**GOVERNMENT ENGINEERING COLLEGE, BHARATPUR**



**A Project report on**

**“**AUDIO RECOGANITION UNIT**”**

**(A.R.U)**

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF BACHELOR OF ENGINEERING IN COMPUTER SCIENCE AND ENGINEERING BY: -

**DEEPAK CHAUDHARY 21EELCSE013**

**AMIT GARG 21EELCSE006**

**ARPIT SAXENA 21EELCSE008**

UNDER THE GUIDENCE OF MR. SUMIT KUMAR

(NETWORK MANAGER)

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**DECLARATION**

We the students of the computer science and engineering from **Government Engineering College Bharatpur** Declare that the work titled with the name **“AUDIO RECOGNIZATION UNIT”** has been successfully completed under the guidance of **MR. SUMIT KUMAR** ,of computer science and engineering department ,of government engineering college bharatpur .This is to certify that in partial fulfilment of **B.TECH CSE DEGREE (SEMESTER-3)** examination during the academic year 2022-2023 had not been submitted for any other examination and it does not part of any other course .It is further certified that the project is completed with the all required phases of subject.

**Further we want to tell that the content that we used in this project report has not been submitted previously by anyone for the award of any degree or diploma university**.

**ABSTRACT**

As we know that the 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 the user. speech recognition is the process of converting speech into text. This is commonly used in voice assistants like Alexa, Siri etc. In python there is an API called **SPEECH RECOGNITION** which allow us to convert speech into the text. Performing many daily tasks like playing music, video, Wikipedia etc. Advancement of technologies are such that they can perform any task with same effectiveness or can say more effectively than us .by making this project we realize that the technology in every field is decreasing the human effort and save time.

Functionalities of the project include:

1. It can send an email.
2. It can read pdf.
3. It can play music.
4. It can do Wikipedia search for you.
5. It can open website like Google, you tube etc in the web browser.
6. It can tell the time.

The application (A.R.U) work like Siri, google assistant, it takes voice as input so all the services provided by the device accessible by the end user on the user voice command.

**AKNOWLEDGEMENT**

We like to take this opportunity to express our sincere gratitude and respect to **GOVERNMENT ENGINEERING COLLEGE, BHARATPUR** for providing the platform to pursue my studies and carry out my project.

We have a great pleasure to express my deep sense of gratitude to **Dr. RAVI GUPTA** Sir, principal GEC bharatpur for his constant encouragement.

We would like to thank **Dr. Prashant Kumar baheti** sir professor and head, department of computer science and engineering, GEC, bharatpur, who has been a constant support throughout the course of this project.

We consider it a privilege and honar to express my sincere gratitude to my guide Mr. Dhawal viyas, Associative professor, department of computer science and engineering.

Finally, I would like to thanks my parents and my friends for all the moral support they provided during the completion of this project.

**TABLE OF CONTENTS**

* CERTIFICATE 1
* DECLARATION 2
* ABSTRACT 3
* AKNOWLEDGEMENT 4
* TABLEOFCONTENTS 5-6
* LISTOFFIGRES 7
* INDIVIDUALCONTRIBUTIONANDREFRENCES 24

|  |  |  |
| --- | --- | --- |
| S.NO | **TOPIC** | **PAGE**  **NO** |
|  | **CHAPTER-1** |  |
|  | **INTRODUCTION** |  |
| 1. | 1.1 PRESENT SYSTEM | 8 |
|  | 1.2 PROPOSED SYTEM | 9 |
|  | 1.3 OBJECTIVE | 10 |
|  | **CHAPTER-2** |  |
|  | **SYSTEM REQUIREMENT AND SPECIFICATION** |  |
| 2 | 2.1 HARDWARE REQUIREMENT | 11 |
|  | 2.2 SOFTWARE SPECIFICATION |  |
|  | **CHAPTER-3** |  |
|  | **SYSTEM ANALYSIS AND DESIGN** |  |
| 3 | 3.1 SYSTEM ARCHITECTURE | 12 |
|  | 3.2 DATA FLOW | 13 |
|  | **CHAPTER-4**  **SOFTWARE DETAILS** |  |
| 4 | 4.1 VS CODE | 14 |
|  | 4.2 GRAPHICAL USER INTERFACE | 15 |
|  | **CHAPTER-5**  **IMPLEMENTATION WORK DETAIL** |  |
|  | 5.1 REAL LIFE APPLICATIONS | 16 |
| 5 | 5.2 LIBRARIES AND PACKAGES | 17 |
|  | 5.3 FUNCTIONS | 18 |
| 6 | **CHAPTER-6**  CODE AND OUTPUT | 19-22 |
| 7 | **CHAPTER-7**  **CONCLUSION** | 23 |
|  | REFERENCES | 24 |

**LIST OF FIGURES**

* **Fig 1.1 objectives 11**
* **Fig 3.1 system architecture 13**
* **Fig 3.2 data flow 14**
* **fig 4.1 visual studio code 15**
* **fig 4.2 pyqt5 16**
* **Fig5.2 modules 17**
* **Fig 6.1 main code of Aru 18**
* **Fig 6.2 gui code of Aru 19**
* **Fig 6.3 interaction with Aru 20**
* **Fig 6.4 Wikipedia result 20**
* **Fig 6.5 gui output of aru 21**

**CHAPTER-1**

**INTRODUCTION**

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 send emails without typing any word, searching on Google without opening the browser, and performing many other daily tasks like playing music, opening your favourite application with the help of a single voice command. In the 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.

The functionalities include It can send emails, it can read PDF, it can send text on WhatsApp, it can open command prompt, your favourite IDE, notepad etc., It can play music, it can do Wikipedia searches for you, it can open websites like Google, YouTube, etc., in a web browser, it can give weather forecast, it can give desktop reminders of your choice. It can have some basic conversation.

Tools and technologies used are VISUAL STUDIO IDE for making this project, and I created all python files in visual studio. Along with this I used following modules and libraries in my project. pyttsx3, Speech Recognition, Datetime, Wikipedia, Smtplib, pywhatkit, pyjokes, pyPDF2, pyautogui, pyQt etc.

I have created a live GUI for interacting with the A.R.U as it gives a design and interesting look while having the conversation.

* 1. **Present System**

We are familiar with many existing voice assistants like Alexa, Siri, Google Assistant, Cortana which uses concept of language processing, and voice recognition .it listens the command given by the user as per their requirements and performs that specific function in a very efficient and effective manner.

As these voice assistants are using Artificial Intelligence hence the result that they are providing are highly accurate and efficient. These assistants 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. These assistants are no less than a human assistant but we can say that they are more effective and efficient to perform any task. The algorithm used to make these assistant focuses on the time complexities and reduces time.

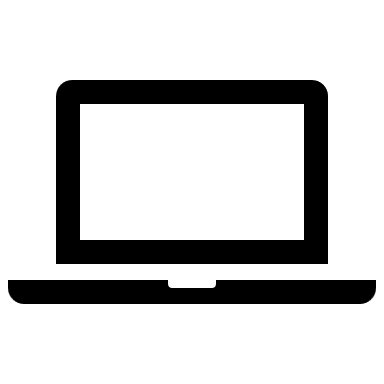
But for using these assistants one should have an account (like Google account for Google assistant, Microsoft account for Cortana) and can use it with internet connection only because these assistants are going to work with internet connectivity. They are integrated with many devices like, phones, laptops, and speakers etc.

* 1. **Proposed system**

It was an interesting task to make my own assistant. It became easier to send emails without typing any word, searching 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. ARU is different from other traditional voice assistants in terms that it is specific to desktop and user does not need to make account to use this, it does not require any internet connection while getting the instructions to perform any specific task.

The IDE used in this project is VISUAL STUDIO. All the python files were created in vs code and all the necessary packages were easily installable in this IDE. For this project following modules and libraries were used i.e., pyttsx3, Speech Recognition, Datetime, Wikipedia, Smtplib, pywhatkit, pyjokes, pyPDF2, pyautogui, pyQt etc. we have created a live GUI for interacting with the ARU as it gives a design and interesting look while having the conversation.

With the advancement JARVIS 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.

****Functionalities of this project include, it can send emails, it can read PDF, it can send text on WhatsApp, it can open command prompt, your favourite IDE, notepad etc., It can play music, it can do Wikipedia searches for you, it can open websites like Google, YouTube, etc., in a web browser, it can give weather forecast, it gives desktop reminders of your choice. It can have some basic conversation.

* 1. **Objective**

**1.IMPROVE MACHINE INTERACTION**

**BY VOICE SPEECH**

**2.IMPROVE SET OF RULES**

**3.IMPROVE SCALIBILITY**

FIG 1.1 Objectives

**CHAPTER-2**

**SYSTEM REQUIRMENT AND SPECIFICATIONS**

This chapter involve both the hardware and software requirements needed for the project and detailed explanation of the specifications**.**

**2.1 Hardware requirements**

1. A pc with windows/ Linux operating system
2. Processor with 1.7ghz or more speed
3. Minimum 8 Gb ram
4. Minimum 128 Gb SSD
   1. **Software specifications**
5. Text editor (vs-code/ web storm/ Py charm)
6. Python modules
7. Python libraries
8. We need vs-code and python console

**CHAPTER-3**

**SYSTEM ANALYSIS AND DESIGN**

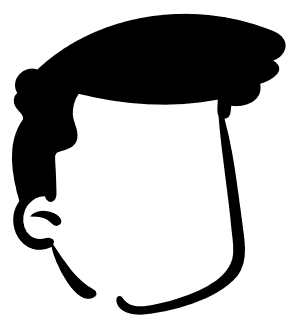
In this chapter we will discuss about the system architecture how the process is running and the data flow of system though pictorial representation.

**3.1 System architecture**

**SPEECH RECOGNITION MODULE**

**HEY ARU**

**VOICE ASSIST**



**API CALLS**

**CONTACT EXTRACTOR**

**TEXT TO SPEECH**

**MODULE**

**PYTHON**

**BACKEND**

FIG 3.1 SYSTEM ARCHITECTURE

The fig 3.1 show the architecture of the voice assistant ARU where firstly the speech recognition module runs when the user ser HEY ARU then the python backend work where it calls the APIS than it works on contact extractor then the text to speech module runs.

**3.2 Data flow**

**FIG 3.2 Data flow**

The upper flowchart diagram shows the proper flow of data in the project of audio recognition unit (ARU) the voice assistant by the python language. The data in this project is nothing but user input, whatever the user says, the assistant performs the task accordingly. The user input is nothing specific but the list of tasks which a user wants to get performed in human language i.e., English.

**CHAPTER-4**

**SOFTWARE DETAILS**

The IDE used in this project is VS CODE. All the python files were created in vs code and all the necessary packages were easily installable in this IDE. For this project following modules and libraries were used i.e., pyttsx3, Speech Recognition, Datetime, Wikipedia, Smtplib, pywhatkit, pyjokes, pyPDF2, pyautogui, pyQt etc. I have created a live GUI for interacting with the ARU as it gives a design and interesting look while having the conversation.

**4.1 VS Code**

It is an IDE i.e., Integrated Development Environment which has many features like it supports scientific tools (like matplotlib, NumPy, SciPy etc) web frameworks (example Django, web2py and Flask) refactoring in Python, integrated python debugger, code completion, code and project navigation etc. It also provides Data Science when used with Anaconda.

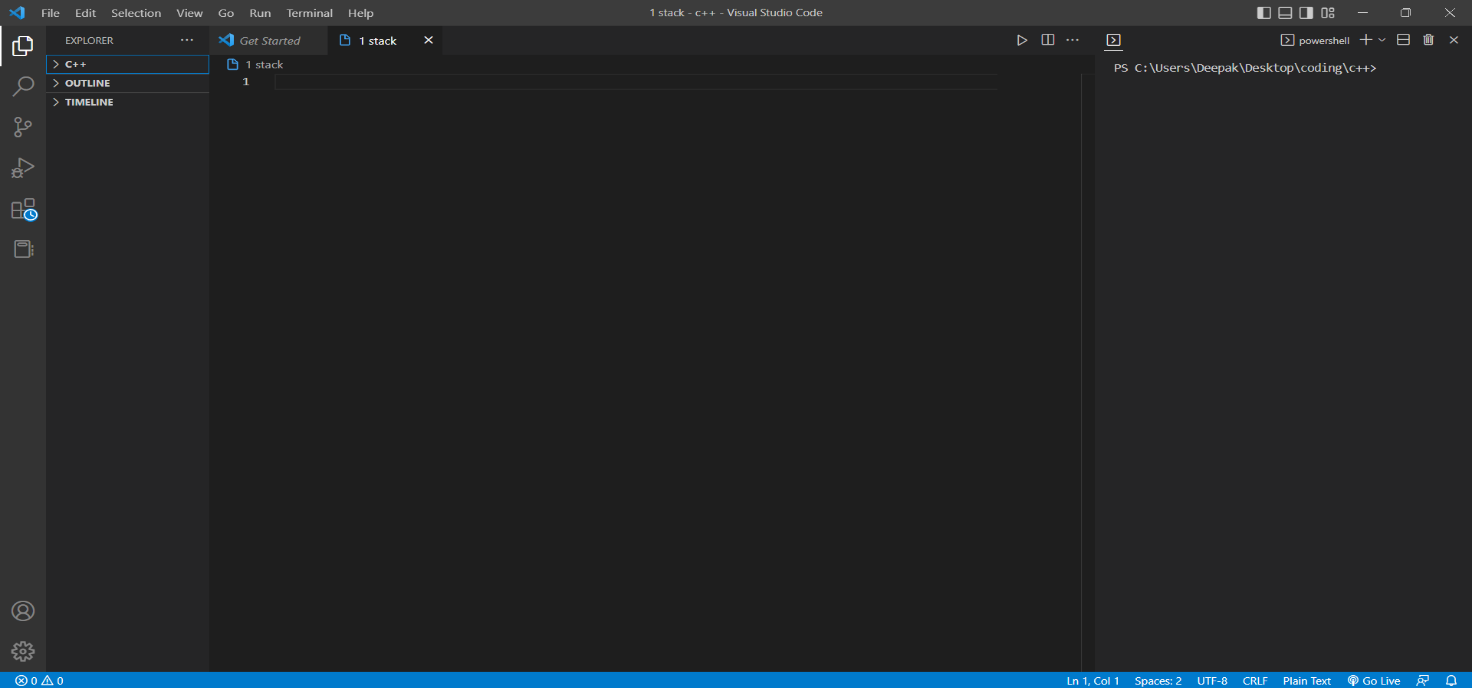


FIG 4.1 VISUAL STUDIO CODE

**4.2 Graphical user interface (GUI)**

PyQt5 is the most important python binding. It contains set of GUI widgets. PyQt5 has some important python modules like Widgets, QtCore, QtGui, and Designer etc.

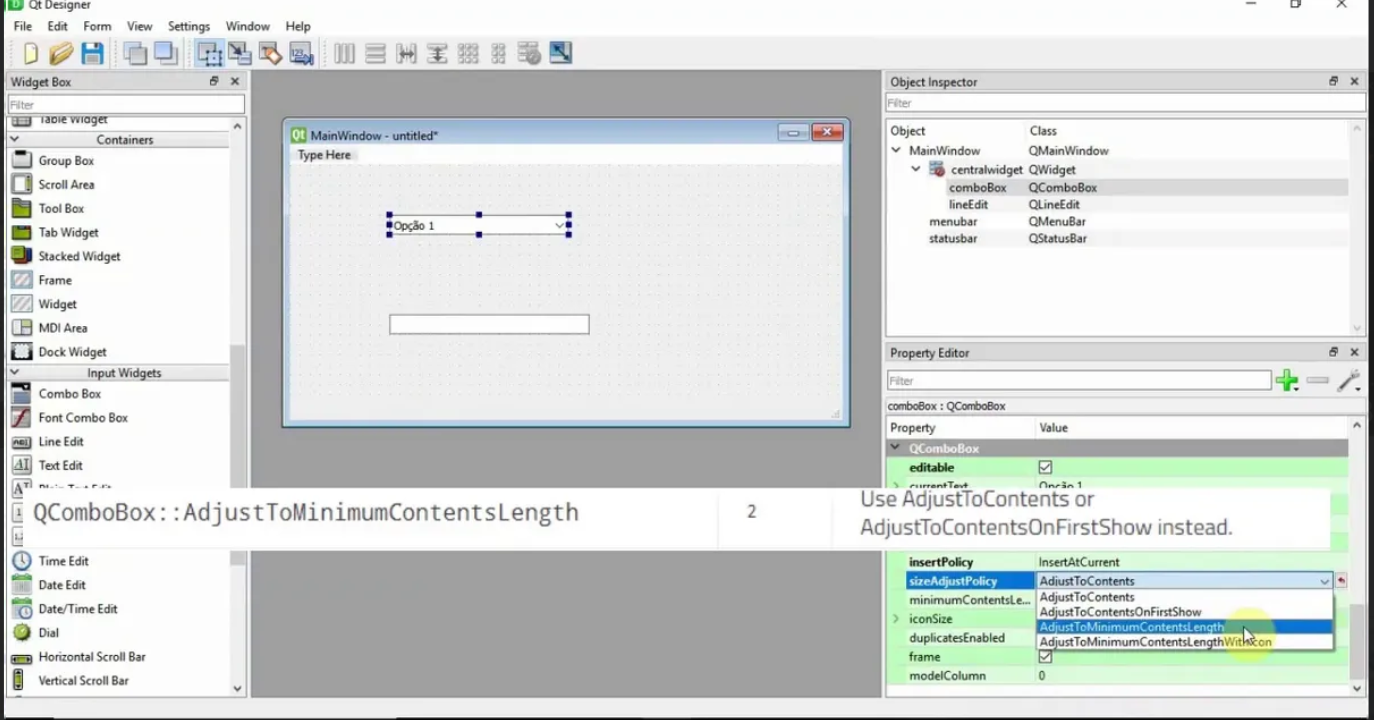


FIG 4.2 PYQT5

**CHAPTER-5**

**IMPLEMENTATION WORK DETAILS**

A.R.U, a desktop assistant is a voice assistant that can perform many daily tasks of desktop like playing music, opening your favourite IDE with the help of a single voice command. ARU is different from other traditional voice assistants in terms that it is specific to desktop and user does not need to make account to use this, it does not require any internet connection while getting the instructions to perform any specific task.

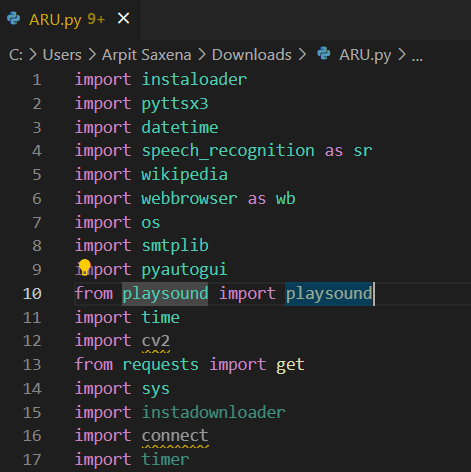
**5.1 Real life application**

* **Saves time:** ARU is a voice assistant which works on the voice command offered to it, it can do voice searching, voice-activated device control let us complete a set of tasks.
* **Conversational interaction** It makes it easier to complete any task as it automatically it by using the essential module or libraries of Python, in a conversational interaction way. Hence any user when instruct any task to it, they feel like giving task to a human assistant because of the conversational interaction for giving input and getting the desired output in the form of task done.
* **Reactive nature** the assistant is reactive which means it know human language very well and understand the context that is provided by the user and gives response in the same way, i.e., human understandable language, English. So, user finds its reaction in an informed and smart way.
* **Multitasking:** The main application of it can be its multitasking ability. It can ask for continuous instruction one after other until the user “QUIT” it.
* **No Trigger phase:** It asks for the instruction and listen the response that is given by user without needing any trigger phase and then only executes the task.

**5.2 Libraries and packages**

As the first step, install all the necessary packages and libraries. The command used to install the libraries is “pip install” and then import it. The necessary packages included are as follows:

* **pyttsx3:** It is a python library which converts text to speech.
* Speech Recognition: It is a python module which converts speech to text.
* **Pywhatkit:** It is python library to send WhatsApp message at a particular time with some additional features.
* **Datetime:** This library provides us the actual date and time.
* **Wikipedia:** It is a python module for searching anything on Wikipedia.
* **Smtplib:** Simple mail transfer protocol that allows us to send mails and to route mails between mail servers.
* **Pypdf2:** It is a python module which can read, split, merge any PDF.
* **Pyjokes:** It is a python library which contains lots of interesting jokes in it.
* **Web browser:** It provides interface for displaying web-based documents to users.
* **Pyautogui:** It is a python library for graphical user interface.
* **os:** It represents Operating System related functionality.
* **sys:** It allows operating on the interpreter as it provides access to the variables and functions that usually interact strongly with the interpreter.

****

**Fig5.2 modules**

**5.3 Functions**

* **take Command ():** The function is used to take the command as input through microphone of user and returns the output as string.
* **wish Me ():** This function greets the user according to the time like Good Morning, Good Afternoon and Good Evening.
* **task Execution ()**: This is the function which contains all the necessary task execution definition like send email (), PDFCreator (), news () and many conditions in if condition like “open google”, “open notepad”, “search on Wikipedia”,” play music” and “open command prompt” etc.

**CHAPTER-6**

**CODE AND OUTPUT**

****

Fig 6.1 main code of ARU

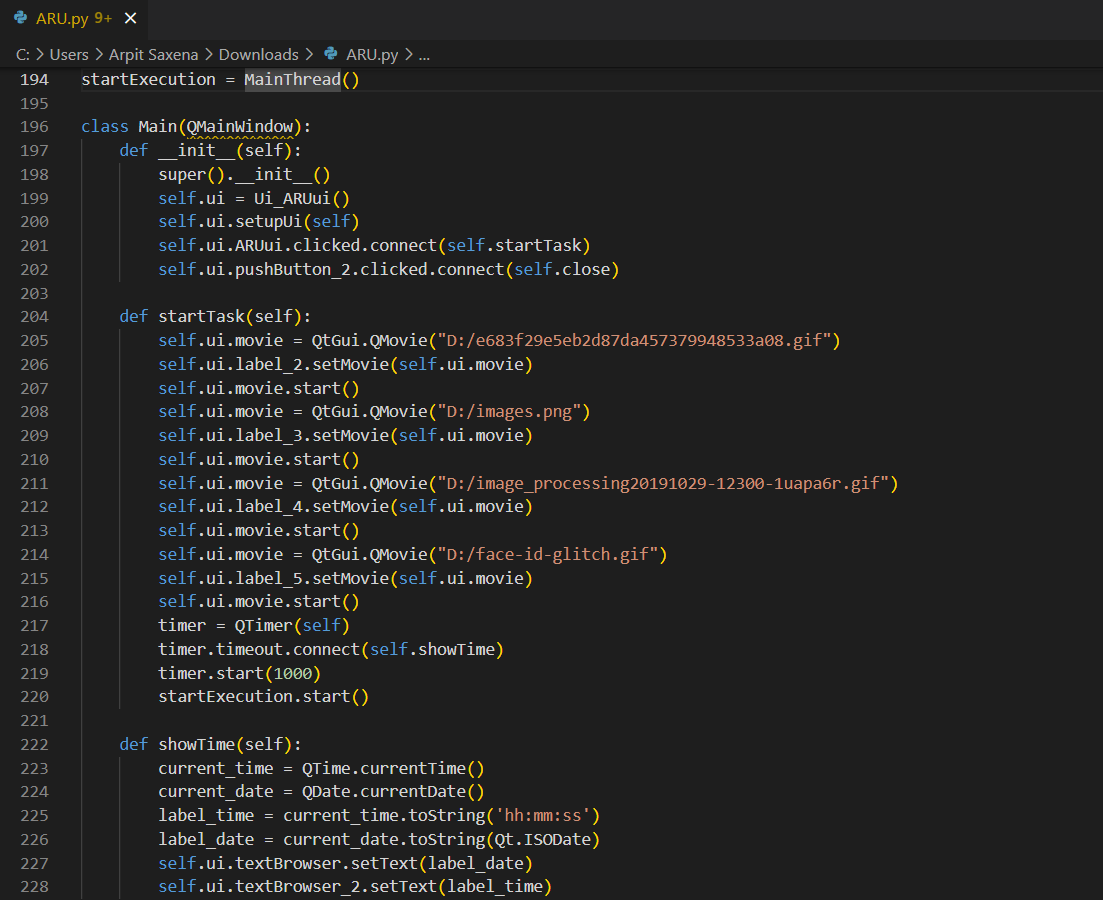


Fig 6.2 GUI code of ARU

A graphics-based operating system interface is that uses icons, menus and a mouse (to click on the icon or pull down the menus) to manage interaction with the system.

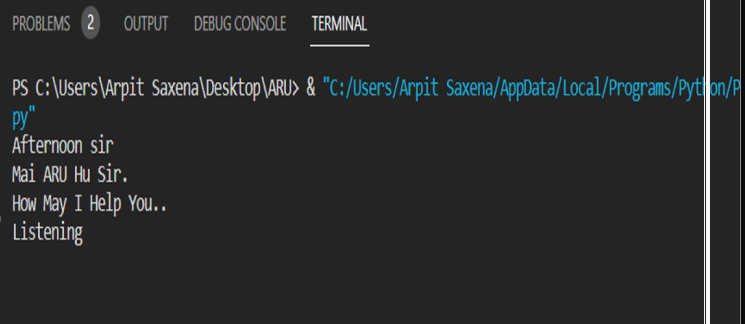
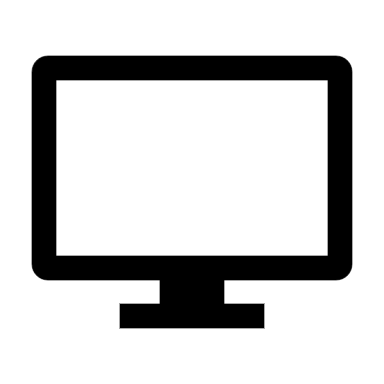


Fig 6.3 interaction with ARU

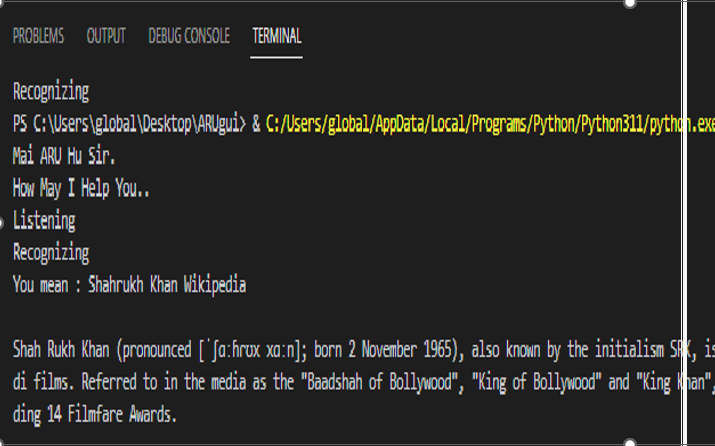


Fig 6.4 Wikipedia result

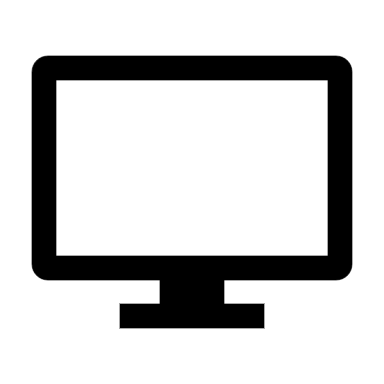
 

Fig 6.5 gui output of ARU

So we have seen the output and the code in this following chapter the futher function and the command are performed in same way so anyone can ask ARU any question and it generate the correct output .

**CHAPTER 7**

**CONCLUSION**

ARU 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:

**Limitations**

* Security is somewhere an issue, there is no voice command encryption in this project.
* Background voice can interfere
* Misinterpretation because of accents and may cause inaccurate results.
* ARU cannot be called externally anytime like other traditional assistants like Google Assistant can be called just by saying, “Ok Google!”

**Scope for future work**

* Make ARU to learn more on its own and develop a new skill in it.
* JARVIS android app can also be developed.
* Make more aru voice terminals.
* Voice commands can be encrypted to maintain security.

**INDIVIDUAL CONTRIBUTION**

In this project all three of us distributed our work as per the conditions and knowledge so the work done by individual is as follows:

**Amit Garg:** write the code for GUI and main program.

**Arpit Saxena:**  making the ppt and helped writing the code.

**Deepak Chaudhary:** made the report file and help in ppt.

All three of us helped each other in making this project and successfully completing the ARU.

**REFRENCES**

* <https://youtu.be/Z1U1BSp_Opc> :AVI Upadhyay YouTube channel help in writing introduction.
* You tube: code with harry writing the code
* Google in importing the modules and further information.
* Fig 4.2 googles