

AI DESTOP ASSISTANT

PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE AWARD OF DEGREE OF MASTER OF COMPUTER APPLICATIONS

SESSION 2022-24



योग: कर्मसु कौशलम्
IN PURSUIT OF PERFECTION

VIVEKANANDA SCHOOL OF INFORMATION TECHNOLOGY

VIVEKANANDA INSTITUTE OF PROFESSIONAL STUDIES-TC

Affiliated to

Guru Gobind Singh Indraprastha University

Sector-16, C- Dwarka, Delhi-110078



Under The Guidance Of:

Dr. Dheeraj Malhotra

Associate Professor – VSIT, VIPS

Submitted By:

Ritoshree Sarkar

(07817704422)

ACKNOWLEDGEMENT

It is our proud privilege to express our profound gratitude to the entire management of Vivekananda Institute of Professional Studies and teachers of the institute for providing us with the opportunity to avail the excellent facilities and infrastructure. The knowledge and values inculcated have proved to be of immense help at the very start of my career. Special thanks to Hon'ble Founder, Vivekananda Institute of Professional Studies, Delhi for having provided us an excellent infrastructure at VSIT.

I am grateful to **Dr. Supriya Madan (Dean, VSIT)** and **Dr. Dheeraj Malhotra** for their astute guidance, constant encouragement and sincere support for this project work.

Sincere thanks to all my family members, seniors and friends for their support and assistance throughout the project.

Ritoshree Sarkar

CERTIFICATE

This is to certify that **Ritoshree Sarkar** from Vivekananda Institute of Professional Studies (Technical Campus), Delhi has presented this project work entitled “AI DESKTOP ASSISTANT”, a speech recognition system working as per your command in partial fulfilment of the requirements for the award of the degree of Masters of Computer Applications (MCA) under our supervision and guidance.

Dr. Dheeraj Malhotra

Associate Professor – VSIT, VIPS

SYNOPSIS

TITLE- AI DESKTOP ASSISTANT

PROBLEM STATEMENT

Design and develop a desktop AI assistant that can effectively understand and respond to user commands, provide accurate information, perform tasks efficiently, and enhance productivity, thereby improving the user's overall desktop experience.

PROPOSED SYSTEM

The project aims to develop a personal-assistant for Linux-based systems. Jarvis draws its inspiration from virtual assistants like Cortana for Windows, and Siri for iOS. It has been designed to provide a user-friendly interface for carrying out a variety of tasks by employing certain well-defined commands. Users can interact with the assistant either through voice commands.

OBJECTIVE

This project will serve the following objectives: -

1. Provides the user with easy and friendly interface
2. Provides speech recognition
3. Provides tasks commonly used on the desktop
4. Provides faster access to different functions
5. Provides an interactive experience

LIMITATIONS

1. Limited functionality
2. Security concerns- Multiple open source libraries used

CONTENTS

<u>1</u>	INTRODUCTION	1
	1.1Objective and Scope of the System	
	1.2Theoretical Background	
	1.3Advantage of the system	
	1.4Definition of Problem	
<u>2</u>	SYSTEM ANALYSIS &DESIGN	5
	2.1Analysis Study	
	2.2User Requirements	
	2.3Final Requirements	
3	SYSTEM PLANNING (PERT Chart)	10
4	DESIGN OF THE SYSTEM	11
	4.1Methodology	
	4.2System requirements	
	4.3System Implementation	
	4.4Details of Hardware and Software	
5	COST & BENEFIT ANALYSIS	17
6	DETAILED LIFECYCLE OF THE PROJECT	19
	6.1 Use Case Diagram	
	6.2 DFD Diagram	
	6.3 Flow Chart	
	6.4 Input & Output Screen Design	
	6.5 Methodology used for testing	
7	CODING	33
8	CONCLUSION	38
9	BIBLIOGRAPHY	40

Chapter 1

INTRODUCTION

AI Desktop assistant is a virtual helper that utilizes speech recognition technology to understand and respond to user commands and queries. This assistant can be integrated into desktop computers or laptops, providing users with a hands-free and convenient way to interact with their devices.

The project aims to develop a private assistant for your computer. It provides a user-friendly interface for completing a spread of tasks by employing certain well-defined commands. Users can interact with the assistant through voice commands. As a private assistant, it assists the end-user with day-to-day activities through voice commands. The software uses a device's microphone to receive voice requests while the output takes place at the system's speaker. It's a mixture of various technologies: voice recognition, voice analysis, and language processing.

1.1 OBJECTIVE OF THE SYSTEM

This project will serve the following objectives: -

1. Provides the user with easy and friendly interface
2. Provides speech recognition
3. Provides tasks commonly used on the desktop
4. Provides faster access to different functions
5. Provides an interactive experience

1.2 THEORETICAL BACKGROUND OF THE SYSTEM

Artificial Intelligence when used with machines, it shows us the capability of thinking like humans. In this, a computer system is designed in such a way that typically requires interaction from human. As we know Python is an emerging language so it becomes easy to write a script for Voice Assistant in Python. The instructions for the assistant can be handled as per the requirement of user. Speech recognition is the Alexa, Siri, etc. In Python there is an API called Speech Recognition which allows us to convert speech into text. It was an interesting task to make my own assistant. It became easier to search on Google without opening the browser, and performing many other daily tasks like playing music, opening your favourite IDE with the help of a single voice command. In the current scenario, advancement in technologies are such that they can perform any task with same effectiveness or can say more effectively than us. By making this project, I realized that the concept of AI in every field is decreasing human effort and saving time. As the voice assistant is using Artificial Intelligence hence the result that it is providing are highly accurate and efficient. The assistant can help to reduce human effort and consumes time while performing any task, they removed the concept of typing completely and behave as another individual to whom we are talking and asking to perform task. The assistant is no less than a human assistant but we can say that this is more effective and efficient to perform any task. The libraries and packages used to make this assistant focuses on the time complexities and reduces time.

FUNCTIONALITIES OF THE SYSTEM:

1. Checking the time
2. Opening the calculator
3. Opening Google and searching for a specific topic

4. Opening YouTube and searching for a video
5. Opening Wikipedia for a specific topic
6. Listening to music
7. Checking the latest news
8. Checking the weather
9. Opening the camera

1.3 ADVANTAGES OF THE SYSTEM

1. **Easiness in modification of system:** The proposed system provides managing of services efficient. It is easily possible to add more services without effecting the current system
2. **User friendly:** The proposed system is user friendly because the accessing of data is fast and efficient. It can perform a wide range of tasks. It can answer questions, provide information, execute commands, and perform various functions based on user input.
3. **User Feedback:** The users of the system are entitled to feedback about the speech recognition accuracy
4. **No or very few paperwork:** The proposed system either does not require paper work or very few paper works is required. All the data is feted into the computer immediately and various feedbacks and reports can be generated through computers. Since all the data is kept in a database no data of the organization can be destroyed. Moreover work becomes very easy because there is no need to keep data on papers.
5. **Support strategic competitive advantage:** Proposed system supports strategic competitive advantages. Since the proposed systems provide easiness in reports generating it will provide strategic advantages among competitors.

1.4 DEFINITION OF PROBLEM

Develop an AI powered desktop assistant that utilizes speech recognition technology to accurately transcribe and interpret user speech recognition technology to accurately transcribe and interpret user speech commands or queries. The assistant should be able to understand and process natural language input in real-time, allowing users to interact with their computer tasks using voice commands.

The problem involves training the AI model to recognize and understand a wide range of spoken words, phrases, and contextual cues, while also accurately transcribing the speech into text format. It requires building a robust speech recognition system that can handle variations in pronunciations, accents, and background noise, ensuring a high level of accuracy in understanding user commands.

Furthermore, the assistant should be equipped with the capability to perform various tasks based on the recognised voice commands, such as opening applications, navigating the file system, setting reminders, scheduling events, performing web searches, and providing relevant information or responses.

The overall goal is to create a reliable and efficient AI Desktop Assistant that enhances user productivity and convenience by enabling hands-free operation and intuitive through speech recognition.

Chapter 2

SYSTEM ANALYSIS AND DESIGN

2.1 ANALYSIS STUDY

1. Open Communication Environment:

The integration of AI and speech recognition brings numerous benefits to users. It provides a natural and intuitive way to interact with computers, as users can simply speak their commands rather than typing or clicking. This can be particularly helpful for individuals with mobility impairments or those who prefer a hands free approach.

2. Increase in productivity:

AI Desktop assistant with speech recognition can enhance productivity by streamlining tasks. Users can quickly dictate the tasks or perform complex actions by simply speaking to their assistant. This not only saves time but also reduces the cognitive load required for traditional keyboard and mouse interactions

3. Secured and Reliable:

The system provides results with as much accuracy possible in the least amount of time. It is a secure system with end to end functioning as per the services required. AI assistant use secure communications between the assistant and external services. This ensures that interactions and data transfers are protected from eavesdropping or tampering.

4. Regular updates:

Developers regularly update AI assistants to address security vulnerabilities and improve reliability. Patches and updates are released to fix any identified issues, ensuring the assistant remains secure against emerging threats

5. Platform Independent

This refers to the ability of the AI Desktop assistant to function across different operating systems and hardware platforms without requiring significant modifications. While achieving complete platform independence can be challenging due to the variations in software and hardware architectures, efforts are made to ensure compatibility and portability.

2.1.1 Feasibility Study

All projects are feasible if they have unlimited resources and infinite time. But the development of software is plagued by the scarcity of resources and difficult delivery rates. It is necessary and prudent to evaluate the feasibility of a project at the earliest possible time. The three considerations are involved in the feasibility analysis.

2.1.2 Technical Feasibility

Technical feasibility centres on the existing computer (hardware, software...etc) and to what extent it can support the proposed addition if the budget is a serious constraint, then the project is judged not feasible. This project requires an internet connection and a web browser application so its technically feasible.

2.1.3 Economical Feasibility

AI Desktop Assistant is economically feasible since it is not using any proprietary software and doesn't include any additional charges.

2.1.4 Operational Feasibility

People are inherently resistant to change and computers provide users with the ease of being able to stay in the environment they are compatible in. There is no need of a technical background to work on the application. All the information needed can be seen on just one click.

2.2 USER REQUIREMENTS

User requirements for an AI desktop assistant can vary depending on individual needs and preferences. However, here are some common user requirements that AI desktop assistants aim to fulfil:

1. **Voice Interaction:** Users often require the ability to interact with the AI assistant through voice commands. This includes tasks such as asking questions, giving instructions, or initiating actions using natural language.
2. **Task Automation:** Users may want the AI assistant to automate repetitive tasks or perform complex actions on their behalf. This could include tasks like scheduling appointments, sending emails, setting reminders, or managing files.
3. **Information Retrieval:** Users often rely on AI assistants to provide quick and accurate information on various topics. This could involve web searches, retrieving weather updates, looking up facts, or providing answers to specific questions.
4. **Personalization:** Users may have preferences for customizing the AI assistant's behaviour or appearance. This could include choosing a preferred voice, adjusting language settings, or customizing the assistant's responses to align with their individual needs.

5. Integration with Other Applications and Services: Users often require their AI assistant to seamlessly integrate with other applications and services on their desktop. This could involve integrating with calendar apps, productivity tools, or home automation systems to enable streamlined workflows.

6. Security and Privacy: Users expect their AI assistant to prioritize security and protect their personal information. This includes implementing strong authentication methods, encrypting data, and adhering to privacy regulations to ensure the confidentiality and integrity of user data.

7. Reliability and Performance: Users require AI desktop assistants to be reliable and responsive, delivering accurate results and fast response times. The assistant should be able to handle user queries and commands effectively, even under varying network conditions or system loads.

8. Continuous Improvement: Users often appreciate AI assistants that continuously learn and improve over time. This involves incorporating user feedback, adapting to user preferences, and refining the assistant's capabilities through regular updates and enhancements.

By considering these user requirements, AI desktop assistants can provide a personalized and effective user experience, assisting users in various tasks and making their desktop computing more efficient and enjoyable.

Inquiry Form/Interviews

Q) What kind of system?

A) A fast and reliable system

Q) User satisfaction from the content?

The content is to be self explanatory and user could easily understand it

Q) Data security?

Data should be in encrypted format so that no third party could access the user credentials.

Q) Using of system

A system that is easy to use and with less complications.

Q) Trustworthiness of the user credentials?

Since the user is providing of its credentials to the application so its rights to use needs to be of the user i.e. without user permissions it cannot be exchanged or changed

2.3 FINAL REQUIREMENTS

User Oriented: A system should be more user friendly not of the technical point of view

Better GUI: All the elements used in the system should be of interactive in nature that is its look and feel is not so boring that the user could get bored while using it.

Reliability: The system should be reliable and fast in processing

Data security: Access to the organizational data is not to be granted to any unknown person which is not a part of the transaction

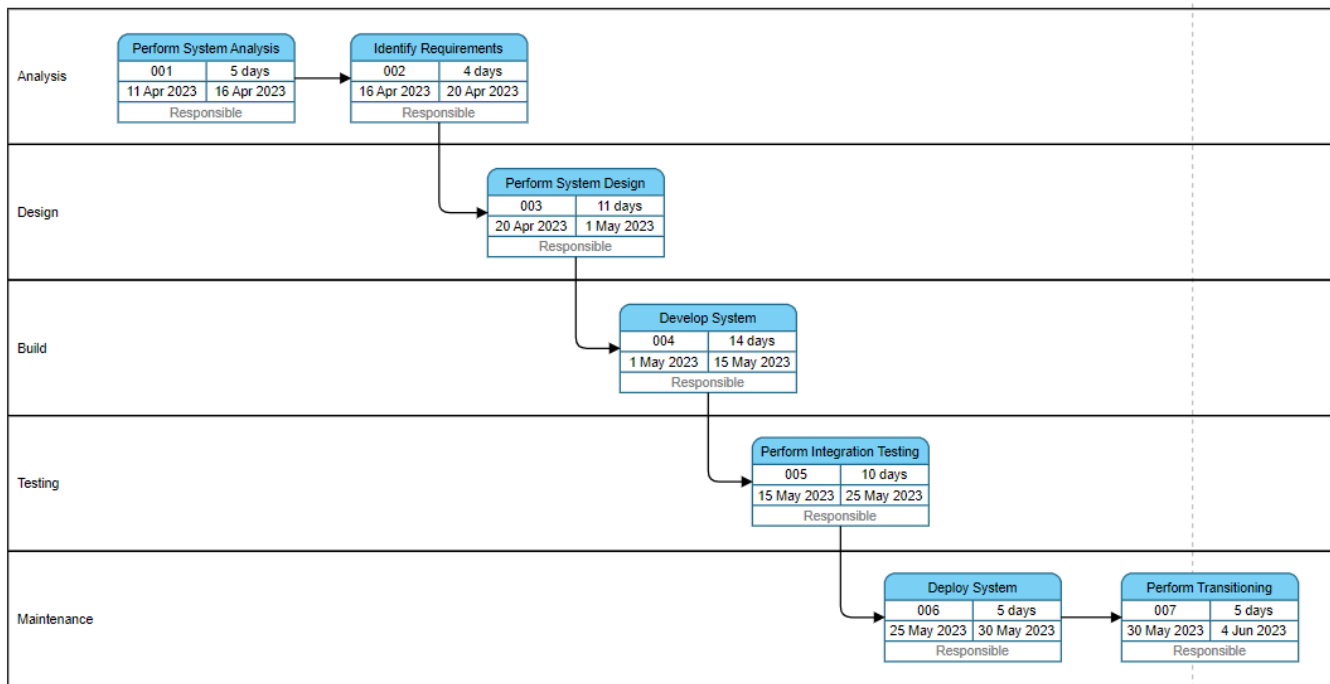
Presentation: The content that is to be presented to the user is to be presented in such a way that is self-explanatory to the user and he/she is satisfied with the data.

Chapter 3

SYSTEM PLANNING

PERT CHART

A PERT chart, also known as a PERT diagram, is a tool used to schedule, organize, and map out tasks within a project. PERT stands for program evaluation and review technique. It provides a visual representation of a project's timeline and breaks down individual tasks.



Chapter 4

DESIGN OF THE SYSTEM

4.1 METHODOLOGY

The methodology for coding and implementing an AI Desktop Assistant using speech recognition involves several steps. Here's a high-level overview of the process:

1. **Choose a Programming Language and Framework:** Select a programming language and framework that support speech recognition and natural language processing capabilities. Popular choices include Python and libraries such as SpeechRecognition, Google Cloud Speech-to-Text, or Mozilla DeepSpeech.
2. **Set Up the Development Environment:** Install the necessary dependencies and tools required for speech recognition and development. This may include installing the chosen speech recognition library, a text-to-speech library for generating responses, and any other additional libraries or packages.
3. **Implement Speech Recognition Functionality:** Write code to capture audio input from the microphone or audio file. Utilize the selected speech recognition library or API to process the audio and convert it into text. Handle any necessary configurations, such as specifying language models, recognition options, or adjusting parameters for noise cancellation or speech enhancement.
4. **Natural Language Understanding (NLU):** Implement a natural language understanding module to extract meaning from the transcribed text. This can involve using libraries such as spaCy, NLTK, or building custom models to perform tasks like part-of-speech tagging, named entity recognition, or sentiment analysis.
5. **Task Execution:** Based on the understood intent from the NLU module, implement the logic to execute tasks or actions requested by the user. This may involve integrating with

existing APIs or libraries to perform actions like opening applications, retrieving information, or interacting with other software systems.

6. Response Generation: Once the task has been executed or the necessary information has been retrieved, generate a response in text format. Utilize a text-to-speech library or service to convert the response into spoken audio.

7. User Interaction Loop: Design a loop that continuously listens for user input, performs speech recognition, and processes the user's request. Handle scenarios such as providing appropriate error handling or fallback responses when the speech is not recognized or the user's intent cannot be understood.

8. User Interface (UI): Implement a user interface component that allows users to interact with the AI Desktop Assistant visually, displaying the transcribed text, response, and any relevant information. This can be a command-line interface, a graphical user interface (GUI), or a web-based interface.

9. Error Handling and Validation: Implement error handling mechanisms to handle unexpected or erroneous user input. Validate user commands or queries to ensure they adhere to expected formats or constraints.

10. Testing and Debugging: Perform thorough testing of the AI Desktop Assistant system, including different speech input scenarios, edge cases, and potential failure points. Debug any issues or errors encountered during testing and ensure the system functions as expected.

11. Deployment and Integration: Prepare the AI Desktop Assistant system for deployment by packaging the code, dependencies, and any necessary configurations. Integrate the assistant with the desktop environment, ensuring it can run seamlessly and interact with other software applications or services.

12. Continuous Improvement and Maintenance: Monitor the assistant's performance, collect user feedback, and make iterative improvements based on user needs and identified issues. Regularly update the system with new features, enhancements, or bug fixes to provide an improved user experience.

4.2 SYSTEM REQUIREMENTS

4.2.1 Operating System

Platform Independent: This refers to the ability of the AI Desktop assistant to function across different operating systems and hardware platforms without requiring significant modifications. While achieving complete platform independence can be challenging due to the variations in software and hardware architectures, efforts are made to ensure compatibility and portability.

4.2.2 Language used

PYTHON:

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

- **Python is Interpreted** – Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- **Python is Interactive** – You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
- **Python is Object-Oriented** – Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
- **Python is a Beginner's Language** – Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

4.3 SYSTEM IMPLEMENTATION

Implementation Work

The following are the libraries used in the project:

- **Pyttsx3** - Pyttsx3 stands for Python Text to Speech. It is a cross-platform Python wrapper for text-to-speech synthesis. It is a Python package supporting common text-to-speech engines on Mac OS X, Windows, and Linux. It works for both Python2.x and 3.x versions. Its main advantage is that it works offline.
- **Datetime** – This module supplies classes for manipulating dates and times. While date and time arithmetic is supported, the focus of the implementation is on efficient attribute extraction for output formatting and manipulation.
- **Speech Recognition** - This is a library for performing speech recognition, with support for several engines and APIs, online and offline. It supports APIs like Google Cloud Speech API, IBM Speech to Text, Microsoft Bing Voice Recognition etc.
- **Wikipedia** – This is a Python library that makes it easy to access and parse data from Wikipedia. Search Wikipedia, get article summaries, get data like links. It wraps the MediaWiki API so that you can focus on using Wikipedia data, not getting it.
- **Webbrowser** - The webbrowser module provides a high-level interface to allow displaying web-based documents to users. Under most circumstances, simply calling the `open()` function from this module will do the right thing.
- **Os** - The OS module in Python provides functions for interacting with the operating system. OS comes under Python's standard utility modules. This module provides a portable way of using operating system-dependent functionality. The `*os*` and `*os.path*` modules include many functions to interact with the file system.

- Subprocess - The subprocess module allows you to spawn new processes, connect to their input/output/error pipes, and obtain their return codes.
- Bs4 - It is a Python library for pulling data out of HTML and XML files. It works with your favourite parser to provide idiomatic ways of navigating, searching, and modifying the parse tree. It commonly saves programmers hours or days of work.
- Tkinter – Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit.

The following are the functions used in the system:

-> speak(): This function is used by the system to speak and provide instructions and results

-> takeCommand(): This function is used to listen to the user voice input, recognize it and interpret it to text for the system to process.

-> wishme(): This functions is used to wish the user as soon as the system starts and provides a menu of services provided

-> speak(): This function is used by the system to speak in response to user queries.

-> get_weather_report(): This function helps get the real time weather report of the city.

4.4 HARDWARE AND SOFTWARE

4.4.1 Software requirements

Platform	Platform Independent
The Operating System	Windows 10
Front-End Tool	Tkinter
Back-End Tool	Python

4.4.2 Hardware Requirements

Processor	Intel Pentium IV 2.9 GHz Other
RAM	Minimum 4 GB
Hard Disk	Minimum 2 GB

Chapter 5

COST BENEFIT ANALYSIS

These are the potential costs of the system:

1. **Research and Development:** The project will require a dedicated team of researchers, engineers, and data scientists to develop and train the speech recognition model. This team will need to be compensated for their time and expertise.
2. **Data Collection and Processing:** Large amounts of speech data will be required to train the AI model effectively. The cost of collecting and processing this data, including hiring voice actors, audio recording equipment, and storage infrastructure, needs to be considered.
3. **Hardware and Infrastructure:** High-performance computing resources and servers will be required to train and deploy the speech recognition system. These hardware costs, including maintenance and upgrades, should be accounted for.
4. **Software Development:** The development of software applications, user interfaces, and integration with existing systems will incur costs. This includes the cost of software licenses, development tools, and quality assurance testing.
5. **Training and Support:** Training end-users and providing ongoing support for the speech recognition system will require resources and staff. This cost should include training materials, documentation, and customer support channels.

These are the benefits of the system:

1. **Increased Efficiency:** The Jarvis AI speech recognition system can automate the transcription process, reducing the time and effort required to convert spoken words into written text. This efficiency gain can benefit various industries, such as legal, medical, and customer service, where accurate transcriptions are essential.
2. **Enhanced Accessibility:** Speech recognition technology can make digital content more accessible to individuals with disabilities, such as those with hearing impairments or motor disabilities. By providing accurate transcriptions, the system can help improve inclusivity and enable a wider range of users to access information.

3. Productivity Improvements: With accurate speech recognition, professionals can dictate their thoughts, ideas, or notes without the need for manual typing. This can increase productivity by allowing individuals to focus on the content rather than the mechanics of writing.

4 Cost Savings: The automation provided by the speech recognition system can reduce the need for manual transcription services, which can be costly. Organizations can potentially save money by replacing or reducing their reliance on external transcription services

5. Competitive Advantage By implementing an advanced speech recognition system organizations can gain a competitive edge in their respective industries. This technology can enhance customer experiences, streamline operations, and differentiate the organization from competitors.

Chapter 6

DETAILED LIFECYCLE OF THE PROJECT

6.1 USE CASE DIAGRAM

A use case is a set of scenarios that describe an interaction between a user and a system.

A use case diagram displays the relationship among actors and use cases. The two main components of a use case diagram are use cases and actors.



Fig. 3.1 Actor and Use case

An actor is represents a user or another system that will interact with the system you are modelling.

A use case is an external view of the system that represents some action the user might perform in order to complete a task.

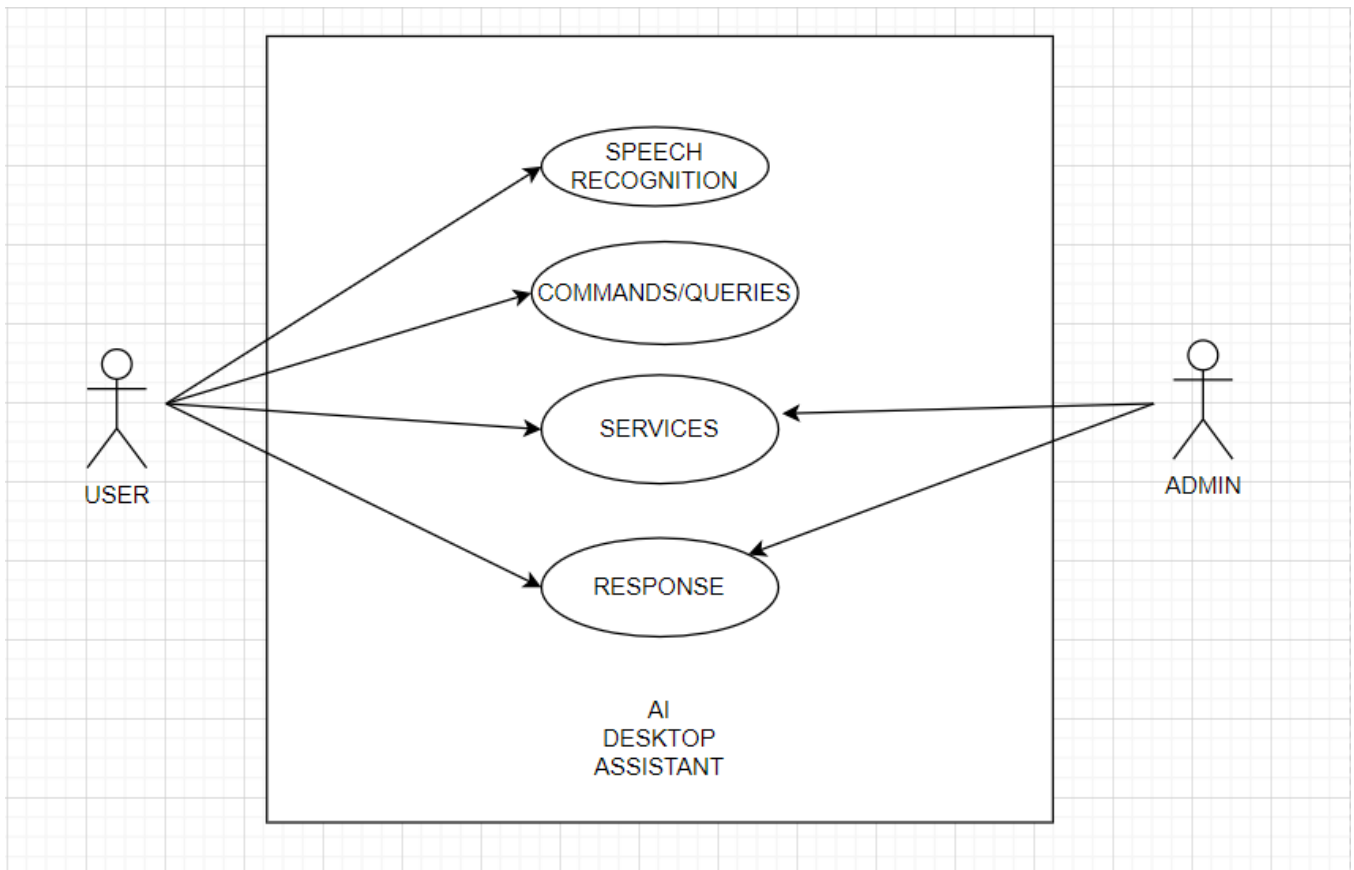


Fig.3.2 Interaction between user and admin

In The above diagram there are 2 actors or users who use the system. These two users are as follows:

1. User-The customer is allowed to provide with commands or queries which are then analysed by the speech recognition. It is then interpreted to commands and provide with appropriate responses.
2. Admin-The admin handles the services provided and the results.

6.2 DFD DIAGRAM

DFD is the abbreviation for Data Flow Diagram. The flow of data of a system or a process is represented by DFD. It also gives insight into the inputs and outputs of each entity and the process itself. DFD does not have control flow and no loops or decision rules are present. Specific operations depending on the type of data can be explained by a flowchart.

It is a graphical tool, useful for communicating with user , managers and other personnel. it is useful for analysing existing as well as proposed system.

It provides an overview of

- What data is system processes.
- What transformation are performed.
- What data are stored.
- What results are produced, etc.

Data Flow Diagram can be represented in several ways. The DFD belongs to structured-analysis modelling tools. Data Flow diagrams are very popular because they help us to visualize the major steps and data involved in software-system processes.

The Data Flow Diagram has 4 components:

- Process Input to output transformation in a system takes place because of process function. The symbols of a process are rectangular with rounded corners, oval, rectangle or a circle. The process is named a short sentence, in one word or a phrase to express its essence
- Data Flow Data flow describes the information transferring between different parts of the systems. The arrow symbol is the symbol of data flow. A relatable name should be given to the flow to determine the information which is being moved. Data flow also represents material along with information that is being moved. Material shifts are modelled in systems that are not merely informative. A given flow should only transfer a single type of information. The direction of flow is represented by the arrow which can also be bi-directional.
- Warehouse: The data is stored in the warehouse for later use. Two horizontal lines represent the symbol of the store. The warehouse is simply not restricted to being a data file rather it can be anything like a folder with documents, an optical disc, a filing cabinet. The data warehouse can be viewed independent of its implementation. When the data flow from the warehouse it is considered as data reading and when data flows to the warehouse it is called data entry or data updating.

- Terminator: The Terminator is an external entity that stands outside of the system and communicates with the system. It can be, for example, organizations like banks, groups of people like customers or different departments of the same organization, which is not a part of the model system and is an external entity. Modelled systems also communicate with terminator.

Rules for creating DFD

- The name of the entity should be easy and understandable without any extra assistance(like comments).
- The processes should be numbered or put in ordered list to be referred easily.
- The DFD should maintain consistency across all the DFD levels.
- A single DFD can have maximum processes upto 9 and minimum 3 processes.

Symbols Used in DFD

- Square Box: A square box defines source or destination of the system. It is also called entity. It is represented by rectangle.
- Arrow or Line: An arrow identifies the data flow i.e. it gives information to the data that is in motion.
- Circle or bubble chart: It represents as a process that gives us information. It is also called processing box.
- Open Rectangle: An open rectangle is a data store. In this data is store either temporary or permanently.

Levels of DFD

DFD uses hierarchy to maintain transparency thus multilevel DFD's can be created. Levels of DFD are as follows:

- 0-level DFD
- 1-level DFD
- 2-level DFD

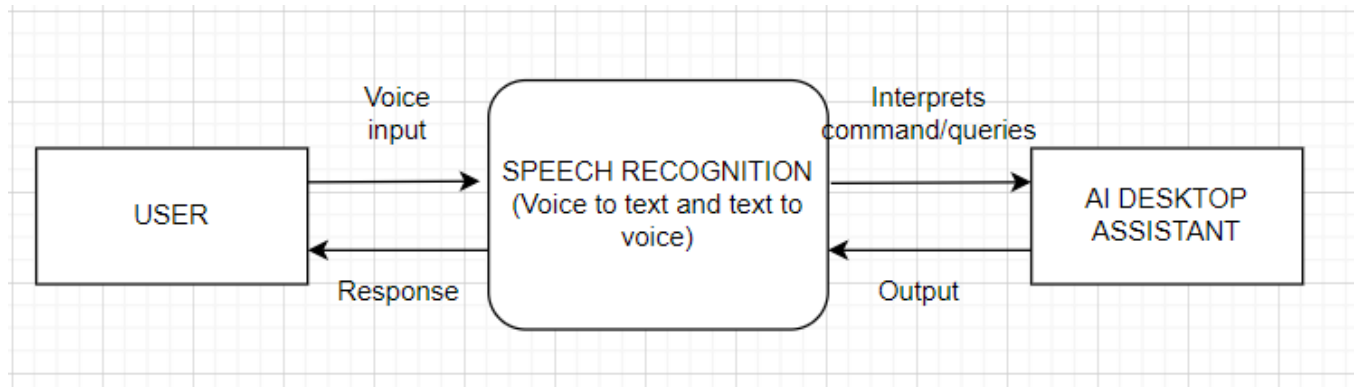
Advantages of DFD

- It helps us to understand the functioning and the limits of a system.
- It is a graphical representation which is very easy to understand as it helps visualize contents.
- Data Flow Diagram represent detailed and well explained diagram of system components.
- It is used as the part of system documentation file.
- Data Flow Diagrams can be understood by both technical or nontechnical person because they are very easy to understand.

Disadvantages of DFD

- At times DFD can confuse the programmers regarding the system.

- Data Flow Diagram takes long time to be generated, and many times due to this reasons analysts are denied permission to work on it.








0- Level DFD

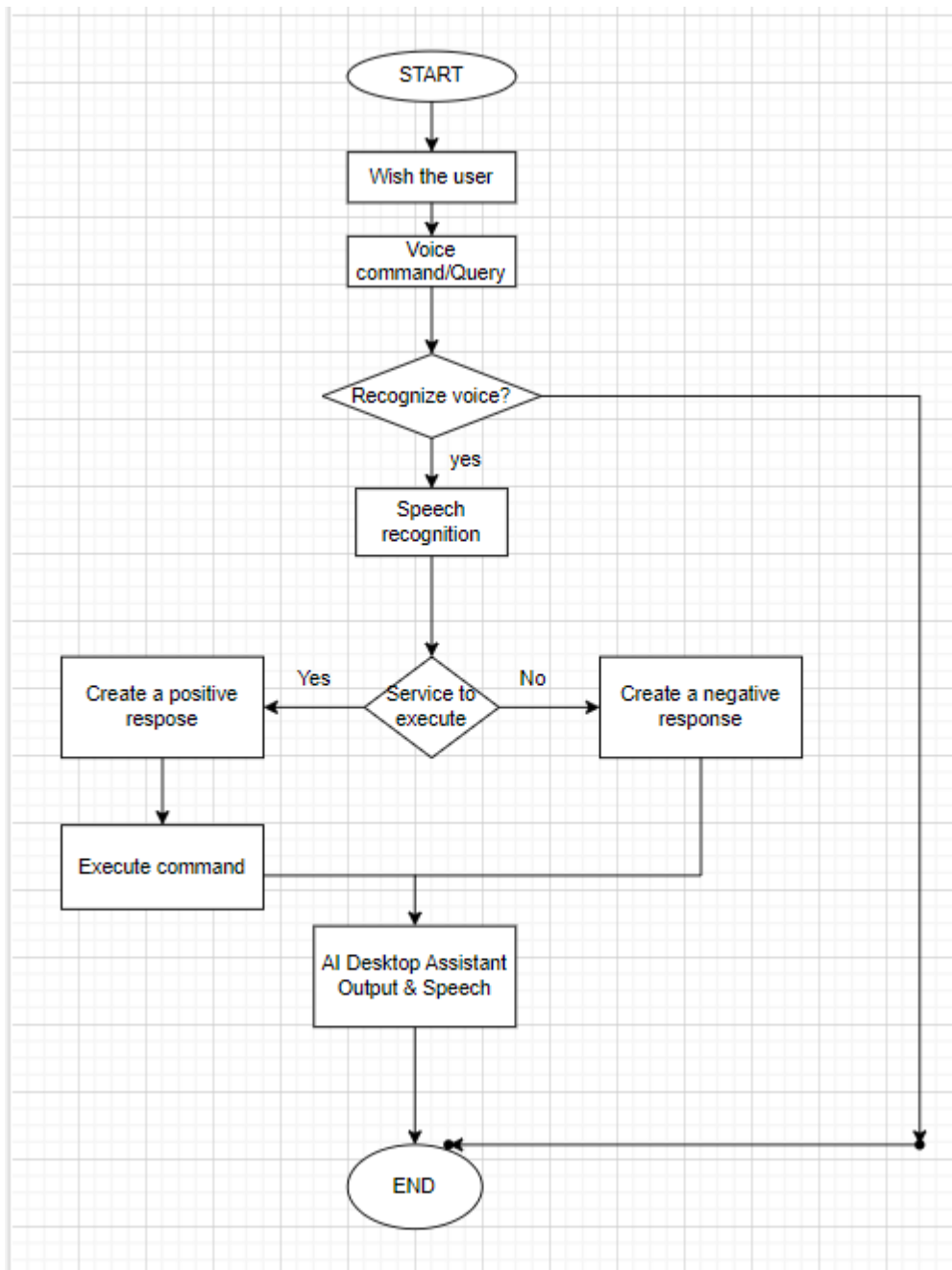
6.3 FLOWCHART

A flowchart is a picture of the separate steps of a process in sequential order. It is a generic tool that can be adapted for a wide variety of purposes, and can be used to describe various processes, such as a manufacturing process, an administrative or service process, or a project plan. Elements that may be included in a flowchart are a sequence of actions, materials or services entering or leaving the process (inputs and outputs), decisions that must be made, people who become involved, time involved at each step, and/or process measurements.

Uses of flowcharts:

- To develop understanding of how a process is done
- To study a process for improvement
- To communicate to others how a process is done
- When better communication is needed between people involved with the same process
- To document a process
- When planning a project

Symbol	Name	Function
	Start/end	An oval represents a start or end point
	Arrows	A line is a connector that shows relationships between the representative shapes
	Input/Output	A parallelogram represents input or output
	Process	A rectangle represents a process
	Decision	A diamond indicates a decision



As the system is started, it first authenticates the authorized user, then voice assistant is on running in the background listening for available voice commands; once the user gives a command, based on the conditions provided to the voice assistant, the voice assistant gives the necessary output. This output is sent to the Speech Recognition which is convert the speech into machine-readable form. Based on the input received the personal voice assistant then performs the desired task.

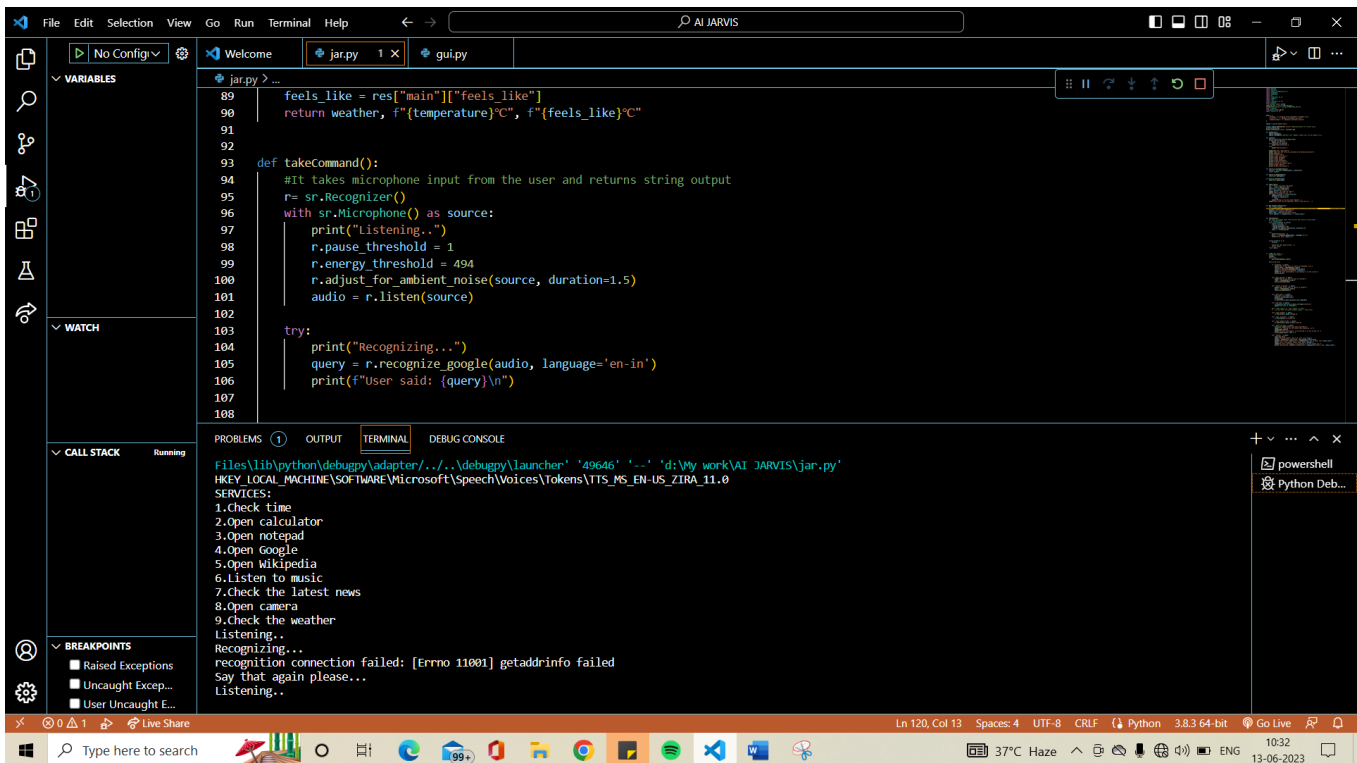
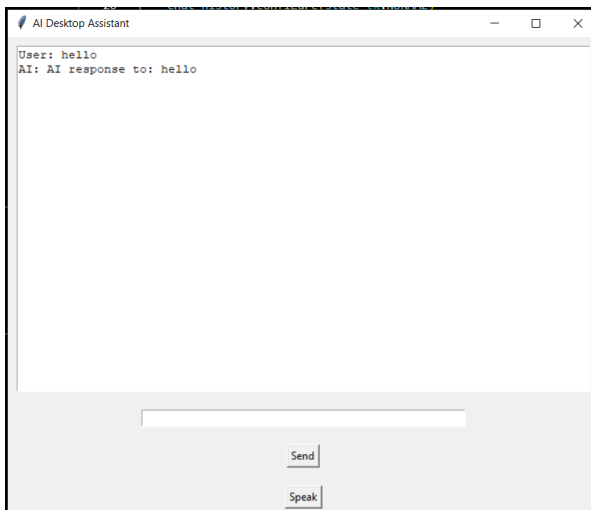
6.4 INPUT AND OUTPUT SCREEN DESIGN

When the system is run, AI Desktop Assistant wishes the user saying, **“Good Morning!**

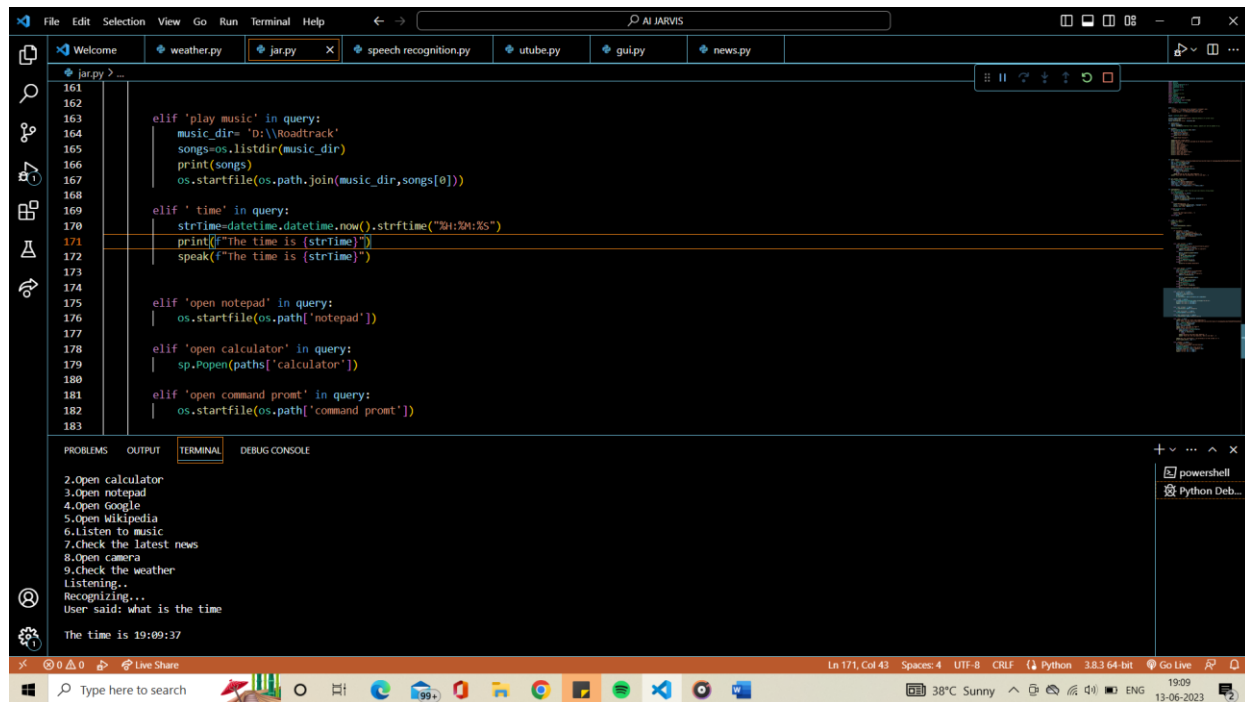
How may I help you? These are the services provided by AI Desktop Assistant”

This is followed result of services provides:

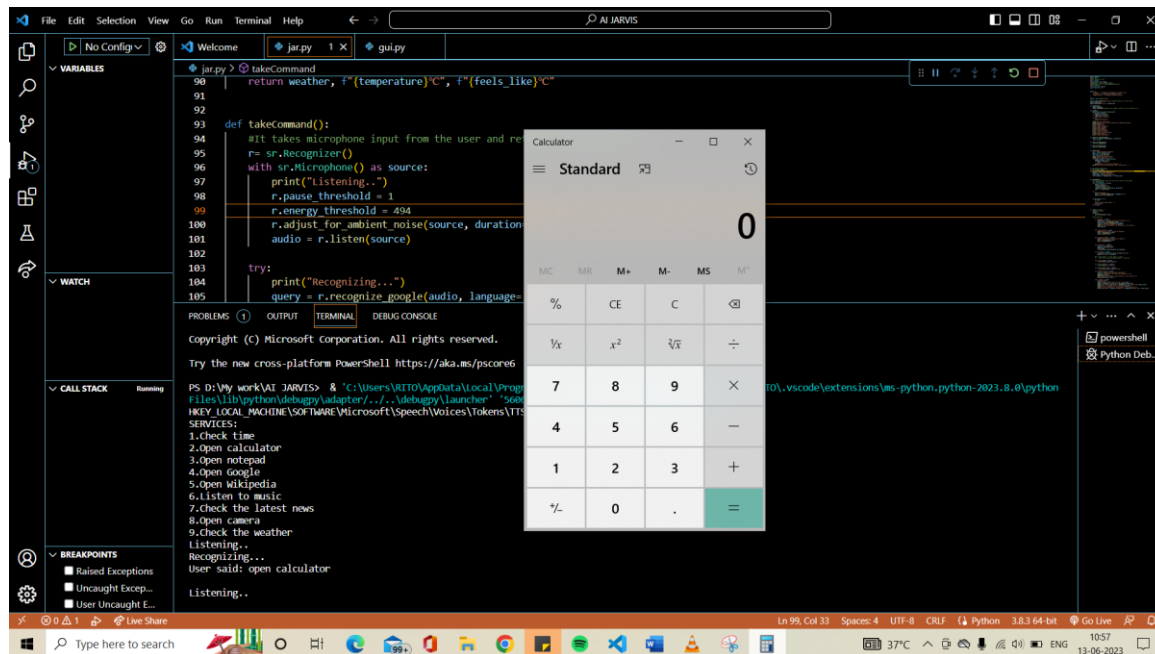
GUI



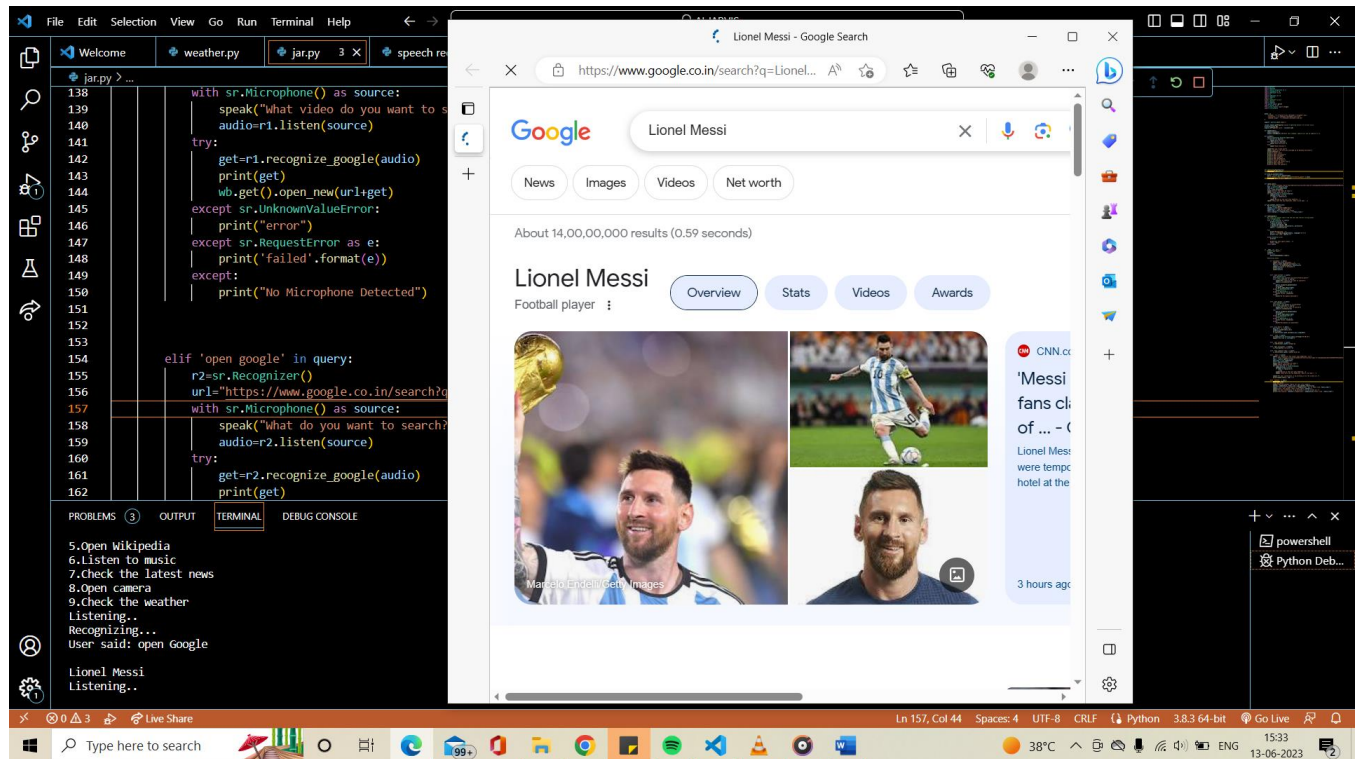
a>Check Time



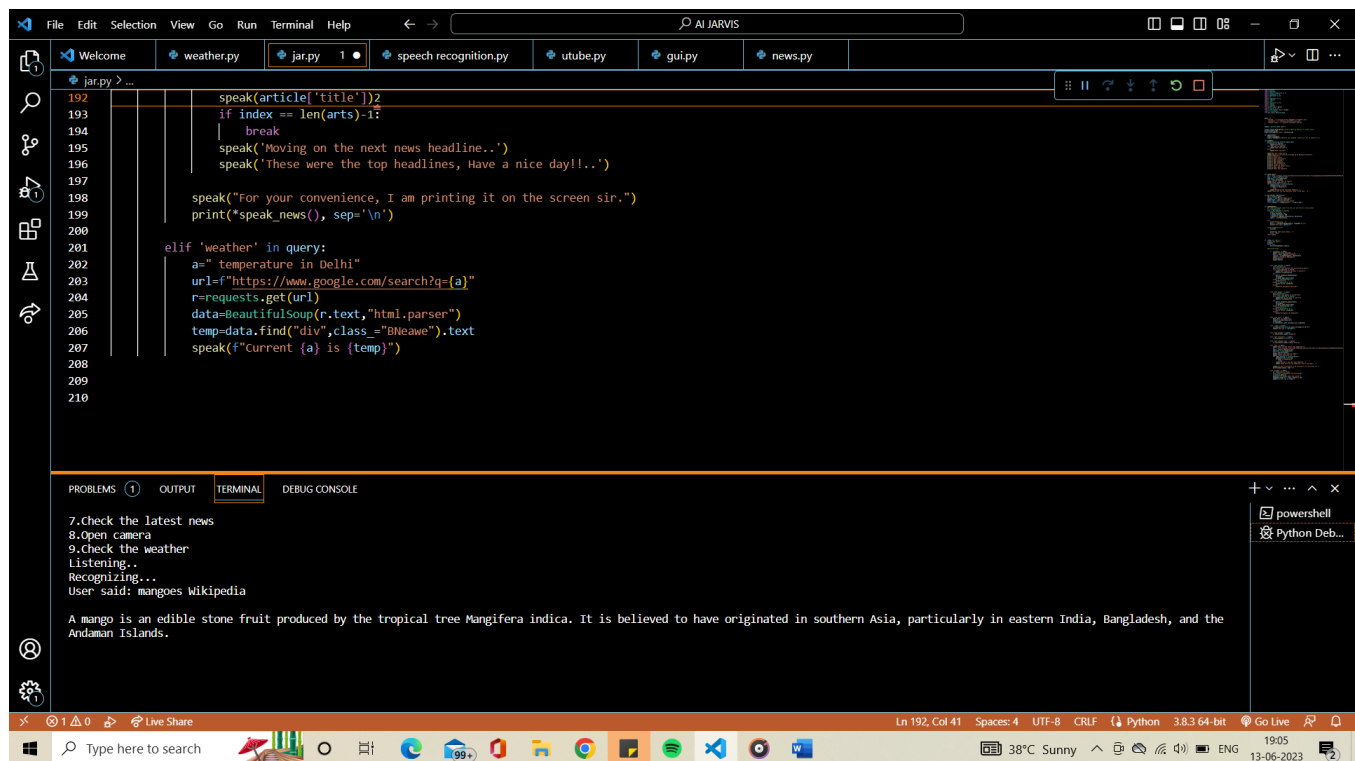
b>Open Calculator



c>Open Google



d>Open Wikipedia



e>Listen to Music

The screenshot shows a music application window titled "My music". It has tabs for "Songs", "Artists", and "Albums". Under the "Songs" tab, there is a list of songs with columns for song name, artist, album, year, genre, and duration. The songs listed are:

Song	Artist	Album	Year	Genre	Duration
"Let it Go" (Full Song by Idina Men	Unknown Artist	Unknown Album	2013	Unknown Genre	3:48
"Let it Go" (Full Song by Idina Men	Unknown Artist	Unknown Album	2013	Unknown Genre	0:50
01 Dream	Unknown Artist	Unknown Album	2013	Unknown Genre	3:31

Below the list is a music player interface showing "A. R. Rahman Medley" by "Aasa Singh | Shambhavi Thak". The player has a progress bar and various control buttons. On the right side of the application, there is a list of files in the "WATCH" section, including various MP3 files and folders.

f>Check the latest news

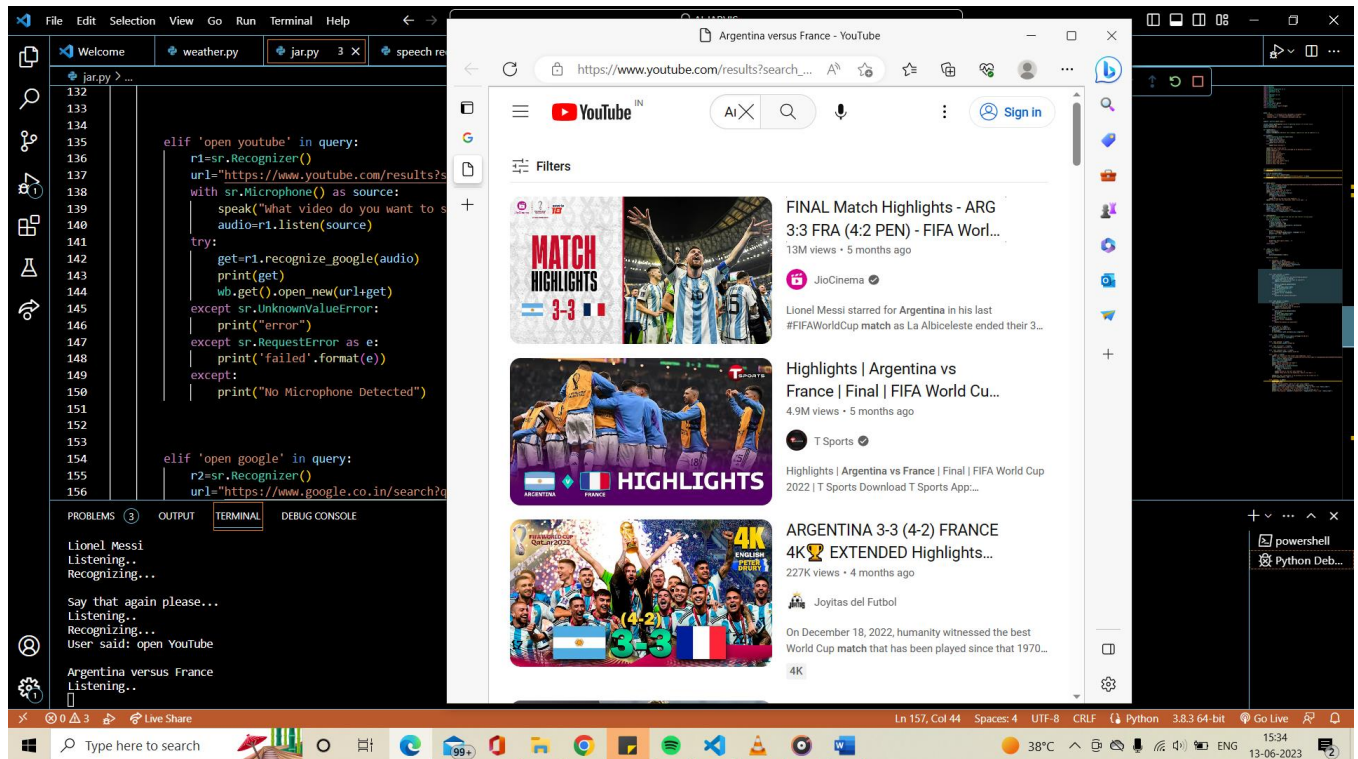
The screenshot shows a Python application running in a terminal window. The application is named "news.py" and is located in the "C:\Users\user\Documents" directory. The code in the file is as follows:

```
184 url = 'http://newsapi.org/v2/top-headlines?sources=the-times-of-india&apiKey=ba1f42b5a6974533a4355cd879817c8a'
185 news = requests.get(url).text
186 news_dict = json.loads(news)
187 arts = news_dict['articles']
188 speak('Source: The Times of India')
189 speak('Today's Headlines are..')
190 for index, article in enumerate(arts):
191     speak(article['title'])
192     if index == len(arts)-1:
193         break
194     speak('Moving on the next news headline..')
195     speak('These were the top headlines, Have a nice day!!!')
196
197 speak('For your convenience, I am printing it on the screen sir.')
198 print(*speak_news(), sep='\n')
```

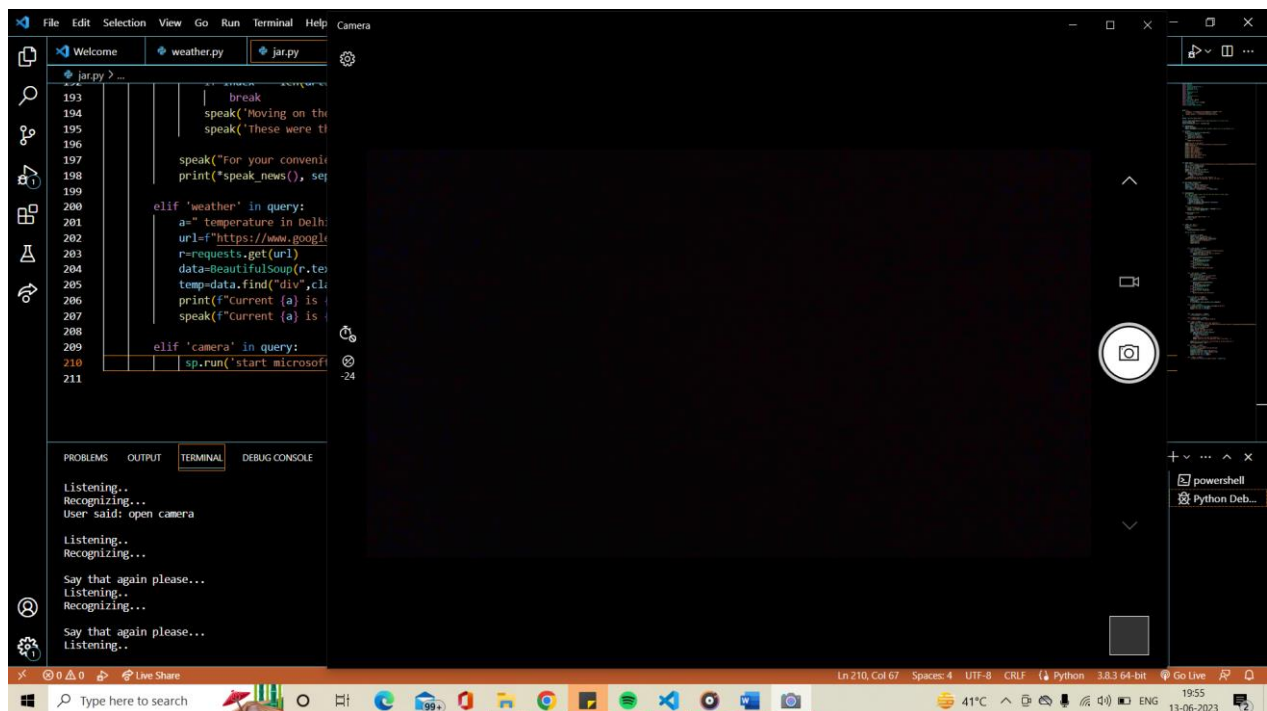
The terminal output shows the application running successfully and displaying the top headlines from The Times of India. The output is as follows:

```
KeyError: 'status'
{
  "status": "ok",
  "totalResults": 7,
  "articles": [
    {
      "source": {
        "id": "the-times-of-india",
        "name": "The Times of India",
        "author": "Rajat Pandit",
        "title": "Army explores procurement of 350 light tanks for mountainous terrain after border standoff with china",
        "description": "India News: The Army is now exploring the possibility of procuring 350 light tanks, which can also be transported by air, to augment its firepower in high-altitude",
        "url": "http://timesofindia.indiatimes.com/india/army-explores-procurement-of-350-light-tanks-for-mountainous-terrain-after-border-standoff-with-china/articleshow/82217825.cms",
        "urlToImage": "https://static.toiimg.com/thumb/msid-82217908,width-1070,height-580,imgsize-264639,resizemode-75,overlay-toi-sw,pt-32,y-pad-40/photo.jpg",
        "publishedAt": "2021-04-23T08:29:00Z",
        "content": "Army explores procurement of 350 light tanks for mountainous terrain after border standoff with china"
      },
      "author": "Times of India",
      "title": "7000 vs 10000: Mumbai Indians seek consistency; Punjab Kings eye return to winning ways",
      "description": "IPL Live score: Mumbai Indians seek consistency; Punjab Kings eye return to winning ways. Stay with TOI to get IPL live score, playing 11, scorecard, highlights and ball by ball score updates of the 17th IPL match between Punjab Kings and Mumbai Indians.",
      "url": "https://timesofindia.indiatimes.com/sports/cricket/ipl/live-blog/punjab-kings-vs-mumbai-indians-pkbs-vs-mi-live-score-ipl-2021-17th-match-chennai/liveblog/82214950.cms",
      "urlToImage": "https://static.toiimg.com/thumb/msid-82214950,width-1070,height-580,imgsize-157009,resizemode-75,overlay-toi-sw,pt-32,y-pad-40/photo.jpg",
      "publishedAt": "2021-04-23T05:44:49Z",
      "content": "Ravi Bishnoi return on the cards? Ravi Bishnoi showed what he is capable of, but greater consistency would be needed from him if the team has to prosper. They bet on Australian pace imports Jhye Richardson and Mitchell Starc."
    },
    {
      "source": {
        "id": "the-times-of-india",
        "name": "The Times of India",
        "author": "PTI",
        "title": "Zydu cadilla gets DCGI nod for hepatitis drug for covid-19 treatment",
        "description": "India News: Drug firm Zydu Cadilla on Friday said it has received restricted emergency use approval from the Indian drug regulator for the use of pegylated interferon-α1b for hepatitis B virus (HBV) treatment in COVID-19 patients.",
        "url": "http://timesofindia.indiatimes.com/india/zydu-cadilla-gets-dcgi-nod-for-hepatitis-drug-for-covid-19-treatment/articleshow/82214909.cms",
        "urlToImage": "https://static.toiimg.com/thumb/msid-82214921,width-1070,height-580,imgsize-98952,resizemode-75,overlay-toi-sw,pt-32,y-pad-40/photo.jpg",
        "publishedAt": "2021-04-23T05:43:00Z",
        "content": null
      },
      "source": {
        "id": "the-times-of-india",
        "name": "The Times of India",
        "author": "Bloomberg",
        "title": "Even record death toll may hide extent of India's covid crisis",
        "description": "India News: Bodies piling up at crematoriums and burial grounds across India are sparking concerns that the death toll from a ferocious new COVID-19 wave may be more than what is being reported.",
        "url": "https://timesofindia.indiatimes.com/india/even-record-death-toll-may-hide-extent-of-indias-covid-crisis/articleshow/82213444.cms",
        "urlToImage": "https://static.toiimg.com/thumb/msid-82213819,width-1070,height-580,imgsize-232887,resizemode-75,overlay-toi-sw,pt-32,y-pad-40/photo.jpg",
        "publishedAt": "2021-04-23T04:41:00Z",
        "content": null
      },
      "source": {
        "id": "the-times-of-india",
        "name": "The Times of India",
        "author": "Dipak K Dash",
        "title": "Government to provide 5 kg free food grains to poor for May & June",
        "description": "India News: The government on Friday announced to provide 5 kg free food grains to the poor for May and June 2021. This will cover nearly 80 crore beneficiaries under the Pradhan Mantri Ujjwala Yojana (PMUY).",
        "url": "http://timesofindia.indiatimes.com/india/government-to-provide-5-kg-free-food-grains-to-poor-for-may-june/articleshow/82213582.cms",
        "urlToImage": "https://static.toiimg.com/thumb/msid-82213583,width-1070,height-580,imgsize-1921513,resizemode-75,overlay-toi-sw,pt-32,y-pad-40/photo.jpg",
        "publishedAt": "2021-04-23T04:23:00Z",
        "content": null
      },
      "source": {
        "id": "the-times-of-india",
        "name": "The Times of India"
      }
    }
  ]
}
```

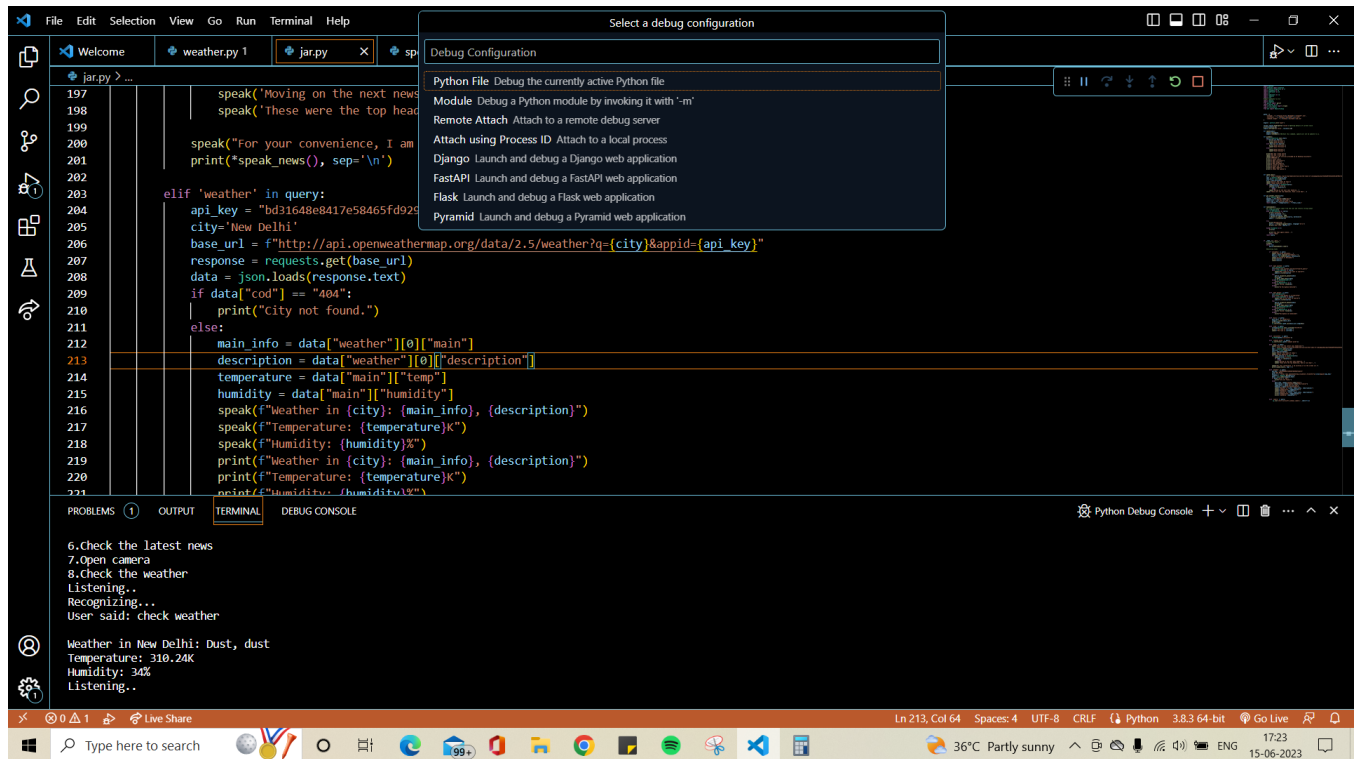
g>Open You Tube



h>Open camera



i>Check the weather



6.5 METHODOLOGY USED FOR TESTING

In the AI Desktop Assistant, various testing methodologies can be employed to ensure the quality, functionality, and reliability of the system. Here are some common testing methodologies that can be applied:

Unit Testing:

Unit testing involves testing individual components or modules of the system in isolation. Each component, such as functions, classes, or methods, is tested to verify its correctness and adherence to the specified requirements. Unit tests are typically automated and focus on testing the smallest units of code.

Integration Testing:

Integration testing aims to verify the correct interaction and integration between different components or modules of the system. It ensures that the components work together as intended and that data flows correctly between them. Integration tests can identify issues related to component dependencies, communication, and data integrity.

System Testing:

System testing is performed on the complete system to evaluate its compliance with the specified requirements. It involves testing the system as a whole, including all components, interfaces, and interactions. System tests focus on validating the system's functionality, performance, security, and usability.

Chapter 7

CODING

```
import pytsx3
from datetime import datetime
import speech_recognition as sr
import wikipedia as wk
import webbrowser as wb
import os
import subprocess as sp
import requests
import json
import smtplib
import requests
from pprint import pprint
from urllib.request import urlopen
import urllib.parse
from bs4 import BeautifulSoup

paths = {
    'calculator': "C:\\Windows\\System32\\calc.exe"
}

engine = pytsx3.init('sapi5')

voices= engine.getProperty('voices') #getting details of current voice
print(voices[1].id)
engine.setProperty('voice', voices[1].id)

def speak(audio):
    engine.say(audio)
    engine.runAndWait() #Without this command, speech will not be audible to us.

def wishme():
    hour=int(datetime.now().hour)
    if hour>=0 and hour<12:
        print("Good morning!")
        speak("Good Morning!")
    elif hour>=12 and hour<18:
        print("Good Afternoon!")
        speak("Good Afternoon!")
    else:
```

```

        print("Good Evening!")
        speak("Good Evening!")

# print("How may I help you?")
speak("How may I help you?")
speak("These are the services provided by AI Desktop Assistant")
print("SERVICES:")
print("1.Check time")
print("2.Open calculator")
print("3.Open Google")
print("4.Open Wikipedia")
print("5.Listen to music")
print("6.Check the latest news")
print("7.Open camera")
print("8.Check the weather")

def speak_news():
    url = 'http://newsapi.org/v2/top-headlines?sources=the-times-of-india&apiKey=ba1f42b5a6974533a4355cd879017c8a'
    news = requests.get(url).text
    news_dict = json.loads(news)
    arts = news_dict[news]
    speak('Source: The Times Of India')
    speak('Todays Headlines are..')
    for index,article in enumerate(arts):
        speak(article['title'])
        if index == len(arts)-1:
            break
        speak('Moving on the next news headline..')
    speak('These were the top headlines, Have a nice day!!..')

def get_weather_report(city):
    res = requests.get()
    weather = res["weather"][0]["main"]
    temperature = res["main"]["temp"]
    feels_like = res["main"]["feels_like"]
    return weather, f"{temperature}°C", f"{feels_like}°C"

def takeCommand():
    #It takes microphone input from the user and returns string output
    r= sr.Recognizer()
    with sr.Microphone() as source:
        print("Listening..")

```



```

r.pause_threshold = 0.8
r.energy_threshold = 1000
r.adjust_for_ambient_noise(source, duration=3)
audio = r.listen(source)

try:
    print("Recognizing...")
    query = r.recognize_google(audio, language='en-in')
    print(f"User said: {query}\n")

except Exception as e:
    print(e)

    print("Say that again please...")
    return "None"
return query

if __name__=="__main__" :
    #speak("Hey Payel")
    wishme()
    while True:
        query=takeCommand().lower()

    #Executing tasks

    if 'wikipedia' in query:
        speak('Searching Wikipedia...')
        query = query.replace("wikipedia"," ")
        results = wk.summary(query, sentences=2)
        speak("According to Wikipedia")
        print(results)
        speak(results)

    elif 'open youtube' in query:
        r1=sr.Recognizer()
        url="https://www.youtube.com/results?search_query="
        with sr.Microphone() as source:
            speak("What video do you want to search?")
            audio=r1.listen(source)

        try:
            get=r1.recognize_google(audio)
            print(get)

```

```

        wb.get().open_new(url+get)
    except sr.UnknownValueError:
        print("error")
    except sr.RequestError as e:
        print('failed'.format(e))
    except:
        print("No Microphone Detected")

elif 'open google' in query:
    r2=sr.Recognizer()
    url="https://www.google.co.in/search?q="
    with sr.Microphone() as source:
        speak("What do you want to search?")
        audio=r2.listen(source)
    try:
        get=r2.recognize_google(audio)
        print(get)
        wb.get().open_new(url+get)
    except sr.UnknownValueError:
        print("error")
    except sr.RequestError as e:
        print('failed'.format(e))
    except:
        print("Microphone not detected")

elif 'music' in query:
    music_dir= 'D:\\Roadtrack'
    songs=os.listdir(music_dir)
    print(songs)
    os.startfile(os.path.join(music_dir,songs[0]))

elif 'time' in query:
    strTime=datetime.now().strftime("%H:%M:%S")
    print(f"The time is {strTime}")
    speak(f"The time is {strTime}")

elif 'calculator' in query:
    sp.Popen(paths['calculator'])

elif 'command prompt' in query:
    os.startfile(os.path['command prompt'])

```

```

elif 'news' in query:
    speak("These are the latest news headlines!")
    url = 'http://newsapi.org/v2/top-headlines?sources=the-times-of-india&apiKey=ba1f42b5a6974533a4355cd879017c8a'
    news = requests.get(url).text
    news_dict = json.loads(news)
    arts = news_dict[news]
    speak('Source: The Times Of India')
    speak('Todays Headlines are..')
    for index,article in enumerate(arts):
        speak(article['title'])
        if index == len(arts)-1:
            break
        speak('Moving on the next news headline..')
        speak('These were the top headlines, Have a nice day!!..')

    speak("For your convenience, I am printing it on the screen sir.")
    print(*speak_news(), sep='\n')

elif 'weather' in query:
    api_key = "bd31648e8417e58465fd929e6f1da172"
    city='New Delhi'
    base_url =
f"http://api.openweathermap.org/data/2.5/weather?q={city}&appid={api_key}"
    response = requests.get(base_url)
    data = json.loads(response.text)
    if data["cod"] == "404":
        print("City not found.")
    else:
        main_info = data["weather"][0]["main"]
        description = data["weather"][0]["description"]
        temperature = data["main"]["temp"]
        humidity = data["main"]["humidity"]
        speak(f"Weather in {city}: {main_info}, {description}")
        speak(f"Temperature: {temperature}K")
        speak(f"Humidity: {humidity}%")
        print(f"Weather in {city}: {main_info}, {description}")
        print(f"Temperature: {temperature}K")
        print(f"Humidity: {humidity}%")

elif 'camera' in query:
    sp.run('start microsoft.windows.camera:', shell=True)

```

Chapter 8

CONCLUSION

AI Desktop Assistant is a very helpful voice assistant without any doubt as it saves time of the user by conversational interactions, its effectiveness and efficiency. But while working on this project, there were some limitations encountered and also realized some scope of enhancement in the future which are mentioned below:

8.1 LIMITATIONS

The following are the limitations of the project:

1. Security is somewhere an issue, there is no voice command encryption in this project.
2. Background voice can interfere
3. Misinterpretation because of accents and may cause inaccurate results.
4. It cannot be called externally anytime like other traditional assistants like Google Assistant can be called just by saying: "Ok Google"

8.3 FUTURE SCOPE

The future scope of AI Desktop Assistant is vast and continuously evolving. Here are some key potential advancements and areas of development:

1. Advanced Natural Language Understanding: AI Desktop Assistants will become more proficient in understanding and interpreting complex natural language inputs, including context, emotions, and intents, enabling more sophisticated interactions and personalized responses.

2. **Multimodal Interaction:** Future assistants may integrate additional modalities such as image recognition, gesture recognition, and touch interfaces, allowing users to interact through a combination of speech, visuals, and physical interactions.
3. **Context-Awareness:** AI Desktop Assistants will leverage contextual information from various sources (e.g., calendar, location, user history) to provide proactive and contextually relevant assistance. They will anticipate user needs, make suggestions, and adapt behavior accordingly.
4. **Enhanced Personalization:** Assistants will continuously learn from user interactions, preferences, and feedback to deliver personalized experiences tailored to individual users, including customized recommendations, content, and task management.
5. **Deep Integration with Applications and Services:** Future assistants will be tightly integrated with a wide range of applications, services, and IoT devices, allowing seamless control and management of various tasks, such as home automation, scheduling, e-commerce, and productivity tools.
6. **Improved Speech Recognition and Language Models:** Ongoing advancements in speech recognition technologies, including neural network architectures, acoustic modeling, and language models, will lead to increased accuracy, better performance in noisy environments, and support for multiple languages.
7. **Privacy and Security Enhancements:** Future AI Desktop Assistants will prioritize user privacy and data security, employing advanced encryption techniques, on-device processing, and strict user consent management to ensure the confidentiality and integrity of user information.

Chapter 9

BIBLIOGRAPHY

- <http://www.w3schools.com>
- [http:// www .stackoverflow.com](http://www.stackoverflow.com)
- <http://www.scalar.com>
- <https://www.google.com>