**AI VOICE ASSISTANT**

**INTERNSHIP REPORT**

A Report Submitted to

Jawaharlal Nehru Technological University Kakinada, Kakinada in partial fulfillment for the award of the degree of

## BACHELOR OF TECHNOLOGY

**IN**

**COMPUTER SCIENCE AND ENGINEERING**

Submitted by

**K. Divya (20KN1A0577)**

**G. Pavani (21KN5A0509)**

**N. Jaswanth (20KN1A05A5)**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

# NRI INSTITUTE OF TECHNOLOGY

**Autonomous**

##### (Approved by AICTE, Permanently Affiliated to JNTUK, Kakinada) Accredited by NBA (CSE, ECE & EEE), Accredited by NAAC with ‘A’ Grade ISO 9001: 2015 CertifiedInstitution

**Pothavarappadu (V), (Via) Nunna, Agiripalli (M), Krishna Dist., PIN: 521212, A.P, India.**

**2022-2023**

|  |  |
| --- | --- |
| NRI | **NRI INSTITUTE OF TECHNOLOGY**  (An Autonomous Institution, Approved by AICTE, Permanently Affiliated to JNTUK, Kakinada)  Accredited by NBA (CSE, ECE & EEE), Accredited by NAAC with ‘A’ Grade ISO 9001: 2015 Certified Institution  Pothavarappadu (V), (Via) Nunna, Agiripalli (M), Krishna Dist., PIN: 521212, A.P, India. |

**CERTIFICATE**

This is to certify that the “**Internship report”** submitted by K. DIVYA, G. PAVANI, N. JASWANTH**(Regd. No.:20KN1A0577, 21KN5A0509, 20KN1A05A5)** is work done by his/her and submitted during YEARS academic year, in partial fulfillment of the requirements for the award of the degree of **BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE AND ENGINEERING,** at **BLACKBUCK ENGINEERS PVT LTD, Road No:36, Jubilee Hills, Hyderabad, Telangana.**

**INTERNSHIP COORDINATOR Head of the Department**

##### 

(R KATHYAYANI) (Dr. D. SUNEETHA)

##### 

**EXTERNAL EXAMINER**

## CERTIFICATE OF INTERNSHIP

## ACKNOWLEDGEMENT

I take this opportunity to thank all who have rendered their full support to my work. The pleasure, the achievement, the glory, the satisfaction, the reward, the appreciation and the construction of my project cannot be expressed with a few words for their valuable suggestions.

I am expressing my heartfelt thanks to **Head of the Department, Dr. D. SUNEETHA**

garu for her continuous guidance for completion of my Project work.

I am extending our sincere thanks to **Dean of the Department, Dr. K. V. SAMBASIVA RAO** for his continuous guidance and support to complete my project successfully.

I am thankful to the **Principal, Dr. C. NAGA BHASKAR** garu for his encouragement to complete the Project work.

I am extending my sincere and honest thanks to the **Chairman, Dr. R. VENKATA RAO garu & Secretary, Sri K. Sridhar** garu for their continuous support in completing the Project work.

Finally, I thank the Administrative Officer, Staff Members, Faculty of Department of CSE, NRI Institute of Technology and my friends, directly or indirectly helped us in the completion of this project.

**K. DIVYA (20KN1A0577)**

**G. PAVANI (21KN5A0509)**

**N. JASWANTH (20KN1A05A5)**

# ABSTRACT

## The main goal of Artificial intelligence (AI) is the realization of natural dialogue between humans and machines. We use python as a programming language because it have a major libraries which is use to execute commands. By using python installer packages our personal virtual assistant recognizes the user voice and process on it.Voice assistants are the great innovation in the field of AI that can change the way of living of the people in a different manner.. Initially, the voice assistant was mostly being used in smartphones and laptops but now it is also coming as home automation and smart speakers. Many devices are becoming smarter in their own way to interact with human in an easy language. The Desktop based voice assistant are the programs that can recognize human voices and can respond via integrated voice system. This paper will define the working of a voice assistants, their main problems and limitations. In this paper it is described that the method of creating a voice assistant without using cloud services, which will allow the expansion of such devices in the future*.*

## Organization Information:

Blackbuck Engineers is started in 2013 with the aim of creating a great ecosystem of academia, research, industry, and individuals. Blackbucks is a premier partner to Govts International Institute of Digital Technologies, and IITs. Blackbuck delivers the TAPTAP AI Driven employability platform to transform the journey of students towards their dream goals while helping 𝗛𝗥𝘀 𝗵𝗶𝗿𝗲 𝗿𝗶𝗴𝗵𝘁 𝘀𝘁𝘂𝗱𝗲𝗻𝘁𝘀.

Blackbuck has the largest chain of excellence in emerging tech across India.

Blackbucks runs post graduation programs in AI, ML and Data Science

www.theblackbucks.com

## Programs and opportunities:

#### This ground-up approach helps us deliver not only the solution to our clients but also add value to at the core Blackbuck Engineers which operates in Five specific domains namely TapTap - AI Driven, Post Graduation Programs, Center of Excellence, Virtual Programming Labs and Happie Days - A social Networking site for the students. TapTap offer services in Campus Recruitment drives for the Employers as well as College authorities. Recruiters can Conduct Customized Online Assessments secured with Best-in-class Proctoring and Schedule the end-to-end hiring process. Under each division we further provide specific industry solutions on focused domains with cutting edge technologies. Blackbuck Engineers emphasize on building relationships with our clients by delivering projects on time and within budget.

**INDEX**

# S.no CONTENTS Page no

1. [Introduction 1](#_TOC_250007)
   1. Modules 2
2. Analysis 3
3. [Software requirements specifications 4](#_TOC_250006)
4. [Technology 5](#_TOC_250005)
   1. [PACKAGES 5](#_TOC_250004)
   2. PYTHON 6
   3. HTML 8
   4. [JAVASCRIPT](#_TOC_250003) 9
   5. CSS………………………………………………………………………………10
   6. JSON 11
5. [Coding… 1](#_TOC_250002)2
6. [Screenshots 1](#_TOC_250001)9
7. [Conclusion](#_TOC_250000) 21
8. Bibilography 22

**Learning Objectives/Internship Objectives**

* Internships are generally thought of to be reserved for college students looking to gain experience in a particular field. However, a wide array of people can benefit from Training Internships in order to receive real world experience and develop their skills.
* An objective for this position should emphasize the skills you already possess in the area and your interest in learning more
* Internships are utilized in a number of different career fields, including architecture, engineering, healthcare, economics, advertising and many more.
* Some internship is used to allow individuals to perform scientific research while others are specifically designed to allow people to gain first-hand experience working.
* Utilizing internships is a great way to build your resume and develop skills that can be emphasized in your resume for future jobs. When you are applying for a Training Internship, make sure to highlight any special skills or talents that can make you stand apart from the rest of the applicants so that you have an improved chance of landing the position.

**WEEKLY OVERVIEW OF INTERNSHIP ACTIVITIES**

|  |  |  |  |
| --- | --- | --- | --- |
| **1st WEEK** | **DATE** | **DAY** | **NAME OF THE TOPIC/MODULE COMPLETED** |
| 13.06.2022 | Monday | Introduce the Topic & the Problem Statement |
| 14.06.2022 | Tuesday | Introduce the Topic & the Problem Statement |
| 15.06.2022 | Wednesday | Introduce the Topic & the Problem Statement |
| 16.06.2022 | Thursday | Introduce the Topic & the Problem Statement |
| 17.06.2022 | Friday | Introduce the Topic & the Problem Statement |

|  |  |  |  |
| --- | --- | --- | --- |
| **2nd WEEK** | **DATE** | **DAY** | **NAME OF THE TOPIC/MODULE COMPLETED** |
| 20.06.2022 | Monday | Abstract Building |
| 21.06.2022 | Tuesday | Abstract Building |
| 22.06.2022 | Wednesday | Abstract Building |
| 23.06.2022 | Thursday | Abstract Building |
| 24.06.2022 | Friday | Abstract Submission |

|  |  |  |  |
| --- | --- | --- | --- |
| **3rd WEEK** | **DATE** | **DAY** | **NAME OF THE TOPIC/MODULE COMPLETED** |
| 27.06.2022 | Monday | Abstract Submission |
| 28.06.2022 | Tuesday | Abstract Submission |
| 29.06.2022 | Wednesday | Explain your Approach to Solving Problem |
| 30.06.2022 | Thursday | Explain your Approach to Solving Problem |
| 01.07.2022 | Friday | Explain your Approach to Solving Problem |

|  |  |  |  |
| --- | --- | --- | --- |
| **4th WEEK** | **DATE** | **DAY** | **NAME OF THE TOPIC/MODULE COMPLETED** |
| 04.07.2022 | Monday | Explain your Approach to Solving Problem |
| 05.07.2022 | Tuesday | Explain Structure of Project |
| 06.07.2022 | Wednesday | Explain Structure of Project |
| 07.07.2022 | Thursday | Explain Structure of Project |
| 08.07.2022 | Friday | Explain Structure of Project |

|  |  |  |  |
| --- | --- | --- | --- |
| **5th WEEK** | **DATE** | **DAY** | **NAME OF THE TOPIC/MODULE COMPLETED** |
| 11.07.2022 | Monday | Data Preprocessing |
| 12.07.2022 | Tuesday | Data Preprocessing |
| 13.07.2022 | Wednesday | Data Preprocessing |
|  | 14.07.2022 | Thursday | Data Preprocessing |
|  | 15.07.2022 | Friday | Data Preprocessing |

|  |  |  |  |
| --- | --- | --- | --- |
| **6th WEEK** | **DATE** | **DAY** | **NAME OF THE TOPIC/MODULE COMPLETED** |
| 18.07.2022 | Monday | Perform Analysis |
| 19.07.2022 | Tuesday | Perform Analysis |
| 20.07.2022 | Wednesday | Perform Analysis |
|  | 21.07.2022 | Thursday | Perform Analysis |
|  | 22.07.2022 | Friday | Perform Analysis |

|  |  |  |  |
| --- | --- | --- | --- |
| **7th WEEK** | **DATE** | **DAY** | **NAME OF THE TOPIC/MODULE COMPLETED** |
| 25.07.2022 | Monday | PPT Preparation |
| 26.07.2022 | Tuesday | PPT Preparation |
| 27.07.2022 | Wednesday | PPT Preparation |
|  | 28.07.2022 | Thursday | PPT Preparation |
|  | 29.07.2022 | Friday | PPT Preparation |

|  |  |  |  |
| --- | --- | --- | --- |
| **8th WEEK** | **DATE** | **DAY** | **NAME OF THE TOPIC/MODULE COMPLETED** |
| 01.08.2022 | Monday | PPT Submission |
| 02.08.2022 | Tuesday | PPT Submission |
| 03.08.2022 | Wednesday | Mid Review |
|  | 04.08.2022 | Thursday | Mid Review |
|  | 05.08.2022 | Friday | Mid Review |

|  |  |  |  |
| --- | --- | --- | --- |
| **9th WEEK** | **DATE** | **DAY** | **NAME OF THE TOPIC/MODULE COMPLETED** |
| 08.08.2022 | Monday | Mid Review |
| 10.08.2022 | Tuesday | Mid Review |
| 11.08.2022 | Wednesday | Building & Applying Algorithm |
|  | 12.08.2022 | Thursday | Building & Applying Algorithm |

|  |  |  |  |
| --- | --- | --- | --- |
| **10th WEEK** | **DATE** | **DAY** | **NAME OF THE TOPIC/MODULE COMPLETED** |
| 16.08.2022 | Tuesday | Building & Applying Algorithm |
| 17.08.2022 | Wednesday | Building & Applying Algorithm |
| 19.08.2022 | Friday | Building & Applying Algorithm |

|  |  |  |  |
| --- | --- | --- | --- |
| **11th WEEK** | **DATE** | **DAY** | **NAME OF THE TOPIC/MODULE COMPLETED** |
| 22.08.2022 | Monday | Concluding Project |
| 23.08.2022 | Tuesday | Concluding Project |
| 24.08.2022 | Wednesday | Concluding Project |
|  | 25.08.2022 | Thursday | Concluding Project |
|  | 26.08.2022 | Friday | Concluding Project |

|  |  |  |  |
| --- | --- | --- | --- |
| **12th WEEK** | **DATE** | **DAY** | **NAME OF THE TOPIC/MODULE COMPLETED** |
| 29.08.2022 | Monday | Final Review |
| 30.08.2022 | Tuesday | Final Review |
| 01.09.2022 | Wednesday | Final Review |
|  | 02.09.2022 | Thursday | Final Review |
|  | 05.09.2022 | Friday | Final Review |

# INTRODUCTION

Virtual assistant is used to run machine like laptop or PC’s on your own command. Virtual assistant is an application program that understands natural language and voice commands to complete tasks for the users. The 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, open or close any application, send messages to anyone on Whatsapp etc. with verbal commands only. Some other types of Voice Assistant are:

• Intelligent Personal Assistant

• Automated Personal Assistant

• Virtual Digital Assistants

• Chat bot

Nowadays virtual assistant is very useful to human. It makes human life easier like operate PC’s or laptop on only voice command. Virtual assistant is a less time consuming. By using virtual assistant we saves our time and contribute in other works. Virtual assistants are typically cloud-based program that requires internet connected devices. Virtual assistant is the flexibility to contract for just the services they need. For creating virtual assistant for your computer go from basics python. Virtual assistants are task-oriented. Virtual assistant’s ability to understand and perform requests. Virtual assistants is a software that understands verbal and written commands and completes task assigned by clients. Virtual assistants are able to interpret human speech and respond via synthesized voices. There are several voice assistants in market like Siri for apple TV remote, Google Assistant for pixel XL smartphones, Alexa as a smart speaker which is developed by using Raspberry Pi, Microsoft Cortana for windows 10. As like this all virtual assistants we also created a virtual assistant for windows. We use Artificial Intelligence technology for this project. Also use python as a programming language, because python offers a good major libraries. For this software use microphone as input device to receive voice requests from user and speaker as output device to give the output voice. This process is the combination of several different technologies like voice recognition, voice analysis and language processing. Virtual assistant use Natural Processing language to match user text or voice input to executable commands. When a user give a command to personal virtual assistant to perform a task, the natural language is converted the audio signals into digital signals Virtual assistants can provide several services which includes,

• Showing weather condition.

• Scheduling appointment.

• Making travelling arrangements.

• Play music, movies, etc.

• Showing date-time.

• Managing emails.

• Open apps.

## 1.1Module Description:

User:

======

1. Give prompt input
2. Give voice commands

Monitor

=======

1. Recognize speech and convert to text
2. Accept user validation
3. Performs system task
4. Display output

Micrphone

=============

1. Listen to spoken commands.

# SYSTEM ANALYSIS

## Requirement Analysis

## Existing System:

## The system will keep listening for commands and the time for listening is variable which can be changed according to user requirements.

## If the system is not able to gather information from the user input it will keep asking again to repeat till the desired no. of times.

## The system can have both male and female voices according to user requirements.

## Features supported in the current version include playing music, emails, texts, search on Wikipedia, or opening system installed applications, opening anything on the web browser, etc.

## The system will keep listening for commands and the time for listening is variable which can be changed according to user requirements.

## Proposed System

The proposed system is an adaptation of existing open-source voice assistants, with improved functionality, accuracy and security measures. It aims to parallel smaller ranks of Google Assistant in terms of accuracy and Apple’s Siri in terms of security. The project is a standalone application that can integrate with any platform and will be released as an open-source software. Through this application, the customer will be able to perform tasks such as searching online, opening applications and conversing with the application. The application analyzes data quickly and return outputs in a matter of seconds. Every time a command is spoken, it translates the speech into text and displays it on the screen.

The Google Speech Recognition package will be used in Python to convert the speech heard to text. The webpage will be displayed for the user to instruct them to start speaking. Then, the input will be received from the in-built microphone through the Python code and recognized words will be sent to the process. From there, the words will be sent to the webpage to display it on the screen. The user can validate recognized speech conversion to text, by clicking one of the two buttons on the screen. If correct, the words are sent back to the process, which sends it to a new Python child process that will perform the appropriate system tasks based on the words provided. Once it finishes, it will send back data to the process, which will display it once again on the webpage, before refreshing the page to take the next input command. Thus, there is a series of interactions between the main process its two child processes , the webpage (GUI) and the user. There are appropriate instructions, error messages and animations on the display, to engage the user.

The resultant application will be runnable on personal computers of different operating systems and will avoid the general threat of security, since it doesn’t save the user data in any way. The application will depend on certain technologies that have been discussed below in detail**.**

# SOFTWARE REQUIREMENTS SPECIFICATIONS

## System configurations

The software requirement specification can produce at the culmination of the analysis task. The function and performance allocated to software as part of system engineering are refined by established a complete information description, a detailed functional description, a representation of system behavior, and indication of performance and design constrain, appropriate validate criteria, and other information pertinent to requirements.

## Software Requirements:

• Operating system : No restrictions

• Front End : Chromium or Chrome (written in JavaScript)

• Back End : Node.js

• Server : Google Speech Recognition API

## Hardware Requirement:

• Processor : Any processor above 1 GHz

• RAM : 1 GB

• Hard Disk : 10 GB

• Input device : In-built microphone, mouse and keyboard

• Output device : Monitor or display

# 4.TECHNOLOGY

## PACKAGES

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

1) **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.

2) **pyttsx3** — pyttxs3 is a text to speech conversion library in python. This package supports text to speech engines on Mac os x, Windows and on Linux.

3) **wikipedia** — Wikipedia is a multilingual online encyclopedia 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.

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

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

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

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

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

9) **Subprocess** — This is a standard library use to process various system commands like to log off or to restart your PC.

10) **Json**- The json module is used for storing and exchanging data.

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.

## PYTHON

Python is a high-level, interpreted programming language designed by Guido van Rossum. It was written majorly to provide a language that has a simple syntax and is readable. Due to shorter codes and ease of writing, programmers began to increasingly stick to Python for coding. It also has many built-in functions and can operate as object- oriented, functional or procedural programming. It is also platform independent. Since it is free and open source, and also has vast library support, it can be used to perform a huge variety of actions and programmers find it easier to learn and implement compared to other languages. It also has exception handling and in-built memory management techniques. Since it is dynamically typed, there are no declarations, making it compact and concise. The most important part of Python is the indentation as it determines the flow of statements.

Python also contains artificial intelligence and natural language processing libraries, which makes it useful in these fields. It is also used in information security, game development and as the base language for Raspberry Pi. However, compared to C/C++, Python is slightly slower and doesn’t support browsers and mobile devices.

Python comes installed in many Linux and Mac OS computers. If your system doesn’t have Python, the installer can be downloaded from the Download page of python.org website. Running the installer and selecting the option to add to PATH variable will setup the Python files. There are two versions of Python: 2.x and 3.x. There are few minor differences between them like print function, error handling and division operator. There are also multiple ways to run code in Python. One way is by typing “python” in command prompt under the installed folder, which will open the Python shell. Here, one statement at a time can be executed. Another way is to execute a Python file containing a sequence of statements. The third way is to use an IDE (integrated software development) like PyCharms.

There are several important libraries in Python like:

* The “Bokeh” library for data visualization that displays informative charts and plots
* The “Numpy” package for array processing for scientific computing of multidimensional arrays
* The “keras” library which is an open source neural network package
* The “Pandas DataFrame” that is a two-dimensional, tabular data structure with two axes.

It also has multiple methods for implementing GUI (graphical user interface), out of which “tkinter” is one of the most popular. It also has modules for math functions, calendar, time, automation and URL handling.

The following are some of the advantages of Python:

* Presence of third-party modules
* Easy to learn
* Dynamically typed
* Portable and interactive

The following are some of the applications of Python:

* Desktop applications with GUI
* Language development
* Prototyping
* Operating systems

## HTML

A markup language for designing webpages, HTML is a standard for describing the structure of webpages. HTML elements are denoted by tags and define the layout of the webpage. These tags usually come in pairs, with a starting and ending tag for every element. In between the two tags, the content of the element is placed. Within the

starting tag, there can be zero or more attributes specified with the values that control the style and settings of the element. Web browsers, like Chrome, are designed to read the HTML documents and render or display the webpage according to the user’s specifications. The latest version of HTML is HTML5, which was released on 2014. There are no installations required for HTML. The code can be written into any text editor like Notepad and saved with the “.htm” or “.html” extension to be able to run it as an HTML document. Every document begins with the <html> tag and ends with the </html> tag. It may or may not contain the doctype specification at the top.

HTML forms are also available to collect user input through various in-built GUI elements. Other than this, there is also a graphics tag called <canvas> and a multimedia tag called <media> for audio and video. There is a geolocation API for detecting user location. HTML also has an alternative for cookies in the form of web storage, which is more secure and performs better even with large amounts of data. There are various tags for designing tables, frames and blockquotes.

## JAVASCRIPT

JavaScript is an interpreter, client-side, event-based, object oriented scripting language that you can use to add dynamic interactivity to your web pages.

JavaScript scripts are written in plain text, like HTML, XML, Java, PHP and just about any other modern computer code. In this code, we will use Windows Note Pad to create and edit our JavaScript code, but there are a large number of alternatives available.

Note Pad is chosen to demonstrate JavaScript’s immediacy and simplicity.

## You can use JavaScript to achieve any of the following:

* Create special effects with images that give the impression that a button is either highlighted or depressed whenever the mouse pointer is hovered over it.
* Validate information that users enter into your web forms
* Open pages in new windows, and customise the appearance of those new windows.
* Detect the capabilities of the user’s browser and alter your page’s content appropriately.
* Create custom pages “on the fly” without the need for a server-side language like PHP. JavaScript is not Java, though if you come from a Java background, you will notice that both languages look similar when written. Java is a full featured and comprehensive programming language similar to C or C++, and although JavaScript can interact with Java web applications, the two should not be confused.

Different web browsers will run your JavaScript in different, sometimes incompatible ways. In order to work around this, it is often necessary to use JavaScript itself to detect the capabilities of the browser in which it finds itself, and alter its operation depending on the result.

## CSS

CSS, or Cascading Style Sheets, is used to describe the style of HTML documents. Since a single CSS file can be used to style more than one HTML document at a time. HTML was not supposed to have styling tags and attributes; it was designed at first with the sole purpose of describing the basic structure and layout of the webpage. But in HTML version 3.2, tags like <font> were added, causing web developers to get confused between the layout and style of the page. Soon after this, CSS was introduced to define a way to separate all the styling aspects of the page. Using CSS, it is possible to define completely different styles and images for different variations of screen size.

CSS defines the look and feel of a web page. Using CSS, you can control:

* + - * The style of fonts
      * The color of the text
      * What background images or colors are used
      * The spacing between paragraphs
      * Layout designs

* + - * How columns are sized and laid out
      * Variations in display for different devices and screen sizes

CSS is easy to understand and easy to learn, but it has powerful control over the appearance of an HTML document. Usually, CSS is combined with HTML or XHTML. It follows global web standards and applies for many devices, hence providing a single point of maintenance.

## JSON

JSON or JavaScript Object Notation is a syntax for exchange and storage of data. When there is data exchange between browser and server, the data can only be in text form. JSON is text. Using JSON, we can convert any JavaScript object into JSON, and send it to the server. Also, we can convert JSON received from the server into JavaScript objects. Hence, we can work with the JavaScript objects, without complex parsing and translations.

JSON is a lightweight, data exchange format. It is easy to understand and language independent. It uses the same syntax as JavaScript. In Electron apps, the starting point of execution is the package file, which is of JSON type. The package file describes which script to execute and runs it accordingly. The main script also has statements that convert the JSON to string format before sending it to the renderer Python files. JSON is a text format that is completely language independent but uses conventions that are familiar to programmers of the C-family of languages. It is easy for machines to parse and generate.

# CODING

from platform import uname

import time

import pyttsx3 # pip install pyttsx3

import speech\_recognition as sr # pip install speechRecognition, pip install pipwin & pipwin install pyaudio

import datetime # pip install datetime

import wikipedia # pip install wikipedia

import webbrowser

import os

import smtplib

import pyjokes # pip install pyjokes

# initialises the pyttsx3 modules

# Microsoft Speech API (SAPI5) is the technology for voice recognition & synthesis, provided by Microsoft

engine = pyttsx3.init('sapi5')

# it'll get the voice property frm the pyttsx3 module

voices = engine.getProperty('voices')

# it'll get list of voices & v can select 1 which v lyk

# voices[0].id -> Male voice

# voices[1].id -> Female voice

engine.setProperty('voice', voices[0].id)

# now vll set the voice speed

# by default it will b 200 words per min

engine.setProperty('rate', 190) # 190 words per min

ass\_name = "Jarvis"

def speak(audio):

engine.say(audio) # convert the written text in2 speach

engine.runAndWait() # pauses the program till the say function is done wyt speaking

def wishMe():

speak("Welcome back!")

hour = int(datetime.datetime.now().hour)

if hour >= 2 and hour < 12:

speak("Good Morning!")

elif hour >= 12 and hour < 17:

speak("Good Afternoon!")

elif hour >= 17 and hour < 23:

speak("Good Evening!")

else:

speak("Good Night!")

speak("I'm your Assistant")

speak(ass\_name)

speak("AI at your service")

def takeCommand():

# It takes microphone input frm the user & returns string output

r = sr.Recognizer() # v r initializing the Recognizer in the r variable

with sr.Microphone() as source: # get input frm the user frm the microphone, it will our source for our input

print("Listening...")

speak("Listening...")

r.pause\_threshold = 2 # it will wait for 2 sec before it strt 2 listen

audio = r.listen(source) # microphone will listen 2 our audio, v hv passed source variable in listen function

try:

print("Recognizing...")

query = r.recognize\_google(audio, language = 'en-in')

print(f"User said: {query}\n")

except Exception as e:

# print(e)

print("Could not understand audio...")

speak("Could not understand audio...")

print("Say that again please...")

speak("Say that again please...")

return "None"

return query

def sendEmail(to, content):

server = smtplib.SMTP('smtp.gmail.com', 587)

server.ehlo()

server.starttls()

# enable low security in gmail

server.login('your emai id', 'your email password')

server.sendmail('your email id', to, content)

server.close()

if \_\_name\_\_ == "\_\_main\_\_":

wishMe()

print('What should I call you ?')

speak('What should I call you ?')

uname = takeCommand()

print('Welcome', uname)

speak('Welcome')

speak(uname)

speak('Please tell me how may I help you')

while True:

# if 1:

query = takeCommand().lower()

# logic for executing tasks based on query

if 'wikipedia' in query:

print('Searching Wikipedia...')

speak('Searching Wikipedia...')

results = wikipedia.summary(query.replace("wikipedia", ""), sentences = 2)

speak("According to Wikipedia")

print(results)

speak(results)

elif 'open youtube' in query:

print('Here you go to YouTube\n')

speak('Here you go to YouTube\n')

webbrowser.open("youtube.com")

elif 'open google' in query:

print('Here you go Google\n')

speak('Here you go Google\n')

webbrowser.open("google.com")

elif 'open stackoverflow' in query:

print('Here you go Stackoverflow\n')

speak('Here you go Stackoverflow\n')

webbrowser.open("stackoverflow.com")

elif 'open github' in query:

print('Here you go Github\n')

speak('Here you go Github\n')

webbrowser.open("github.com")

elif 'open dev community' in query:

print('Here you go Dev Community\n')

speak('Here you go Dev Commmunity\n')

webbrowser.open("dev.to")

elif 'open kaggle' in query:

print('Here you go Dev Kaggle\n')

speak('Here you go Dev Kaggle\n')

webbrowser.open("kaggle.com")

elif 'open protonmail' in query:

print('Here you go Proton Mail\n')

speak('Here you go Proton Mail\n')

webbrowser.open("mail.protonmail.com")

elif 'open brave' in query:

print('Here you go Brave Browser\n')

speak('Here you go Brave Browser\n')

codePath = "C:\\Program Files\\BraveSoftware\\Brave-Browser\\Application\\brave.exe" # ur brave browser directory

os.startfile(codePath)

elif 'open firefox' in query:

print('Here you go Firefox Browser\n')

speak('Here you go Firefox Browser\n')

codePath = r"C:\\Program Files\\Mozilla Firefox\\firefox.exe" # ur firefox browser directory

os.startfile(codePath)

elif 'open chrome' in query:

print('Here you go Chrome Browser\n')

speak('Here you go Chrome Browser\n')

codePath = r"C:\\ProgramData\\Microsoft\\Windows\\Start Menu\\Programs\\Google Chrome.lnk" # ur chrome browser directory

os.startfile(codePath)

elif 'open whatsapp' in query:

print('Here you go WhatsApp\n')

speak('Here you go WhatsApp\n')

codePath = r"C:\\Users\\Rahul\\AppData\\Roaming\\Microsoft\\Windows\\Start Menu\\Programs\\WhatsApp\\WhatsApp.lnk" # ur whatsapp directory

os.startfile(codePath)

elif 'open telegram' in query:

print('Here you go Telegram\n')

speak('Here you go Telegram\n')

codePath = r"C:\\Users\\Rahul\\AppData\\Roaming\\Microsoft\\Windows\\Start Menu\\Programs\\Telegram Desktop\\Telegram.lnk" # ur telegram directory

os.startfile(codePath)

elif 'open jetbrains toolbox' in query:

print('Here you go Jetbrains Toolbox\n')

speak('Here you go Jetbrains Toolbox\n')

codePath = r"C:\\Users\\Rahul\\AppData\\Roaming\\Microsoft\\Windows\\Start Menu\\Programs\\JetBrains Toolbox\\JetBrains Toolbox.lnk" # ur jetbrains toolbox directory

os.startfile(codePath)

elif 'open vs code' in query or 'visual studio code' in query:

print('Here you go VS Code\n')

speak('Here you go VS Code\n')

codePath = r"C:\\Users\\Rahul\\AppData\\Roaming\\Microsoft\\Windows\\Start Menu\\Programs\\Visual Studio Code\\Visual Studio Code.lnk" # ur vs code directory

os.startfile(codePath)

elif 'open code' in query:

print('Here you go Project Code\n')

speak('Here you go Project Code\n')

codePath = r"D:\\Projects\\Python Projects\\Voice\_Assistant.py" # ur project directory

os.startfile(codePath)

elif 'play music' in query:

music\_dir = 'D:\\Non Critical\\songs\\Favorite Songs2' # ur music directory

songs = os.listdir(music\_dir)

print(songs)

os.startfile(os.path.join(music\_dir, songs[0]))

elif 'send a mail to' in query:

try:

speak("What should I say ?")

content = takeCommand()

speak("whom should i send")

to = input()

sendEmail(to, content)

speak("Email has been sent!")

except Exception as e:

print(e)

speak("Sorry my friend, I'm not able to send this email")

elif 'the time' in query:

strTime = datetime.datetime.now().strftime("%H:%M:%S")

speak(f"Sir, the time now is {strTime}")

elif 'how are you' in query:

print("I'm fine, Thank you")

speak("I'm fine, Thank you")

print("How are you,", uname)

speak("How are you,")

speak(uname)

elif 'fine' in query or 'good' in query:

print("It's good to know that you are fine")

speak("It's good to know that you are fine")

elif 'who i am' in query or 'who am i' in query:

print('If you talk then definitely your human and your name is', uname)

speak('If you talk then definitely your human and your name is')

speak(uname)

elif 'change my name' in query:

speak('What would you like me to call you ?')

print('What would you like me to call you ?')

uname = takeCommand()

speak('Your name is changed to')

speak(uname)

print('Your name is changed to', uname)

elif 'change your name' in query:

print('What would you like to call me,', uname)

speak('What would you like to call me,')

speak(uname)

ass\_name = takeCommand()

print('Thanks for naming me as', ass\_name)

speak('Thanks for naming me as')

speak(ass\_name)

elif "what's your name" in query or 'what is your name' in query:

print('My friends call me', ass\_name)

speak('My friends call me')

speak(ass\_name)

elif 'what is my name' in query or "what's my name" in query:

print("Your name is ", uname)

speak('Your name is')

speak(uname)

elif 'who made you' in query or 'who created you' in query:

print('I have been created by Rahul')

speak('I have been created by Rahul')

elif 'exit' in query or 'quit' in query:

print('Thanks', uname,'for giving me your time')

speak('Thanks')

speak(uname)

speak('for giving me your time')

exit()

elif 'joke' in query:

speak(pyjokes.get\_joke())

elif 'why you came to earth' in query or 'why did you come to earth' in query:

print("Thanks to Rahul, further It's a secret")

speak("Thanks to Rahul, further It's a secret")

elif 'why have you been created' in query or 'reason for you' in query or 'reason for your creation' in query:

print('I was created as a project by Mister Rahul')

speak('I was created as a project by Mister Rahul')

elif 'where is' in query:

location = query.replace('where is', '')

print(uname, 'asked to locate', location)

speak(uname)

speak('asked to locate')

speak(location)

webbrowser.open("https://www.google.nl/maps/place/" + location + "")

elif "write a note" in query:

print('What should I write', uname)

speak('What should I write')

speak(uname)

note = takeCommand()

file = open('jarvis.txt', 'w')

print(uname, 'Should I include date and time')

speak(uname)

speak("Should I include date and time")

snfm = takeCommand()

if 'yes' in snfm or 'sure' in snfm or 'ok' in snfm:

strTime = datetime.datetime.now().strftime("%H:%M:%S")

file.write(strTime)

file.write(" :- ")

file.write(note)

else:

file.write(note)

elif "show note" in query or 'open note' in query:

print("Showing Notes")

speak("Showing Notes")

file = open("jarvis.txt", "r")

print(file.read())

speak(file.read(6))

elif "don't listen" in query or "stop listening" in query:

speak("for how much time you want to stop jarvis from listening commands")

sleep\_time = int(takeCommand())

time.sleep(sleep\_time)

print(sleep\_time)

# most asked question frm google Assistant

elif "will you be my gf" in query or "will you be my bf" in query:

print("I'm not sure about, may be you should give me some time")

speak("I'm not sure about, may be you should give me some time")

elif "i love you" in query:

print("It's hard to understand")

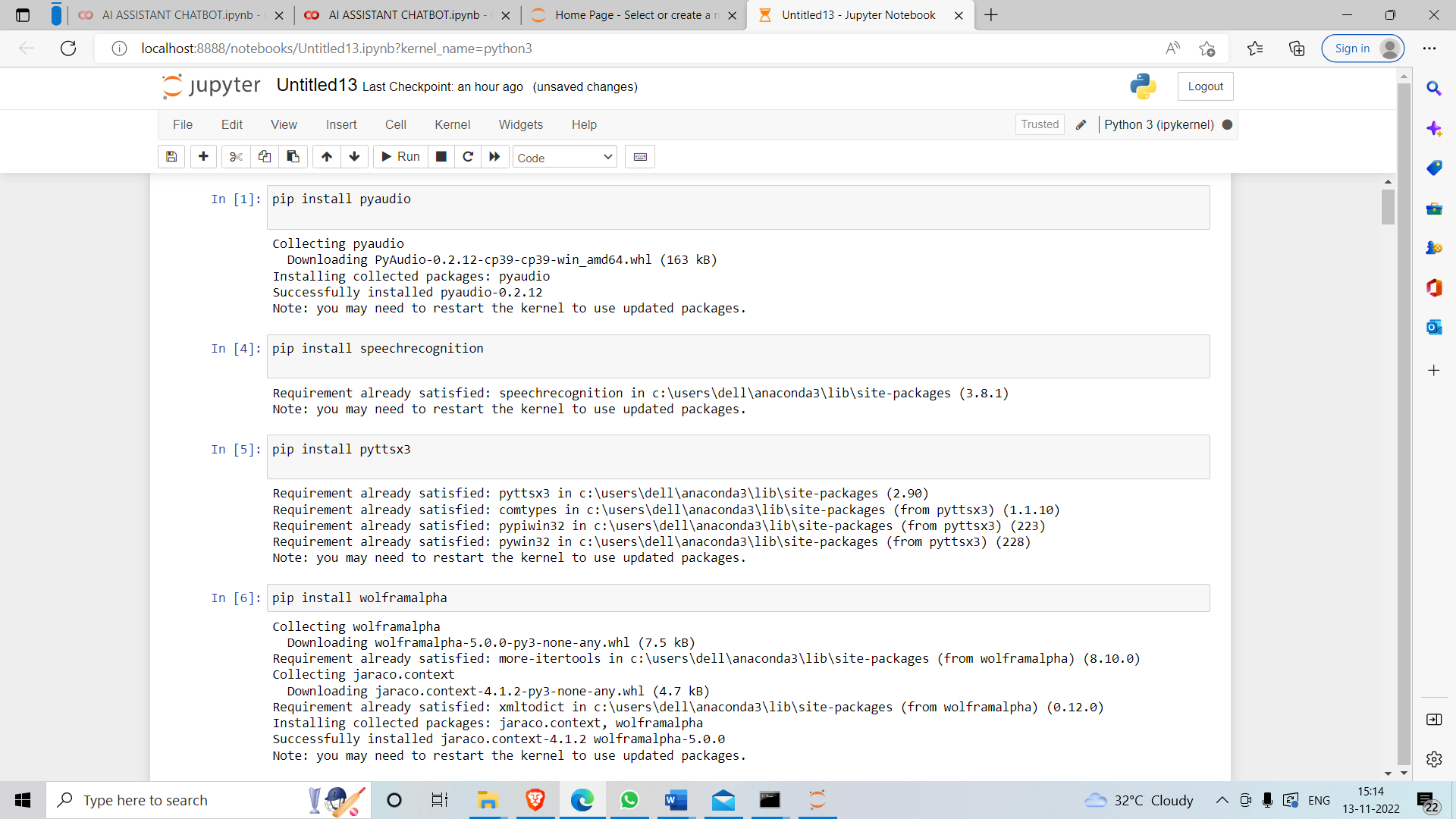
speak("It's hard to understand")

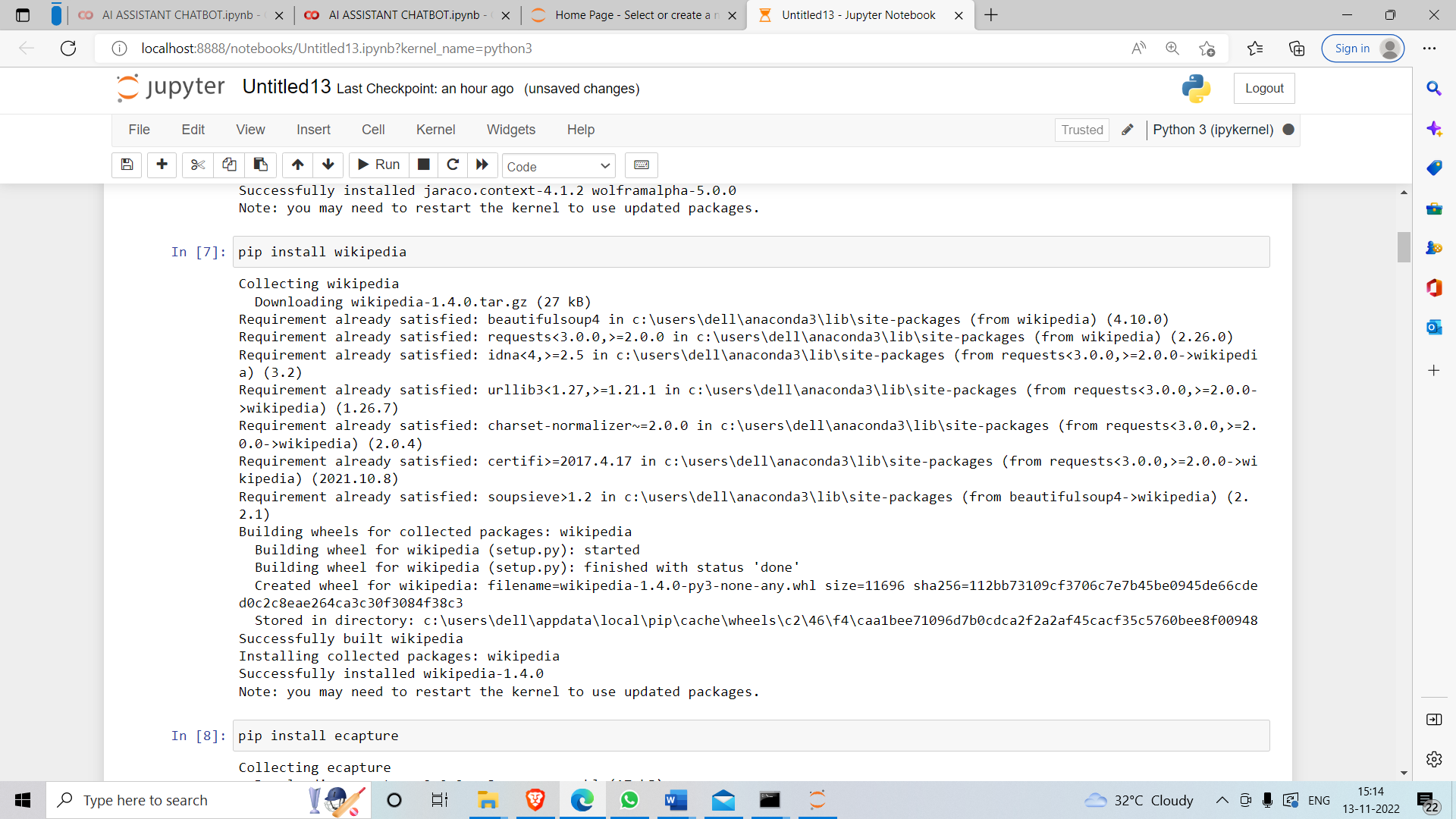
# elif "" in query:

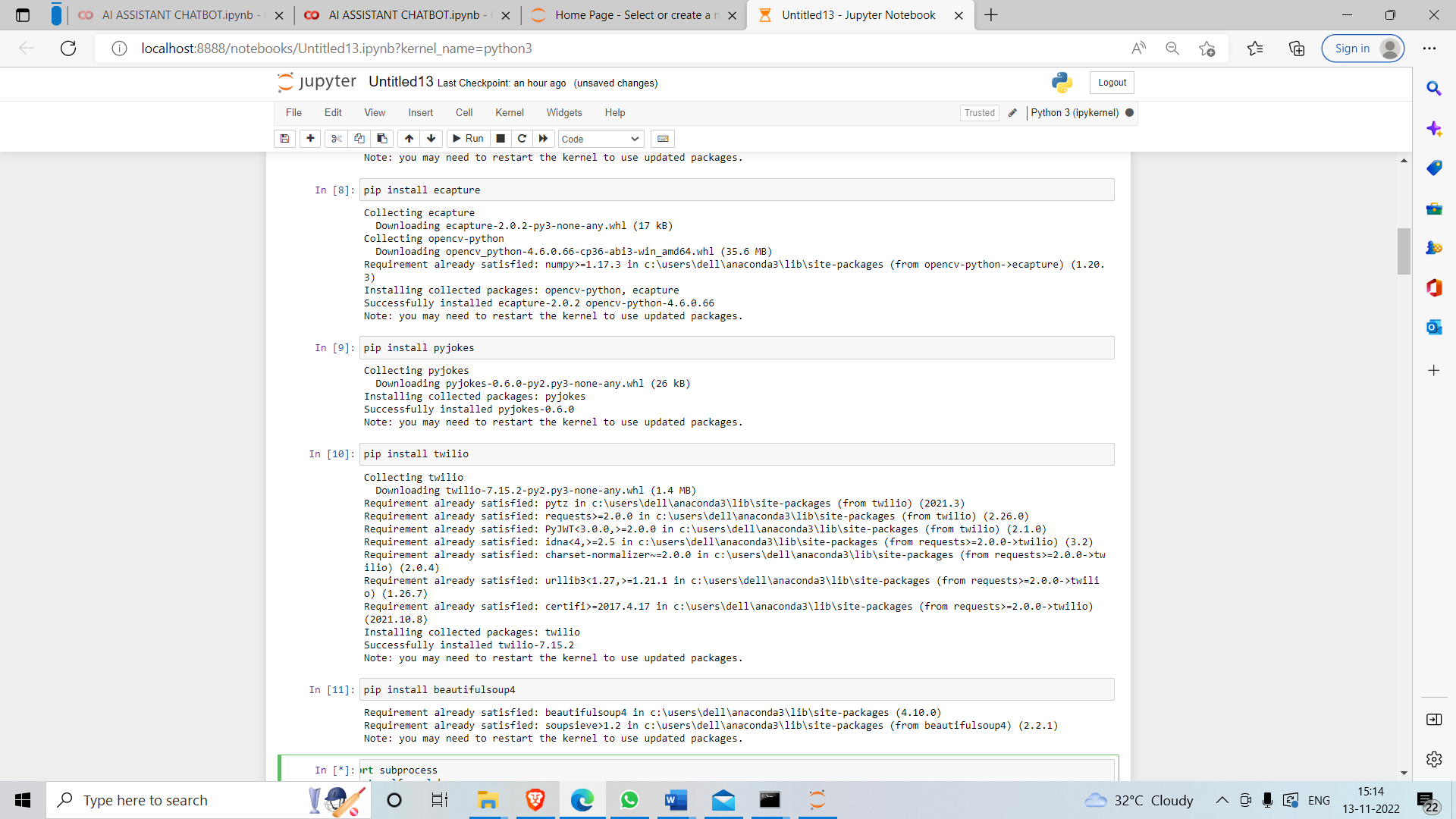
# command go here

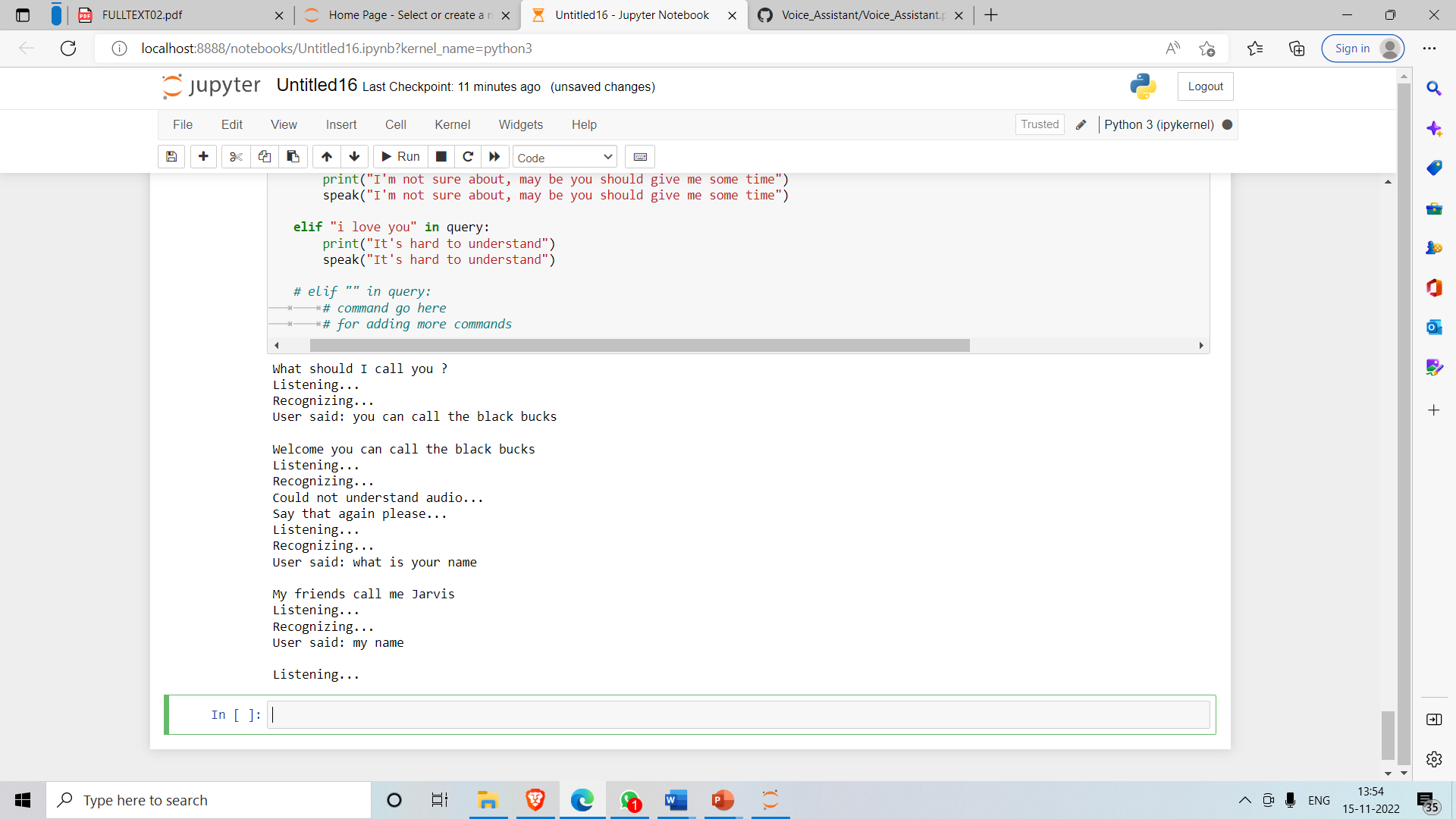
# for adding more commands

# SCREENSHOTS









# CONCLUSION

The final outcome of this project is an intelligent voice assistant, as described in the title. It combines natural language processing techniques with web programming to present an efficient digital personal assistant. It can give answers for some of the user questions.It can perform everyday tasks on the system based on the spoken command of the user. It can recognize the words, map the speech into text and decide what task to execute accordingly. The application can perform operations on the device such as open applications like settings, calculator, Microsoft Word, media players, etc. It can also change volume settings, open websites, perform online searches and minimize all open windows on the computer.

# BIBLOGRAPHY

[1]. CortanaIntelligence,GoogleAssistant,AppleSiri

[2]. <https://data-flair.training/blogs/artificial-intelligence-project-ideas/>

[3]. https://www.upgrad.com/blog/top-artificial-intelligence-project-ideas-topics-for-beginners/

[4]. <https://www.activestate.com/blog/how-to-build-a-digital-virtual-assistant-in-python/>

[5]. https://towardsdatascience.com/how-to-build-your-own-ai-personal-assistant-using-pythonf57247b4494b Research Paper on Desktop Voice Assistant <https://www.section.io/engineering-education/creating-a-virtual-assistant-using-python/>

[6]. <https://medium.com/codex/making-your-own-ai-virtual-assistant-with-python-5c2046dadfa7>

## WEBLINKS:

[1]. <https://www.codewithharry.com/>

[2]. [www.stackoverflow.com](http://www.stackoverflow.com)

[3]. [www.pythonprogramming.net](http://www.pythonprogramming.net)

[4]. [www.codecademy.com](http://www.codecademy.com)

[5]. [www.tutorialspoint.com](http://www.tutorialspoint.com)

[6]. [www.google.co.in](http://www.google.co.in)

[7]. [www.python.org](http://www.python.org)