create workflows.
- Python can connect to database systems. It can also read and modify files.
- Python can be used to handle big data and perform complex mathematics.
- Python can be used for rapid prototyping, or for production-ready software development.

Why Python?

- Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc.)
- Python has a simple syntax like the English language.
- Python has syntax that allows developers to write programs with fewer lines than some other programming languages.
- Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.
- Python can be treated in a procedural way, an object-oriented way or a functional way.

Advantages of Python Programming Language:

- Smooth learning curve
- High speed of development
- Portability and extensibility to other languages
- Versatility plus extensive toolset for almost everything

Disadvantages of Python Programming Language:

- Speed limitations
- No multithreading
- High memory consumption
- Hassles with mobile and front-end development

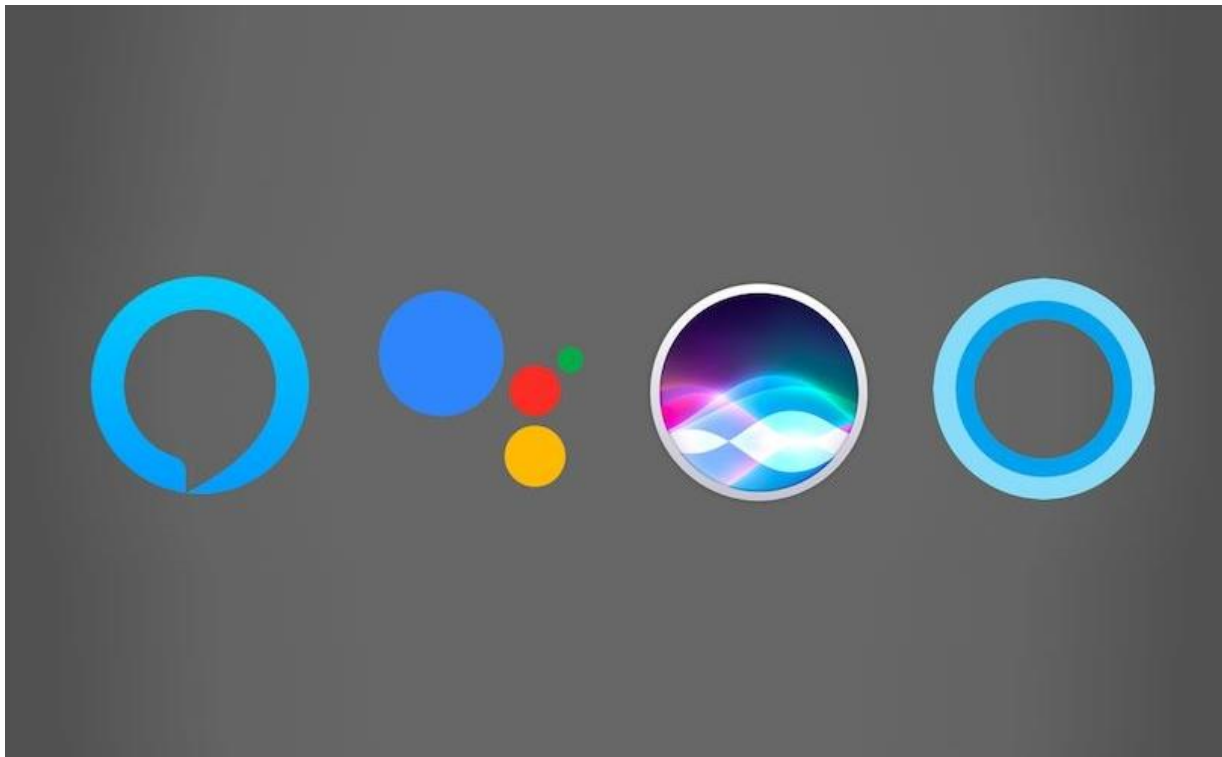


Fig 1.3

CHAPTER 2

AIM AND SCOPE OF SMART ASSISTANT

2.1 Aim

To build a smart assistant that can simply things and can be used by the staff and students of our university.

2.2 Scope

Voice assistants are one of many 'smart' devices making their way into our lives that make up the Internet of Things. These are devices that are connected to wireless networks which can communicate with other devices within our homes.

In 2017, there was an estimated 27 billion Internet of Things devices and this is expected to grow by 12% every year to reach more than 125 billion devices by 2030. It is clear big tech companies are investing heavily in these devices, so it will be interesting to consider how these will have developed by 2030.

TASKS VOICE ASSISTANTS BE ABLE TO DO IN THE FUTURE

The ambitions of some companies show what voice assistants could successfully do in the future. As an example, Google has developed Google Duplex, which is a technology that can conduct natural conversations and carry out "real world" tasks over the phone.

Users can ask questions or give commands through Google Assistant and this can then use Google Duplex to carry out specific tasks such as making a restaurant booking or scheduling a hair appointment.

ROLES BY AFFECTED BY VOICE ASSISTANTS?

Using Google Duplex as an example, roles that heavily require phone calling may be open to disruption. Google Duplex is seen as beneficial to businesses by being able to arrange appointments, gather information from different sources and address accessibility and language barriers.

Therefore, roles such as a PA could be changed with improvements in voice assistant technology. However, they will not necessarily be made redundant, as day to day interactions can be complex for a computer and there would need to be a human operator to defer to from time to time for a difficult task to be completed. It could be that a PA turns into a VA manager.

There can be no doubt that voice assistants are, and will continue to become, a great feat of human ingenuity and they are already creeping into our lives in some shape or form. With the eventual roll-out of 5G and the improvement in machine learning voice assistants may be setting themselves up to be a tool we cannot live without.

However, before we get to that stage, there are hurdles to cross which include heavy investment, improvement in the technology and confidence from consumers that this device that is in their lives does not pose a risk to their privacy.



Fig 2.1

CHAPTER 3

Experimental or materials and methods; algorithms

3.1 Algorithm

Step 1: Import the modules.

Step 2: Implementing the functionality to allow the users to pass command line arguments.

Step 3: Writing function that understands voice commands

1. Create a voice request.
2. Create a function.
3. Execute the function based on the input.
4. Send the combined frame and receive responses.
5. Parse the responses and print the results.

Step 4: Writing function to print the results in a certain format.

3.2 Python

Python is a computer programming language often used to build websites and software, automate tasks, and conduct data analysis. Python is a general-purpose language, meaning it can be used to create a variety of different programs and is not specialized for any specific problems. This versatility, along with its beginner-friendliness, has made it one of the most-used programming languages today.

3.3 Modules used

To build a personal voice assistant it is necessary to install the following packages in your system using the pip command.

1) Tkinter -- Python offers multiple options for developing GUI (Graphical User Interface). Out of all the GUI methods, tkinter is the most used method. It is a standard Python interface to the Tk GUI toolkit shipped with Python. Python with tkinter is the fastest and easiest way to create the GUI applications. Creating a GUI using tkinter is an easy task.

- To create a tkinter app:
- Importing the module – tkinter
- Create the main window (container)
- Add any number of widgets to the main window
- Apply the event Trigger on the widgets.

Importing tkinter is same as importing any other module in the Python code. Note that the name of the module in Python 2.x is 'Tkinter' and in Python 3.x it is 'tkinter'.

You can install tkinter using the following command:

pip install tk

2) PIL -- PIL stands for Python Imaging Library adds image processing capabilities to your Python interpreter.

This library provides extensive file format support, an efficient internal representation, and powerful image processing capabilities.

The core image library is designed for fast access to data stored in a few basic pixel formats. It should provide a solid foundation for a general image processing tool.

3) **Speech recognition** — Speech recognition is an important feature used in house automation and in artificial intelligence devices. The main function of this library is it tries to understand whatever the humans speak and converts the speech to text.

4) **pyttsx3** — pyttsx3 is a text to speech conversion library in python. This package supports text to speech engines on Mac os x, Windows and on Linux.

5) **wikipedia** — Wikipedia is a multilingual online encyclopaedia used by many people from academic community ranging from freshmen to students to professors who wants to gain information over a particular topic. This package in python extracts data's required from Wikipedia.

6) **ecapture** — This module is used to capture images from your camera

7) **datetime** — This is an inbuilt module in python and it works on date and time

8) **os** — This module is a standard library in python and it provides the function to interact with operating system

9) **time** — The time module helps us to display time

10) **Web browser** — This is an in-built package in python. It extracts data from the web

11) **request**- The request module is used to send all types of HTTP request. Its accepts URL as parameters and gives access to the given URL'S.

12) **wolfram alpha** — Wolfram Alpha is an API which can compute expert-level answers using Wolfram's algorithms, knowledge base and AI technology. It is made possible by the Wolfram Language.

13) **Numpy** -- NumPy is a Python library used for working with arrays. It also has functions for working in domain of linear algebra, Fourier transform, and matrices. NumPy was created in 2005 by Travis Oliphant. It is an open-source project and you can use it freely.

NumPy stands for Numerical Python

Why Use NumPy?

In Python we have lists that serve the purpose of arrays, but they are slow to process. NumPy aims to provide an array object that is up to 50x faster than traditional Python lists. The array object in NumPy is called ndarray, it provides a lot of supporting functions that make working with ndarray very easy. 12 Arrays are very frequently used in data science, where speed and resources are very important.

3.4 Artificial Neural Networks

Artificial neural networks (ANN) or connectionist systems are computing systems vaguely inspired by the biological neural networks that constitute animal brains. Such systems "learn" to perform tasks by considering examples, generally without being programmed with task-specific rules. For example, in image recognition, they might learn to identify images that contain cats by analyzing example images that have been manually labelled as "cat" or "no cat" and using the results to identify cats in other images. They do this without any prior knowledge of cats, for example, that they have fur, tails, whiskers, and cat-like faces. Instead, they automatically generate identifying

characteristics from the examples that they process. An ANN is based on a collection of connected units or nodes called artificial neurons, which loosely model the neurons in a biological brain. Each connection, like the synapses in a biological brain, can transmit a signal to other neurons. An artificial neuron that receives a signal then processes it and can signal neurons connected to it. In ANN implementations, the "signal" at a connection is a real number, and the output of each neuron is computed by some nonlinear function of the sum of its inputs. The connections are called edges. Neurons and edges typically have a weight that adjusts as learning proceeds. The weight increases or decreases the strength of the signal at a connection. Neurons may have a threshold such that a signal is sent only if the aggregate signal crosses that threshold. Typically, neurons are aggregated into layers. Different layers may perform different transformations on their inputs. Signals travel from the first layer (the input layer), to the last layer (the output layer), possibly after traversing the layers multiple times. The original goal of the ANN approach was to solve problems in the same way that a human brain would. But over time, attention moved to performing specific tasks, leading to deviations from biology. ANNs have been used on a variety of tasks, including computer vision, speech recognition, machine translation, social network filtering, playing board and video games, medical diagnosis, and even in activities that have traditionally been considered as reserved to humans, like painting.

3.5 Recurrent Neural Network (RNN)

It is a type of Neural Network where the output from the previous step is fed as input to the current step. In traditional neural networks, all the inputs and outputs are independent of each other, but in cases like when it is required to predict the next word of a sentence, the previous words are required and hence there is a need to remember the previous words. Thus, RNN came into existence, which solved this issue with the help of a Hidden Layer. The main and most important feature of RNN is Hidden state, which remembers some information about a sequence.

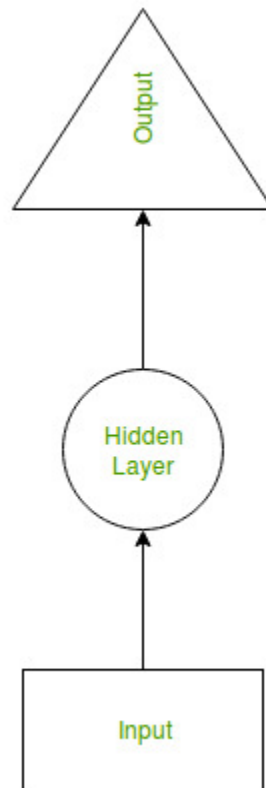


Fig 3.1

RNN

RNN have a “memory” which remembers all information about what has been calculated. It uses the same parameters for each input as it performs the same task on all the inputs or hidden layers to produce the output. This reduces the complexity of parameters, unlike other neural networks.

How RNN works

The working of an RNN can be understood with the help of the below example:

Example: Suppose there is a deeper network with one input layer, three hidden layers, and one output layer. Then like other neural networks, each hidden layer will have its own set of weights and biases, let's say, for hidden layer 1 the weights and biases

are (w_1, b_1) , (w_2, b_2) for the second hidden layer, and (w_3, b_3) for the third hidden layer. This means that each of these layers is independent of the other, i.e. they do not memorize the previous outputs.

Now the RNN will do the following:

RNN converts the independent activations into dependent activations by providing the same weights and biases to all the layers, thus reducing the complexity of increasing parameters and memorizing each previous output by giving each output as input to the next hidden layer.

Hence these three layers can be joined together such that the weights and bias of all the hidden layers are the same, in a single recurrent layer.

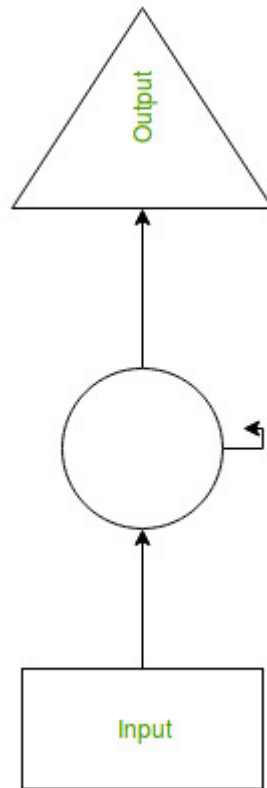
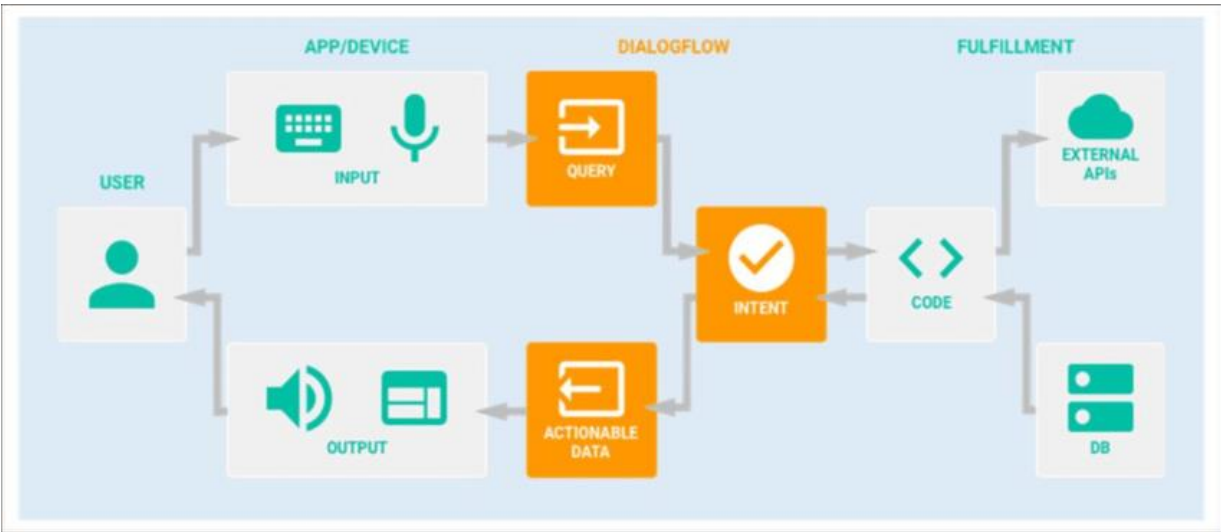


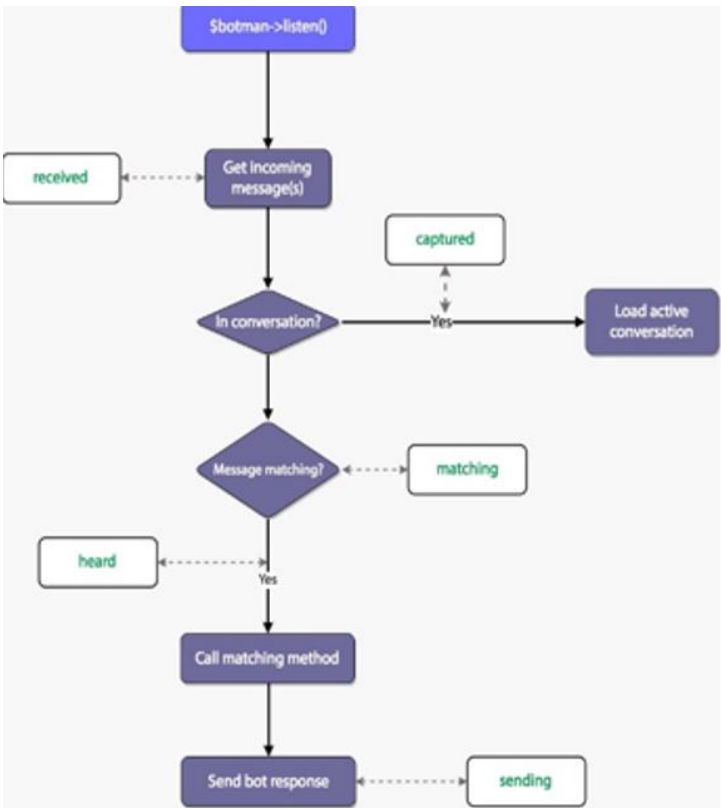
Fig 3.2

RNN

3.6 Architecture Diagram



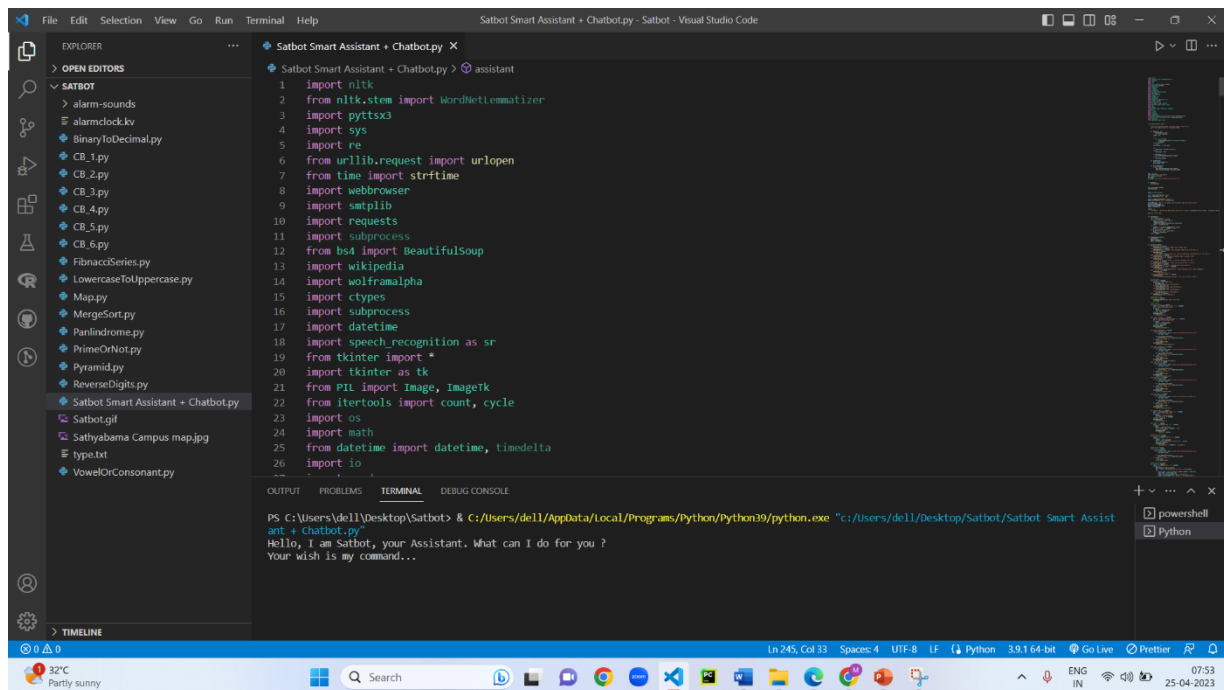
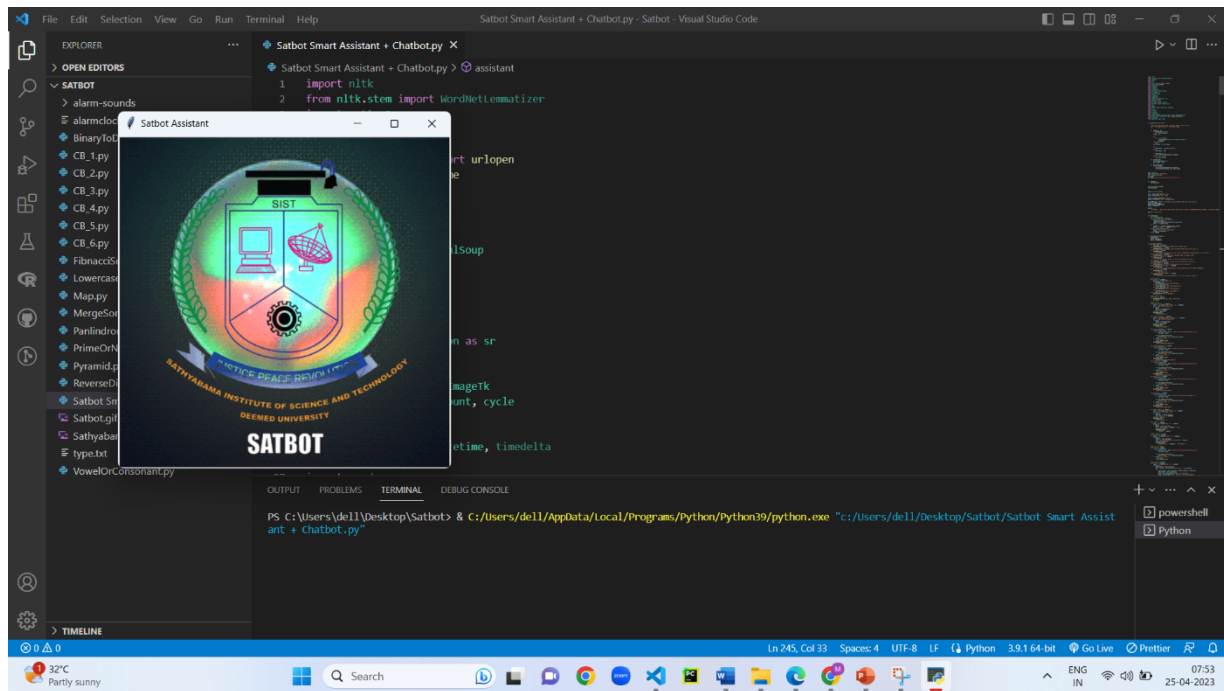
3.7 Flowchart of Virtual Assistant



CHAPTER -4

RESULTS & DISCUSSION

4.1 Results



CHAPTER -5

CONCLUSION

Smart assistant use has spread with impressive speed. The voice assistant's technology is more than affordable and offers many benefits to its users. Having a personal assistant with access to the unlimited knowledge stored on the internet. However, smart speakers are not just another consumer electronics device. They are a completely new sales channel for local small and medium-sized businesses.

REFERENCES

1. <https://pypi.org/project/SpeechRecognition/>
2. <https://pypi.org/project/argparse/>
3. <https://pypi.org/project/wikipedia/>
4. <https://pypi.org/project/nltk/>
5. A. Kumar and F. Shaik, "Importance of Image Processing", 2016.
6. Chenjing Yan, Congyan Lang, Tao Wang, Xuetao Du, and Chen Zhang, "Age Estimation Based on Convolutional Neural Network", 2014 Springer International Publishing Switzerland.
7. Hang Qi and Liqing Zhang, "Age Classification System with ICA Based Local Facial Features", 2009 Springer-Verlag Berlin Heidelberg.
8. Eran Eidinger, Roee Enbar, and Tal Hassner, "Age and Gender Estimation of Unfiltered Faces", 2014 IEEE.
10. Aditya K. Saxena, Shweta Sharma and Vijay K. Chaurasiya, "Neural Network based Human Age-group Estimation in Curvelet Domain", 2015 The Authors Published by Elsevier.
11. J. Chen, S. Shan, C. He, G. Zhao, M. Pietikainen, X. Chen, and W. Gao. Wld: "A robust local image descriptor", Trans.Pattern Anal. Mach. Intell.

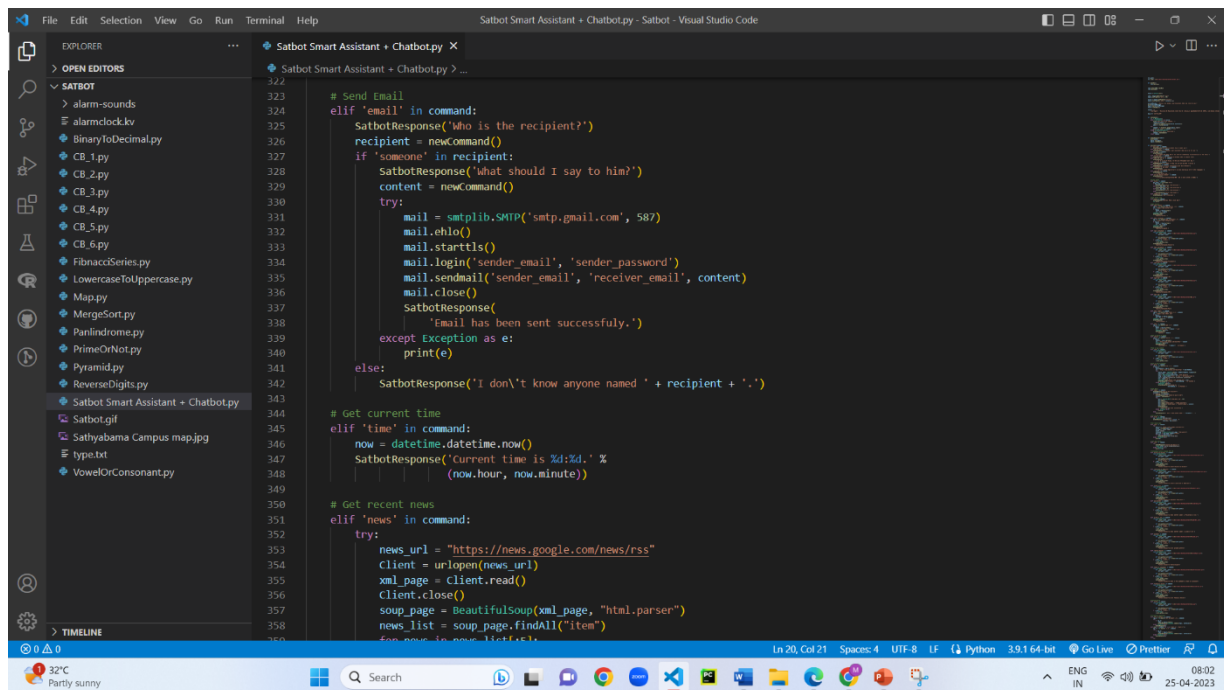
Appendix

Source Code:

<https://github.com/JayanthTunk/Sathyabama-SmartAssistant> SATBOT

Screenshots:

Python File



```
322
323 # Send Email
324 elif 'email' in command:
325     SatbotResponse('who is the recipient?')
326     recipient = newCommand()
327     if 'someone' in recipient:
328         SatbotResponse('what should I say to him?')
329         content = newCommand()
330         try:
331             mail = smtplib.SMTP('smtp.gmail.com', 587)
332             mail.ehlo()
333             mail.starttls()
334             mail.login('sender_email', 'sender_password')
335             mail.sendmail('sender_email', 'receiver_email', content)
336             mail.close()
337             SatbotResponse('Email has been sent successfully.')
338         except Exception as e:
339             print(e)
340     else:
341         SatbotResponse('I don\'t know anyone named ' + recipient + '.')
342
343 # Get current time
344 elif 'time' in command:
345     now = datetime.datetime.now()
346     SatbotResponse('Current time is %d:%d.' %
347                   (now.hour, now.minute))
348
349 # Get recent news
350 elif 'news' in command:
351     try:
352         news_url = "https://news.google.com/news/rss"
353         Client = urlopen(news_url)
354         xml_page = Client.read()
355         Client.close()
356         soup_page = BeautifulSoup(xml_page, "html.parser")
357         news_list = soup_page.findAll("item")
358         for news in news_list[:5]:
359             print(news.title.text)
```

```
248 elif 'open map' in command:
249     class Callpy(object):
250         def __init__(self, path="c:\\Users\\dell\\Desktop\\Satbot\\Map.py"):
251             self.path = path
252
253         def call_python_file(self):
254             call(["python", "{}".format(self.path)])
255
256     if __name__ == "__main__":
257         c = Callpy()
258         c.call_python_file()
259         SatbotResponse(
260             'Opening Sathyabama Map')
261
262 elif 'open reddit' in command:
263     reg_ex = re.search('open reddit (.*)', command)
264     url = 'https://www.reddit.com/'
265     if reg_ex:
266         subreddit = reg_ex.group(1)
267         url = url + 'r/' + subreddit
268         webbrowser.open(url)
269         SatbotResponse(
270             'Opening Reddit.')
271
272 elif 'open' in command:
273     reg_ex = re.search('open (.*)', command)
274     if reg_ex:
275         domain = reg_ex.group(1)
276         url = 'https://www.' + domain + ".com"
277         webbrowser.open(url)
278         SatbotResponse(
279             'Opening ' + domain)
280
281 # Make a search on Google
282 elif 'search' in command:
283     reg_ex = re.search('search (.*)', command)
284     if reg_ex:
285         subject = reg_ex.group(1)
```

```
198 elif 'open sathyabama' in command:
199     class Callpy(object):
200         def __init__(self, path="c:\\Users\\dell\\Desktop\\Satbot\\CB_3.py"):
201             self.path = path
202
203         def call_python_file(self):
204             call(["python", "{}".format(self.path)])
205
206     if __name__ == "__main__":
207         c = Callpy()
208         c.call_python_file()
209         SatbotResponse(
210             'Opening Sathyabama Website')
211
212 elif 'open placements' in command:
213     class Callpy(object):
214         def __init__(self, path="c:\\Users\\dell\\Desktop\\Satbot\\CB_4.py"):
215             self.path = path
216
217         def call_python_file(self):
218             call(["python", "{}".format(self.path)])
219
220     if __name__ == "__main__":
221         c = Callpy()
222         c.call_python_file()
223         SatbotResponse(
224             'Opening Sathyabama Placements')
225
226 elif 'open lms' in command:
227     class Callpy(object):
228         def __init__(self, path="c:\\Users\\dell\\Desktop\\Satbot\\CB_5.py"):
229             self.path = path
230
231         def call_python_file(self):
232             call(["python", "{}".format(self.path)])
233
234     if __name__ == "__main__":
235         c = Callpy()
236         c.call_python_file()
237         SatbotResponse(
```

This screenshot shows the Visual Studio Code editor with the file 'Satbot Smart Assistant + Chatbot.py' open. The 'EXPLORER' sidebar on the left lists various Python files, including 'Satbot Smart Assistant + Chatbot.py'. The main editor area displays the code for the 'newCommand()' function, which uses the 'SpeechRecognition' library to listen for voice commands and respond accordingly. The status bar at the bottom indicates the file is encoded in UTF-8, uses 4 spaces for indentation, and is running on a Python 3.9.164-bit interpreter.

```
def newCommand():
    r = sr.Recognizer()
    with sr.Microphone() as source:
        print('Your wish is my command...')
        r.pause_threshold = 1
        r.adjust_for_ambient_noise(source, duration=1)
        audio = r.listen(source)

    try:
        command = r.recognize_google(audio).lower()
        print('You said: ' + command + '\n')
    except sr.UnknownValueError:
        print('Sorry I can\'t understand')
        command = newCommand()
    return command

def SatbotResponse(audio):
    print(audio)
    engine.say(audio)
    engine.runAndWait()

def assistant(command):
    if 'your name' in command:
        SatbotResponse('My name is Satbot. Nice to meet you!')
    elif 'who are you' in command:
        SatbotResponse('I\'m Satbot, your assistant! What can I do for you?')
    elif 'a joke' in command:
        SatbotResponse('My neighbor has been mad at his wife for sunbathing. I personally am on the fence.')
    elif 'old are you' in command:
        SatbotResponse('I was launched in November 2022 by Jayanth Tunk')
    elif 'what can you do' in command:
        SatbotResponse('I can do a lot of things, to help you throughout your day.')
    elif 'hello me' in command:
```

This screenshot shows the Visual Studio Code editor with the same file 'Satbot Smart Assistant + Chatbot.py' open. The main editor area displays the code for the 'main()' function, which initializes the PyTTS3 engine, sets properties like 'rate' and 'voices', and starts the main loop. The status bar at the bottom indicates the file is encoded in UTF-8, uses 4 spaces for indentation, and is running on a Python 3.9.164-bit interpreter.

```
root = Tk()
root.title("Satbot Assistant")
lbl = ImageLabel(root)
lbl.pack()
lbl.load(r"Users\de11\Desktop\Satbot\Satbot.gif")

def mainWin():
    root.destroy()

root.after(3000, mainWin)
root.mainloop()

engine = pyttsx3.init()
rate = engine.getProperty('rate')
engine.setProperty('rate', 170)
voices = engine.getProperty('voices')
engine.setProperty('voice', voices[1].id)

welcomeMessage = 'Hello, I am Satbot, your Assistant. What can I do for you?'
print(welcomeMessage)
engine.say(welcomeMessage)
engine.runAndWait()

headers = {
    'user-agent': 'Mozilla/5.0 (Macintosh; Intel Mac OS X 10_11_6) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/53.0.2785.143 Safari/537.36'
}
app_id = 'wolfram_API'

def newCommand():
    r = sr.Recognizer()
```

This screenshot shows the Visual Studio Code editor with the file explorer on the left displaying a project named 'SATBOT'. The file 'Satbot Smart Assistant + Chatbot.py' is open in the editor. The code defines a class 'ImageLabel' that inherits from 'Label'. It includes a docstring, a 'load' method that takes an image or string and loads it into a list of frames, and a 'next_frame' method that returns the next frame in the sequence. The status bar at the bottom indicates the file is in Python 3.9.164-bit.

```
36 class ImageLabel(Label):
37     """
38     A Label that displays images, and plays them if they are gifs
39     :im: A PIL Image instance or a string filename
40     """
41
42     def load(self, im):
43         if isinstance(im, str):
44             im = image.open(im)
45             frames = []
46
47         try:
48             for i in count(1):
49                 frames.append(imageTk.PhotoImage(im.copy()))
50                 im.seek(1)
51         except EOFError:
52             pass
53         self.frames = cycle(frames)
54
55         try:
56             self.delay = im.info['duration']
57         except:
58             self.delay = 100
59
60         if len(frames) == 1:
61             self.config(image=next(self.frames))
62         else:
63             self.next_frame()
64
65     def unload(self):
66         self.config(image=None)
67         self.frames = None
68
69     def next_frame(self):
70         if self.frames:
71             self.config(image=next(self.frames))
72
```

This screenshot shows the Visual Studio Code editor with the same file 'Satbot Smart Assistant + Chatbot.py' open. The code displays a list of imports at the top of the file, including modules for text processing (nltk, nltk.stem, pytttsx3), web requests (urllib.request, requests), image handling (PIL, imageTk), and other utilities (itertools, random, warnings, subprocess). The status bar at the bottom indicates the file is in Python 3.9.164-bit.

```
1 import nltk
2 from nltk.stem import WordNetLemmatizer
3 import pytttsx3
4 import sys
5 import re
6 from urllib.request import urlopen
7 from time import strftime
8 import webbrowser
9 import smtplib
10 import requests
11 import subprocess
12 from bs4 import BeautifulSoup
13 import wikipedia
14 import wolframalpha
15 import ctypes
16 import subprocess
17 import datetime
18 import speech_recognition as sr
19 from tkinter import *
20 import tkinter as tk
21 from PIL import Image, ImageTk
22 from itertools import count, cycle
23 import os
24 import math
25 from datetime import datetime, timedelta
26 import io
27 import random
28 import string
29 import warnings
30 import numpy as np
31 from sklearn.feature_extraction.text import TfidfVectorizer
32 from sklearn.metrics.pairwise import cosine_similarity
33 import warnings
34 from subprocess import call
35
36
37 class ImageLabel(Label):
```

```
371 # Sathyabama Python programs CDD LAB
372 elif 'binary to decimal' in command:
373     class Callpy(object):
374         def __init__(self, path="C:\\Users\\dell\\Desktop\\Satbot\\BinaryToDecimal.py"):
375             self.path = path
376
377         def call_python_file(self):
378             call(["python", "{}".format(self.path)])
379
380     if __name__ == "__main__":
381         c = Callpy()
382         c.call_python_file()
383         SatbotResponse(
384             'Python program to convert Binary to decimal')
385
386 elif 'lowercase to uppercase' in command:
387     class Callpy2(object):
388         def __init__(self, path="C:\\Users\\dell\\Desktop\\Satbot\\LowercaseToUppercase.py"):
389             self.path = path
390
391         def call_python_file(self):
392             call(["python", "{}".format(self.path)])
393
394     if __name__ == "__main__":
395         c = Callpy2()
396         c.call_python_file()
397         SatbotResponse(
398             'Python program to convert Lowercase to Uppercase')
399
400 elif 'merge sort' in command:
401     class Callpy3(object):
402         def __init__(self, path="C:\\Users\\dell\\Desktop\\Satbot\\Mergesort.py"):
403             self.path = path
404
405         def call_python_file(self):
406             call(["python", "{}".format(self.path)])
407
408     if __name__ == "__main__":
409         c = Callpy3()
```

```
499         c.call_python_file()
500
501 # Ask general questions
502 elif 'tell me about' in command:
503     reg_ex = re.search('tell me about (.*)', command)
504     try:
505         if reg_ex:
506             topic = reg_ex.group(1)
507             SatbotResponse(wikipedia.summary(topic, sentences=3))
508     except Exception as e:
509         SatbotResponse(e)
510 elif any(c in command for c in ("what is", "what's")):
511     reg_ex = re.search('(.*)', command)
512     try:
513         if reg_ex:
514             topic = reg_ex.group(1)
515             SatbotResponse(wikipedia.summary(topic, sentences=2))
516     except Exception as e:
517         SatbotResponse(e)
518 else:
519     try:
520         # wolframalpha
521         client = wolframalpha.Client(app_id)
522         res = client.query(command)
523         answer = next(res.results).text
524         SatbotResponse(answer)
525     except:
526         try:
527             # wikipedia
528             SatbotResponse(wikipedia.summary(command, sentences=2))
529         except Exception as e:
530             SatbotResponse(e)
531
532 #####
533 #####
534
535 while True:
```

Additional Python Files

 alarmclock.kv	24-01-2023 14:33	KV File	5 KB
 BinaryToDecimal	21-04-2023 18:12	Python Source File	1 KB
 CB_1	25-03-2023 03:46	Python Source File	3 KB
 CB_2	21-04-2023 15:39	Python Source File	9 KB
 CB_3	21-04-2023 15:48	Python Source File	1 KB
 CB_4	21-04-2023 16:56	Python Source File	1 KB
 CB_5	21-04-2023 15:54	Python Source File	1 KB
 CB_6	24-04-2023 14:01	Python Source File	1 KB
 FibnacciSeries	21-04-2023 18:10	Python Source File	1 KB
 LowercaseToUppercase	21-04-2023 17:46	Python Source File	1 KB
 Map	21-04-2023 16:14	Python Source File	1 KB
 MergeSort	21-04-2023 18:23	Python Source File	1 KB
 Panlindrome	21-04-2023 18:08	Python Source File	1 KB
 PrimeOrNot	21-04-2023 17:52	Python Source File	1 KB
 Pyramid	21-04-2023 17:53	Python Source File	1 KB
 ReverseDigits	21-04-2023 18:49	Python Source File	1 KB
 Satbot Smart Assistant + Chatbot	24-04-2023 13:58	Python Source File	17 KB
 Satbot	21-04-2023 11:04	GIF File	20,093 KB
 Sathyabama Campus map	21-04-2023 16:13	JPG File	3,899 KB

Code

```
import nltk
from nltk.stem import WordNetLemmatizer
import pyttsx3
import sys
import re
from urllib.request import urlopen
from time import strftime
import webbrowser
import smtplib
import requests
import subprocess
from bs4 import BeautifulSoup
import wikipedia
import wolframalpha
import ctypes
import subprocess
import datetime
import speech_recognition as sr
from tkinter import *
import tkinter as tk
from PIL import Image, ImageTk
from itertools import count, cycle
import os
import math
from datetime import datetime, timedelta
import io
import random
import string
import warnings
import numpy as np
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity
import warnings
from subprocess import call

class ImageLabel(tk.Label):
    """
    A Label that displays images, and plays them if they are gifs
    :im: A PIL Image instance or a string filename
    """

    def load(self, im):
```

```

if isinstance(im, str):
    im = Image.open(im)
frames = []

try:
    for i in count(1):
        frames.append(ImageTk.PhotoImage(im.copy()))
        im.seek(i)
except EOFError:
    pass
self.frames = cycle(frames)

try:
    self.delay = im.info['duration']
except:
    self.delay = 100

if len(frames) == 1:
    self.config(image=next(self.frames))
else:
    self.next_frame()

def unload(self):
    self.config(image=None)
    self.frames = None

def next_frame(self):
    if self.frames:
        self.config(image=next(self.frames))
        self.after(self.delay, self.next_frame)

root = tk.Tk()
root.title("Satbot Assistant")
lbl = ImageLabel(root)
lbl.pack()
lbl.load(r"\\Users\\dell\\Desktop\\Satbot\\Satbot.gif")

def mainWin():
    root.destroy()

root.after(3000, mainWin)
root.mainloop()

```

```

engine = pyttsx3.init()

rate = engine.getProperty('rate')
engine.setProperty('rate', 170)

voices = engine.getProperty('voices')
engine.setProperty('voice', voices[1].id)

welcomeMessage = 'Hello, I am Satbot, your Assistant. What can I do for you ?'
print(welcomeMessage)
engine.say(welcomeMessage)
engine.runAndWait()

headers = {
    'user-agent': 'Mozilla/5.0 (Macintosh; Intel Mac OS X 10_11_6) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/53.0.2785.143 Safari/537.36'}

app_id = 'wolfram_API'

def newCommand():
    r = sr.Recognizer()
    with sr.Microphone() as source:
        print('Your wish is my command...')
        r.pause_threshold = 1
        r.adjust_for_ambient_noise(source, duration=1)
        audio = r.listen(source)
    try:
        command = r.recognize_google(audio).lower()
        print('You said: ' + command + '\n')
    except sr.UnknownValueError:
        print('Sorry I can\'t understand')
        command = newCommand()
    return command

def SatbotResponse(audio):
    print(audio)
    engine.say(audio)
    engine.runAndWait()

def assistant(command):
    if 'your name' in command:
        SatbotResponse('My name is Satbot. Nice to meet you!')

```

```

elif 'who are you' in command:
    SatbotResponse('I\'m Satbot, your assistant! What can i do for you ?')
elif 'a joke' in command:
    SatbotResponse(
        'My neighbor has been mad at his wife for sunbathing. I personally am on the
fence.')
elif 'old are you' in command:
    SatbotResponse('I was launched in November 2022 by Jayanth Tunk')
elif 'what can you do' in command:
    SatbotResponse(
        'I can do a lot of things, to help you throughout your day.')
elif 'help me' in command:
    SatbotResponse('I\'m here to help, you can ask me what I can do.')
elif 'you single' in command:
    SatbotResponse('I am in a relationship with Harry Potter3')
elif 'languages do you speak' in command:
    SatbotResponse(
        'I currently speak English but I am also learning a lot of other languages ')
elif 'love you' in command:
    SatbotResponse(
        'I love myself too...')
elif 'are you dating someone' in command:
    SatbotResponse(
        'I am currently dating Harsha BRO ! who is also called as daddy ')

# Greet Satbot
elif 'hello' in command:
    day_time = int(strftime('%H'))
    if day_time < 12:
        SatbotResponse('Hello, Good morning!')
    elif 12 <= day_time < 16:
        SatbotResponse('Hello, Good afternoon!')
    elif 16 <= day_time < 18:
        SatbotResponse('Hello, Good evening!')
    else:
        SatbotResponse('Hello, Good night!')
elif 'thank you' in command:
    SatbotResponse('You\'re Welcome!')

# Make Satbot stop
elif 'exit' in command:
    SatbotResponse('Bye bye. Have a nice day!')
    sys.exit()

# Open Twitter
elif 'open twitter' in command:

```

```

reg_ex = re.search('open twitter (.*)', command)
url = 'https://www.twitter.com/'
if reg_ex:
    handle = reg_ex.group(1)
    url = url + handle
webbrowser.open(url)
SatbotResponse(
    'Opening Twitter.')

elif 'open instagram' in command:
    reg_ex = re.search('open instagram (.*)', command)
    url = 'https://www.instagram.com/'
    if reg_ex:
        handle = reg_ex.group(1)
        url = url + handle
    webbrowser.open(url)
    SatbotResponse(
        'Opening Instagram.')

elif 'open sathyabama' in command:
    class Callpy(object):
        def __init__(self, path=r"C:\Users\dell\Desktop\Satbot\CB_3.py"):
            self.path = path

        def call_python_file(self):
            call(["python", "{}".format(self.path)])
    if __name__ == "__main__":
        c = Callpy()
        c.call_python_file()
    SatbotResponse(
        'Opening Sathyabama Website')

elif 'open placements' in command:
    class Callpy(object):
        def __init__(self, path=r"C:\Users\dell\Desktop\Satbot\CB_4.py"):
            self.path = path

        def call_python_file(self):
            call(["python", "{}".format(self.path)])
    if __name__ == "__main__":
        c = Callpy()
        c.call_python_file()
    SatbotResponse(
        'Opening Sathyabama Placements')

```