**SIX WEEKS SUMMER TRAINING PROGRAMME**

**AMPHISOFT TECHNOLOGIES PVT LTD (E – Box)**

**Project: Voice Command Assistant “JARVIS”**

**A training report**

Submitted in partial fulfilment of the requirements of the award of degree of

**Bachelor of Technology**

**(Computer Science and Engineering)**

Submitted to

**LOVELY PROFESSIONAL UNIVERSITY**

**Logo

Description automatically generated with low confidencePHAGWARA, PUNJAB**

**From 1 June 2021** to **5 July 2021**

**SUBMITTED BY**

**Name: ABHISHEK SAINI**

**Registration Number: 11902879**

**Shape

Description automatically generated with low confidenceSignature:**

**DECLARATION**

I hereby declare that I have completed my six weeks summer training at **AMPHISOFT TECHNOLOGIES PVT LTD (E – Box)** from **1st** **June 2021** to **5th July 2021** under the guidance of **MR. VIGNEESH AR**. I have declare that I have worked with full dedication during these five to six weeks of training and my learning outcomes fulfil the requirements of training for the award of degree of B. tech (Computer Science & Engineering), **Lovely Professional University, Phagwara.**

**Name: Abhishek Saini**

**Shape

Description automatically generated with low confidenceSignature:**

**Summer training Certificate from AMPHISOFT TECHNOLOGIES PVT. LTD**

**(E - Box):**

****

**Original Image/Copy of Certificate**

Chapter-1. **INTRODUCTION OF THE PROJECT UNDERTAKEN**

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 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 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 Apps/Software 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.

Functionalities of this project include:

1. It can send emails.

2. It can give recent covid-19 updates of any country.

3. It can also give specific data direct from NASA.

4. It can open command prompt, your favourite IDE, notepad etc.

5. It can play music.

6. It can do Wikipedia searches for you.

7. It can open websites like Google, YouTube, etc., in a web browser.

8. It can give weather forecast.

9. It can open camera with a single voice command.

10. It can have some basic conversations.

Now the basic question arises in mind that how it is an AI? The virtual assistant that I have created is like if it is not an A.I, but it is the output of a bundle of the statement. But fundamentally, the main purpose of A.I machines is that it can perform human tasks with the same efficiency or even more efficiently than humans. It is a fact that my virtual assistant is not a very good example of A.I., but it is an A.I and performing basic activities by its own which ultimately is an example of Artificial Intelligence.

**OBJECTIVES**

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 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. In the current scenario, advancement in technologies is 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 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 have some basic conversation.

Tools and technologies used are VS CODE IDE and Python Programming language for making this project, and I created all PY files in VS CODE. Along with this I used following modules and libraries in my project. pyttsx3, Speech Recognition, Date-time, Wikipedia, Smtplib, pywhatkit, pyjokes, bs4 etc.

Chapter-2. **INTRODUCTION OF THE AMPHISOFT TECHNOLOGIES PVT. LTD**

(SUMMER INTERNSHIP COURSE COMPLETION)

* **Company’s Mission/Vison:** “TO BE THE BEST YOU NEED TO BE WITH THE BEST”

The Amphisoft Team consists of people from diverse cultural backgrounds to make the world a better place to live in through our software engines and services.

* **Origin and Growth of Company:** The 11 years journey of Amphisoft from a modest beginning with a 4-member team in a 1-bed room house to an active 200-member team in various locations

**2009:** Amphisoft was found

**2010:** QI – Tester and contest went live

**2011:** Scodhvest Component Went live

**2012:** Design and DBMS components went live. E-box was used in academic lab.

**2013:** E-box was used for Skill Enablement.

**2014:** Cognizant’s Enterprise Deal was signed and GTP Programme was launched in E-box.

**2015:** Early Engagement Programmes

**2016:** Auto Evaluation for VLSI, Verification and Web Frameworks etc.

**2017:** Various Reskilling and Upskilling Programmes Introduced.

**2018:** E-box Online Learning Launched, E – learning Started.

* **Various Departments:** Amphisoft believe in building success out of ideas. We have proved our expertise in converting ideas into products, ideas into reality and ideas into business. We partner with our clients in Developing disruptive and game changing products Maximizing their commercial success Working towards increasing customer delight.

1. **Data And Analysis:** Data Analysis is not just about handling a set of tools. It requires deeper expertise to apply concepts and math to dive into extracting the patterns and knowledge. We hold that expertise to handle and extract pattern from data spread in a Hyper-Dimensional space. We hold thorough knowledge and skills to provide end to end data analytics service from requirement understanding to solution providing to diving deep into data to extracting knowledge packets to summarising and porting the visualisations into your applications. Our expertise is exhibited through our products like S-Box and D-Box. If you are looking for a genuine Data Analytics services, then Amphisoft is the Company you should be looking for. We exhibit our capability through our products and not just through words or claims.
2. **Big Data and Computing:** Many applications lack scalability. Big Data Computing services is not just about Hadoop, it's about distributed computing and storage. It's not just about handling data it's also about the computational blocks remodelled into a distributed processing structure. If you are a company generating or holding VLDBs but slowed down and unscaled applications, then do not hold with it. There's lot more we can do for you. We can remodel your entire application to suit a big data computing environment. Understand the bottlenecks and loopholes that causes scalability and provide you the best architecture to cake your application and business. Our expertise in scalability and big data is exhibited with our products like E-Box.
3. **ML and AI Consultants:** ML and AI is not about using Python or TensorFlow libraries. It's about solving a problem methodically by applying appropriate ML Algorithms and AI techniques. Adding intelligence into your existing application/platform/product requires a blended expertise of application architecting and a thorough research experience in AI & ML domain. Amphisoft provides you the blend with teams formed of knowledge required for real time modelling and realising the model into an intelligent application or AI driven business.
4. **Cloud Computing:** Until and unless you deal with a product that runs on a scale, you would always lack the experience of using the cloud effectively for scalable and computationally intensive applications. And re-architecting your product to suit scalable business and cloud deployment is another challenge that you may face unless you have multiple years of experience on scalable product deployment, cloud architecting and effective cloud management for optimising cost. Amphisoft holds all these experiences through its live scaled up S/W products, S/W engines and S/W platforms. Amphisoft would be the apt company to outsource your software product engineering needs and gather consultancies for quickening your product development process.

Chapter-3. **BRIEF DESCRIPTION OF THE WORK DONE**

**Technology Learnt**

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.

## **History of Python**

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, Small Talk, and UNIX shell and other scripting languages.

Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL).

Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

**Reason for choosing this technology**

* **Easy-to-learn** − Python has few keywords, simple structure, and a clearly defined syntax. This allows us to pick up the language quickly.
* **Easy-to-read** − Python code is more clearly defined and visible to the eyes.
* **Easy-to-maintain** − Python's source code is fairly easy-to-maintained.
* **A broad standard library** − Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
* **Interactive Mode** − Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
* **Portable** − Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
* **Extendable** − we can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
* **Databases** − Python provides interfaces to all major commercial databases.
* **GUI Programming** − Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
* **Scalable** − Python provides a better structure and support for large programs than shell scripting.

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 IDE with the help of a single voice command. In the current scenario, advancement in technologies is 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 Python in every field is doing marvellous job. As I want to make my career in AI/ML in future and except python there is no language better than this which have easy access to AI/ML concepts for the basic start.

**Learning Outcome**

As I am beginner in programming, and I was not much aware about these programming languages during my school days. But as I am pursuing my graduation in computer science and engineering so for me it is very important to learn a computer language. As I chose python which is a very interesting language I have come across after learning from **E – Box** Platform. Most of my basics are clear throughout this six week training programme, now I can work on my python projects with more confidence.

**Existing System**

We are familiar with many existing voice assistants like Alexa, SIRI, Google Assistant, Cortana which uses concept of language processing, and voice recognition. They listen 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.

**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. Jarvis 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. 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, pyQt etc. 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 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 regular covid-19 updates of any country, It can also fetch the data from NASA, It can have some basic conversation and much more.

**Design: Data flow of JARVIS Project**

The system is designed using the concept of Artificial Intelligence and with the help of necessary packages of Python. Python provides many libraries and packages to perform the tasks. The details of these packages are mentioned in above pages of this report.

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

**Software Requirement Analysis**

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, SpeechRecognition, Datetime, Wikipedia, Smtplib, pywhatkit, pyjokes, pyQt etc.

**Visual Studio Code**

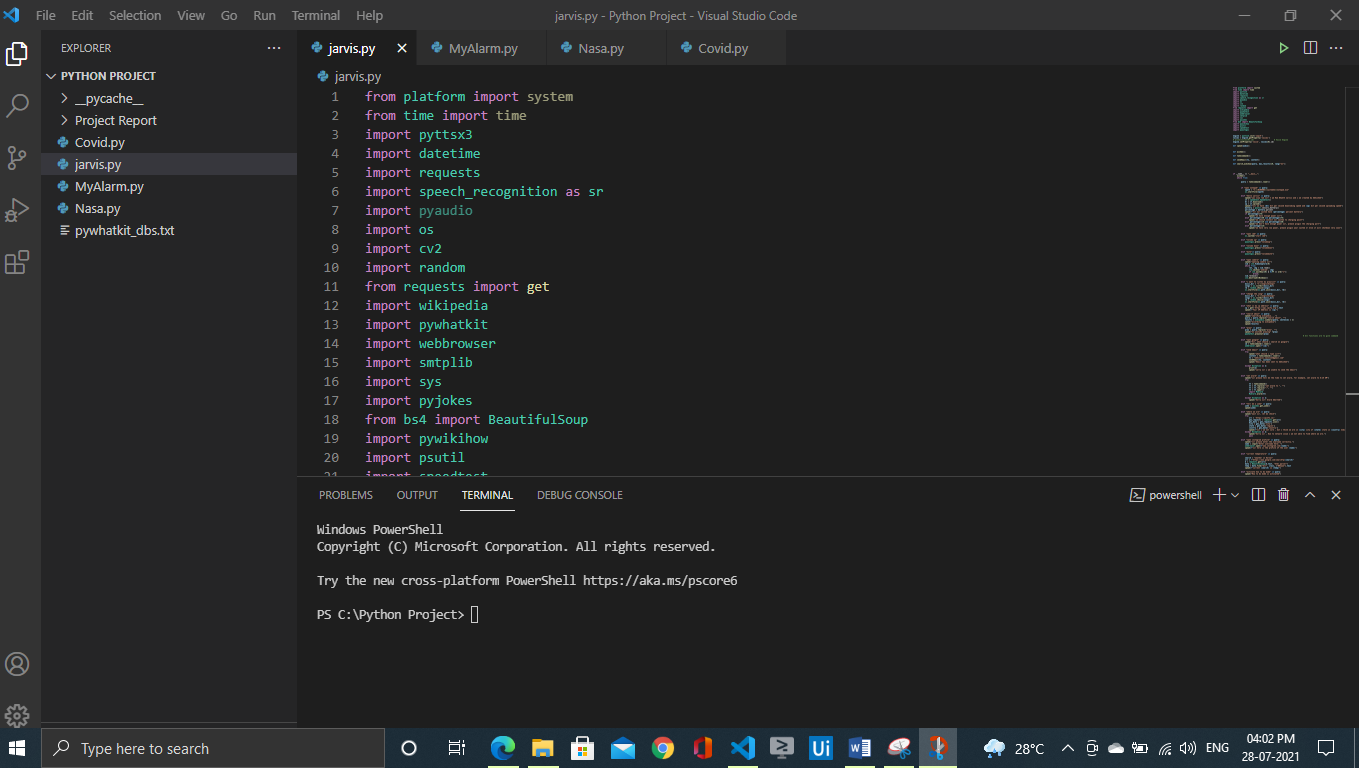
Visual Studio Code combines the simplicity of a source code editor with powerful developer tooling, like IntelliSense code completion and debugging.

First and foremost, it is an editor that gets out of your way. The delightfully frictionless edit-build-debug cycle means less time fiddling with your environment, and more time executing on your ideas.

At its heart, Visual Studio Code features a lightning-fast source code editor, perfect for day-to-day use. With support for hundreds of languages, VS Code helps you be instantly productive with syntax highlighting, bracket-matching, auto-indentation, box-selection, snippets, and more. Intuitive keyboard shortcuts, easy customization and community-contributed keyboard shortcut mappings let you navigate your code with ease.

And when the coding gets tough, the tough get debugging. Debugging is often the one feature that developers miss most in a leaner coding experience, so we made it happen. Visual Studio Code includes an interactive debugger, so you can step through source code, inspect variables, view call stacks, and execute commands in the console.

VS Code also integrates with build and scripting tools to perform common tasks making everyday workflows faster. VS Code has support for Git so you can work with source control without leaving the editor including viewing pending changes diffs.



**Image: VS Code Interface**

**PYTHON LIBRARIES**

In JARVIS following python libraries were used:

1**. pyttsx3:** It is a python library which converts text to speech.

2. **SpeechRecognition:** It is a python module which converts speech to text.

3. **pywhatkit:** It is python library to open Apps in Python.

4. **Datetime:** This library provides us the actual date and time.

5. **Wikipedia:** It is a python module for searching anything on Wikipedia.

6. **Smtplib:** Simple mail transfer protocol that allows us to send mails and to route mails between mail servers.

7. **Pyjokes:** It is a python library which contains lots of interesting jokes in it.

8. **Webbrowser:** It provides interface for displaying web-based documents to users.

9**. os:** It represents Operating System related functionality.

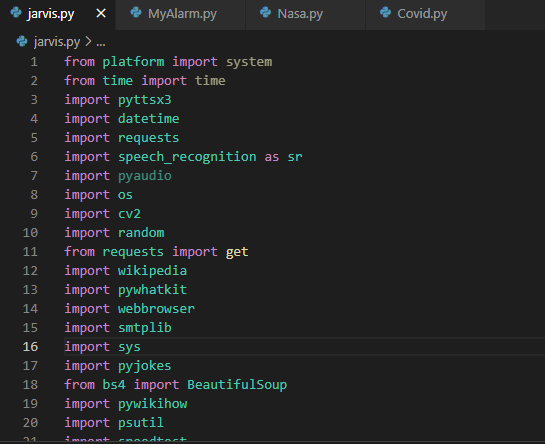
10**. sys:** It allows operating on the interpreter as it provides access to the variables and functions that usually interact strongly with the interpreter.

11**. pyautogui:** It is use to take access of automatic keys from keyboard in this project.

12. **pywikihow:** The module use to access the how to do mode.

13. **bs4:** Use to access the APIs of the websites direct to the software.

14**. psutil:** It is use to access some informative features direct from CPU, battery of the system.

****

**Image: Imported Modules**

Chapter-4. **DOCUMENTATION OF PYTHON LIBRARIES USED IN “JARVIS” PROJECT**

1. **pyttsx3 :** is a text-to-speech conversion library in Python. Unlike alternative libraries, it works offline and is compatible with both Python 2 and 3. An application invokes the pyttsx3.init() factory function to get a reference to a pyttsx3. Engine instance. it is a very easy to use tool which converts the entered text into speech.  
   The pyttsx3 module supports two voices first is female and the second is male which is provided by “sapi5” for windows.
2. **Speech Recognition:** The first component of speech recognition is, of course, speech. Speech must be converted from physical sound to an electrical signal with a microphone, and then to digital data with an analog-to-digital converter. Once digitized, several models can be used to transcribe the audio to text. Speech Recognition will work out of the box if all you need to do is work with existing audio files. Specific use cases, however, require a few dependencies. Notably, the PyAudio package is needed for capturing microphone input.
3. **pywhatkit:** Python offers numerous inbuilt libraries to ease our work. Among them **pywhatkit** is a Python library for sending WhatsApp messages at a certain time, it has several other features too.

Following are some features of pywhatkit module:

1. Send WhatsApp messages.
2. Play a YouTube video.
3. Perform a Google Search.
4. Get information on a particular topic.

The pywhatkit module can also be used for converting text into handwritten text images.

**4.** **Datetime:** In Python, date and time are not a data type of its own, but a module named datetime can be imported to work with the date as well as time. Datetime module comes built into Python, so there is no need to install it externally.   
Datetime module supplies classes to work with date and time. These classes provide a few functions to deal with dates, times and time intervals. Date and datetime are an object in Python, so when you manipulate them, you are manipulating objects and not string or timestamps.

5. **Wikipedia:** The Internet is the single largest source of information, and therefore it is important to know how to fetch data from various sources. And with Wikipedia being one of the largest and most popular sources for information on the Internet.

**Wikipedia** is a multilingual online encyclopaedia created and maintained as an open collaboration project by a community of volunteer editors using a wiki-based editing system.

6. **Smtplib:** Simple Mail Transfer Protocol (SMTP) is a protocol, which handles sending e-mail and routing e-mail between mail servers.

Python provides **smtplib** module, which defines an SMTP client session object that can be used to send mail to any Internet machine with an SMTP or ESMTP listener daemon.

An SMTP object has an instance method called **sendmail**, which is typically used to do the work of mailing a message. It takes three parameters −

* The *sender* − A string with the address of the sender.
* The *receivers* − A list of strings, one for each recipient.
* The *message* − A message as a string formatted as specified in the various RFCs

7. **Pyjokes:** Python supports creation of random jokes using one of its libraries. Let us explore it a little more, Pyjokes is a python library that is used to create one-line jokes for programmers. Informally, it can also be referred as a fun python library which is pretty simple to use.

8. **Webbrowser:** In Python, webbrowser module provides a high-level interface which allows displaying Web-based documents to users. The webbrowser module can be used to launch a browser in a platform-independent manner.

9. **OS Module:** 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.

10. **sys:** The sys module in Python provides various functions and variables that are used to manipulate different parts of the Python runtime environment. It allows operating on the interpreter as it provides access to the variables and functions that interact strongly with the interpreter.

11. **pyautogui:** PyAutoGUI lets your Python scripts control the mouse and keyboard to automate interactions with other applications.

PyAutoGUI has several features:

* Moving the mouse and clicking in the windows of other applications.
* Sending keystrokes to applications (for example, to fill out forms).
* Take screenshots, and given an image (for example, of a button or checkbox), and find it on the screen.
* Locate an application’s window, and move, resize, maximize, minimize, or close it (Windows-only, currently).
* Display alert and message boxes.

**12. bs4:** [Beautiful Soup](http://www.crummy.com/software/BeautifulSoup/) 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.

These instructions illustrate all major features of Beautiful Soup 4, with examples. I show you what the library is good for, how it works, how to use it, how to make it do what you want, and what to do when it violates your expectations.

13. **psutil:** psutil (python system and process utilities) is a cross-platform library for retrieving information on running processes and system utilization (CPU, memory, disks, network) in Python. It is useful mainly for system monitoring, profiling and limiting process resources and management of running processes. It implements many functionalities offered by command line tools such as: *ps, top, lsof, netstat, ifconfig, who, df, kill, free, nice, ionice, iostat, iotop, uptime, pidof, tty, taskset, pmap*. It currently supports Linux, Windows, OSX, FreeBSD and Sun Solaris, both 32-bit and 64-bit architectures, with Python versions from 2.6 to 3.4 (users of Python 2.4 and 2.5 may use [2.1.3](https://pypi.python.org/pypi?name=psutil&version=2.1.3&:action=files) version). [PyPy](http://pypy.org/) is also known to work.

**A picture containing diagram

Description automatically generated**

**HELLO SIR! I AM YOUR COMMAND ASSISTANT JARVIS**

**I DO EXIST IN REAL WORLD AND MAKE YOUR TASK EASY…**

**Implementation**

JARVIS, 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. Jarvis 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.

**REAL WORLD APPLICATION**

1. **Saves time:** JARVIS is a desktop voice assistant which works on the voice command offered to it, it can do voice searching, voice-activated device control and can let us complete a set of tasks.
2. **Conversational interaction:** It makes it easier to complete any task as it automatically do 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.
3. **Reactive nature:** The desktop 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.
4. **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.
5. **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.

**DATA IMPLEMENTATION AND PROGRAM EXECUTION**

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:

**LIBRARIES AND PACKAGES**

1. **pyttsx3:** It is a python library which converts text to speech.
2. **SpeechRecognition:** It is a python module which converts speech to text.
3. **pywhatkit:** It is python library to open Apps in Python.
4. **Datetime:** This library provides us the actual date and time.
5. **Wikipedia:** It is a python module for searching anything on Wikipedia
6. **Smtplib:** Simple mail transfer protocol that allows us to send mails and to route mails between mail servers.
7. **Pyjokes:** It is a python libraries which contains lots of interesting jokes in it.
8. **Webbrowser:** It provides interface for displaying web-based documents to users.
9. **os:** It represents Operating System related functionality.
10. **sys:** It allows operating on the interpreter as it provides access to the variables and functions that usually interact strongly with the interpreter.
11. **pyautogui:** It is use to take access of automatic keys from keyboard in this project.
12. **pywikihow:** The module use to access the how to do mode.
13. **bs4:** Use to access the API’s of the websites direct to the software.
14. **psutil:** It is use to access some informative features direct from CPU, battery of the system.
15. **Pyaudio:** With PyAudio, we can easily use Python to play and record audio on a variety of platforms.

**FUNCTIONS**

1. **takeCommand ():** The function is used to take the command as input through microphone of user and returns the output as string.
2. **wishMe ():** This function greets the user according to the time like Good Morning, Good Afternoon and Good Evening.
3. **taskExecution():** This is the function which contains all the necessary task execution definition like sendEmail(), NASA and many conditions in if condition like “open google”, “open notepad”, “search on Wikipedia” ,”play music” and “open command prompt” etc.

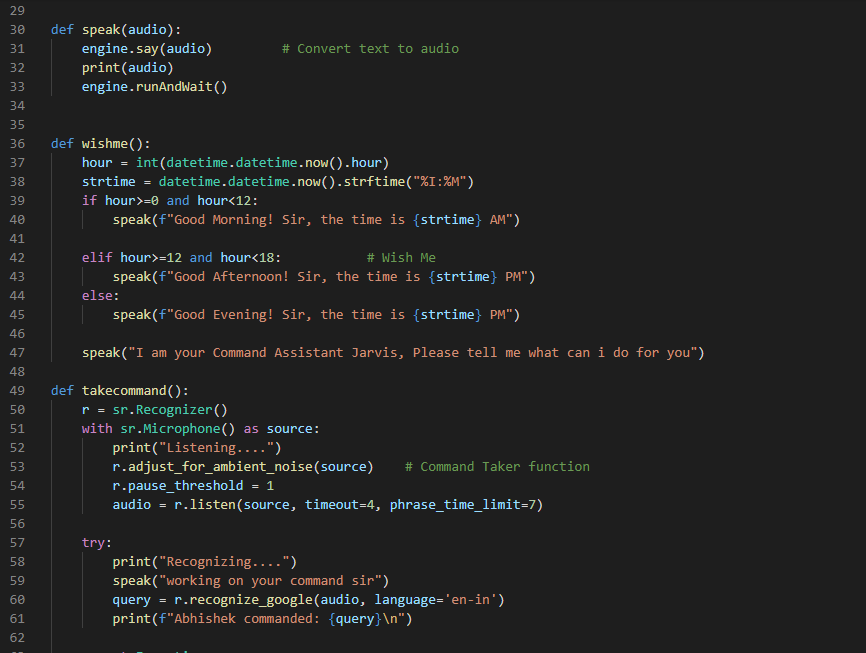
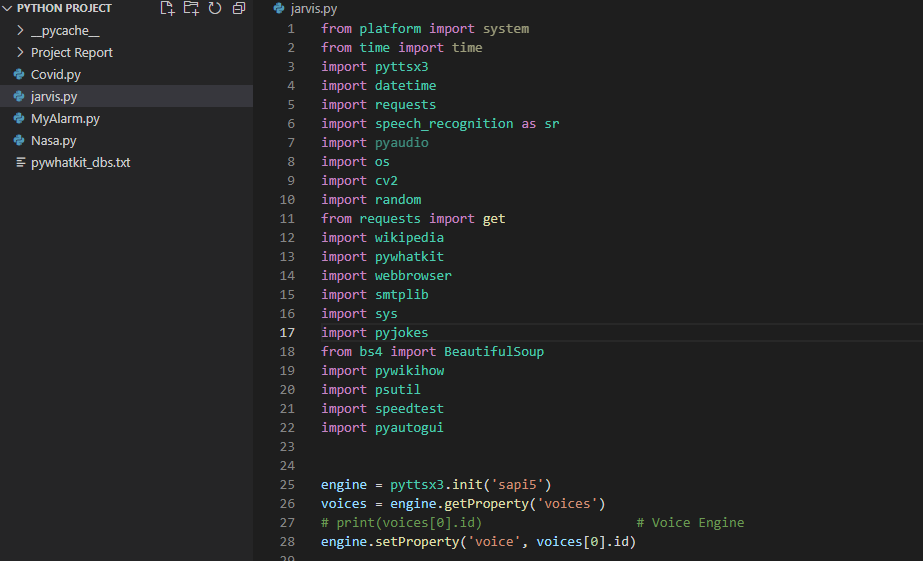
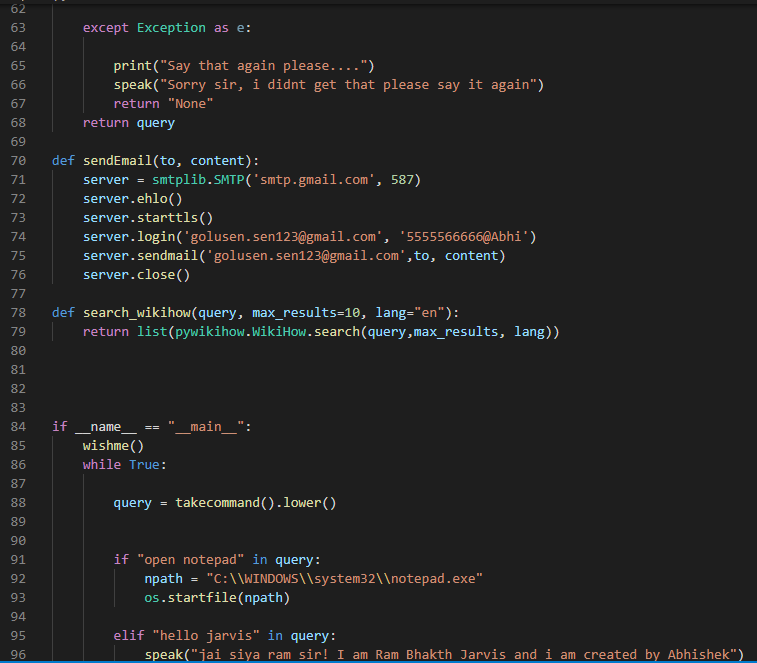
**Source Code**

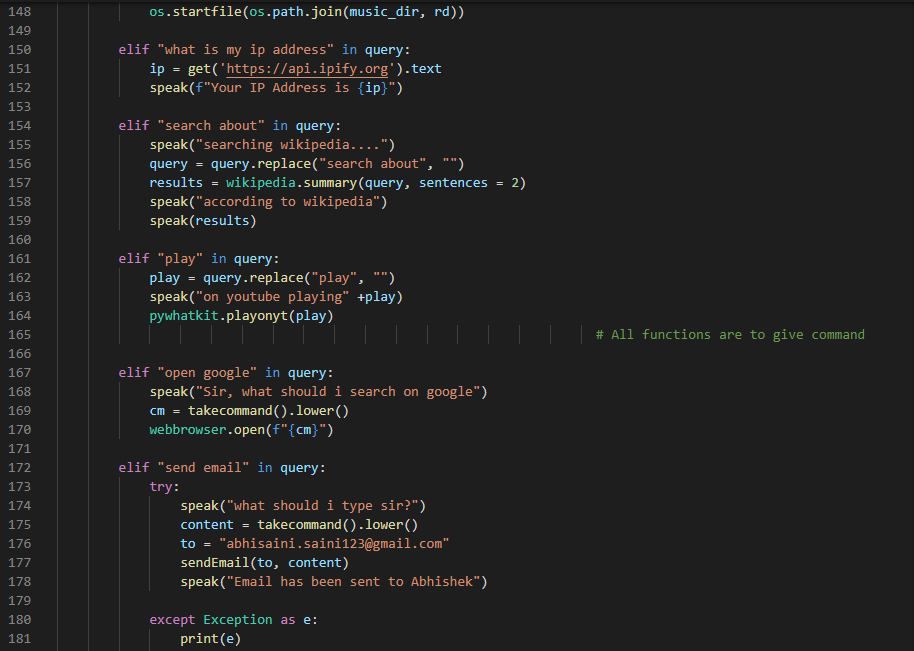
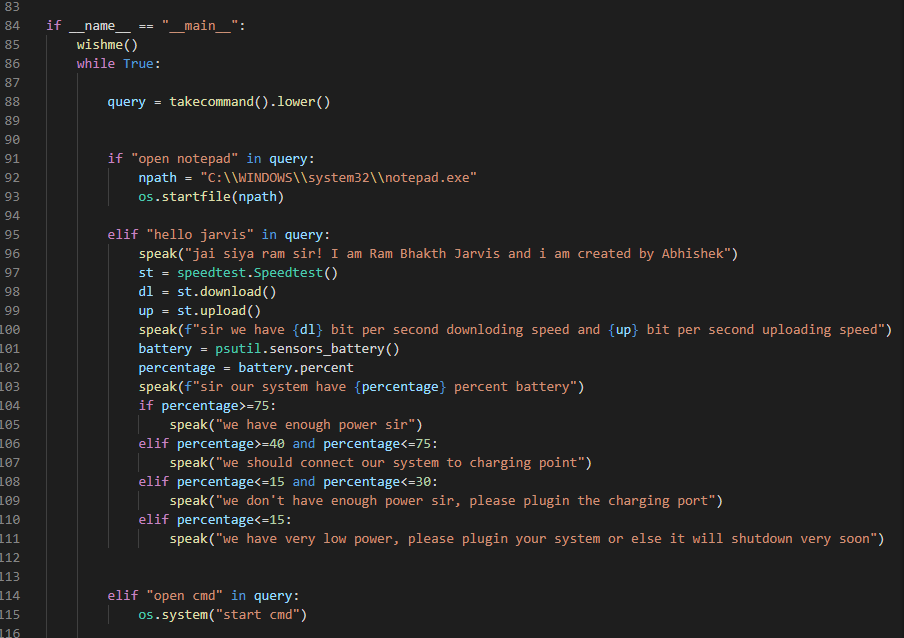


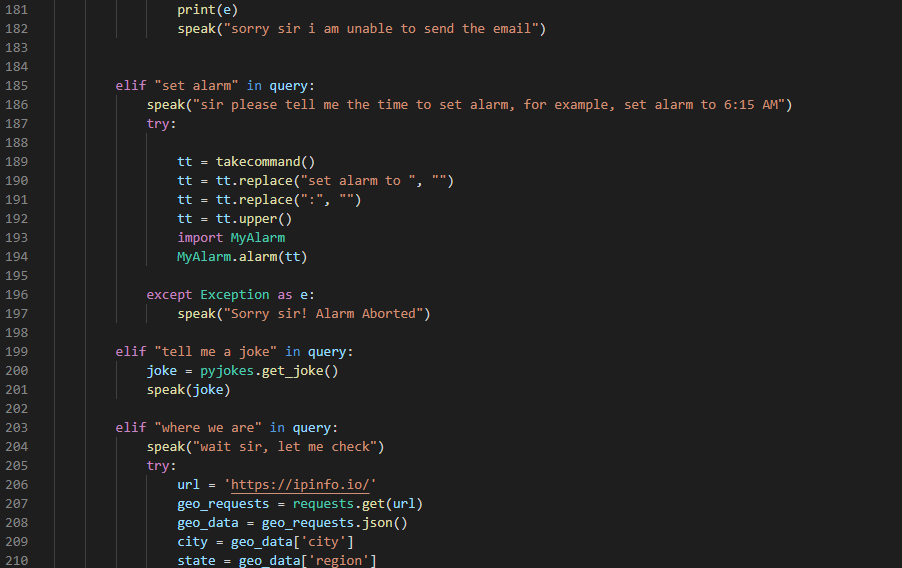
**SCREEN SHOTS**

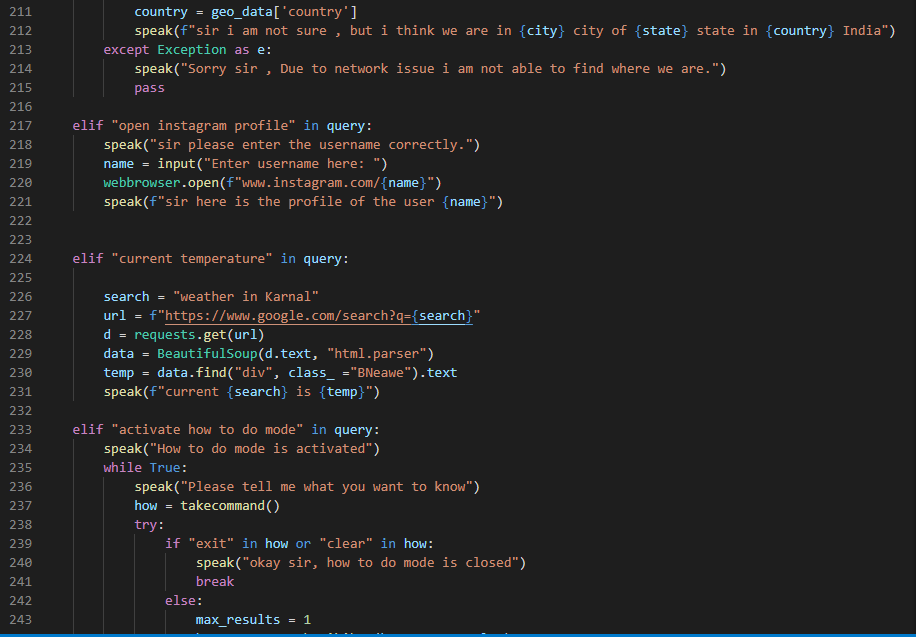
PYTHON PROGRAMME “JARVIS” PROJECT

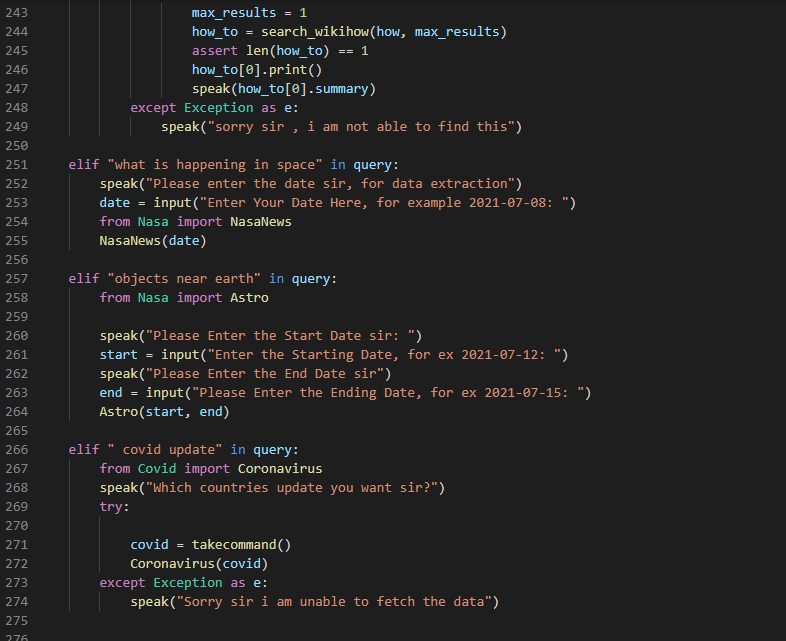


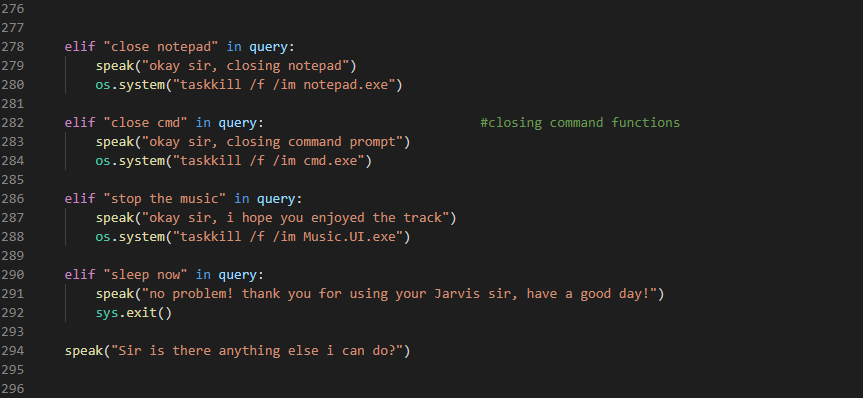
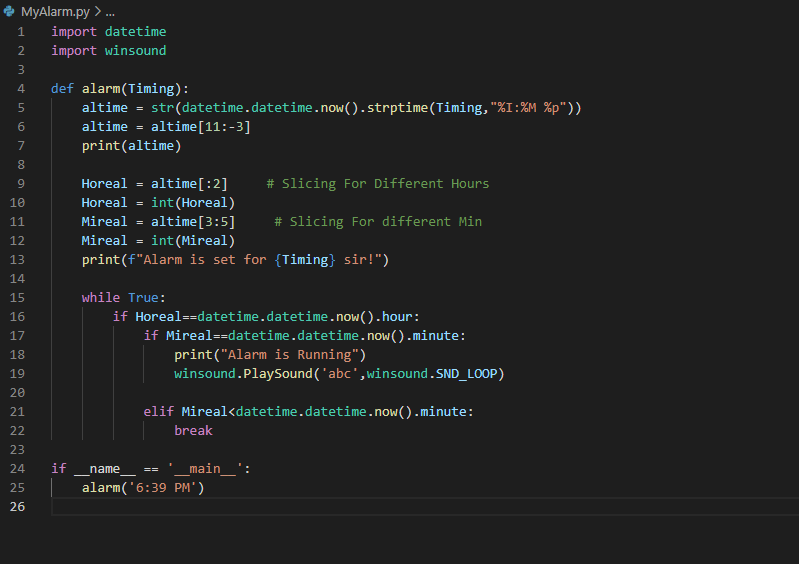
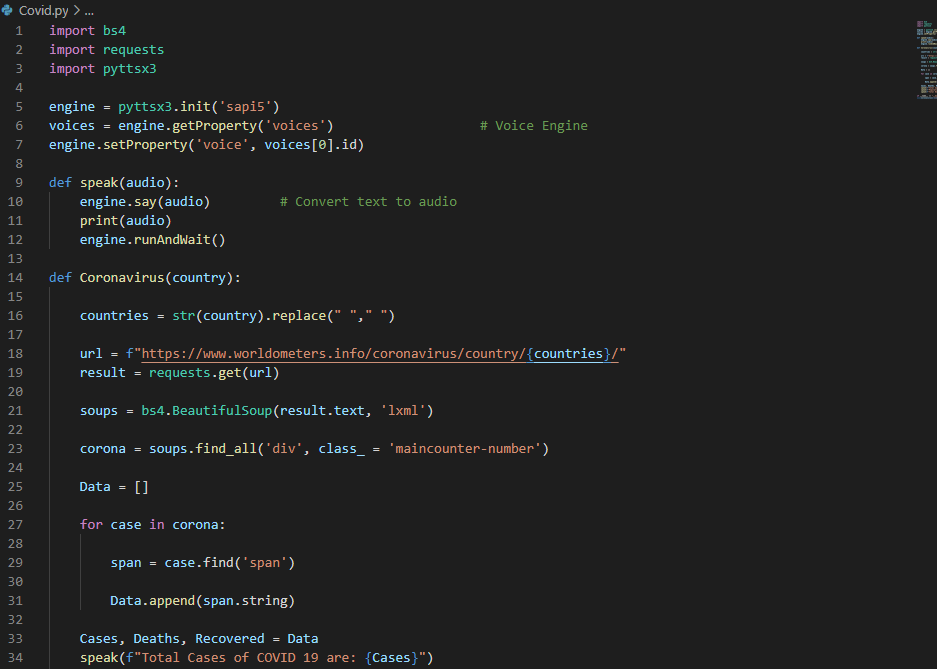
****

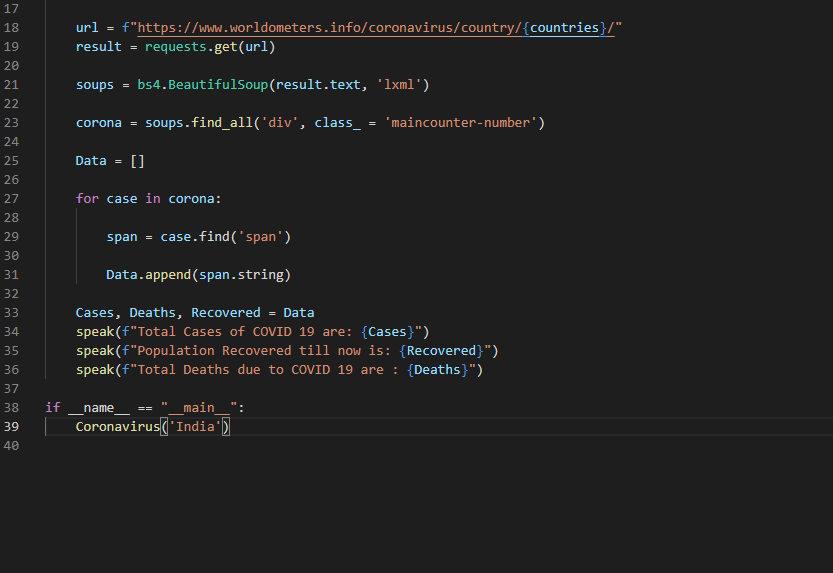
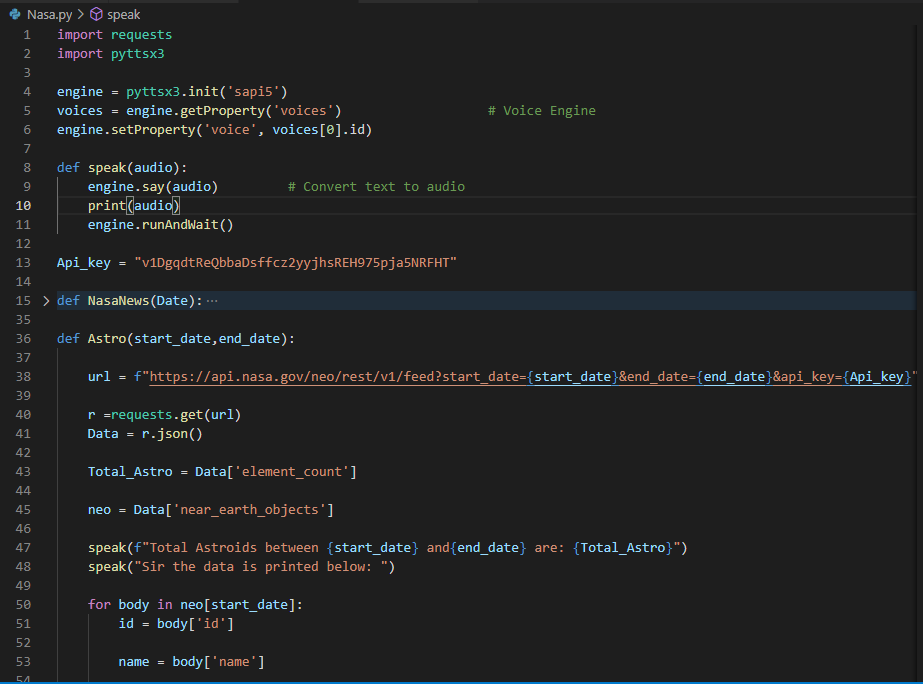
****

****

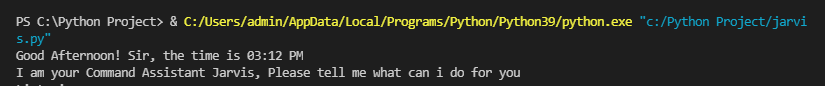
****

****

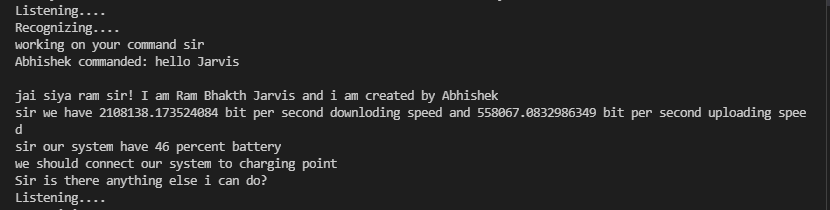
****

****

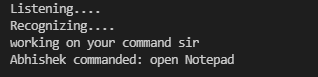
**Some Input / Output Screenshots:**

****

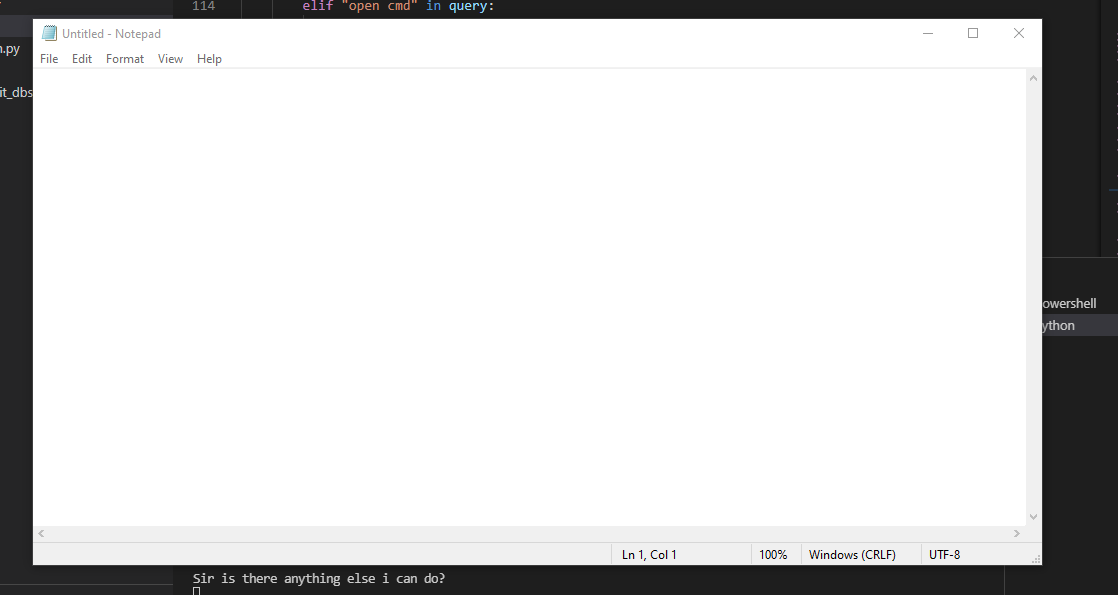
* Wish me Function will work in starting.

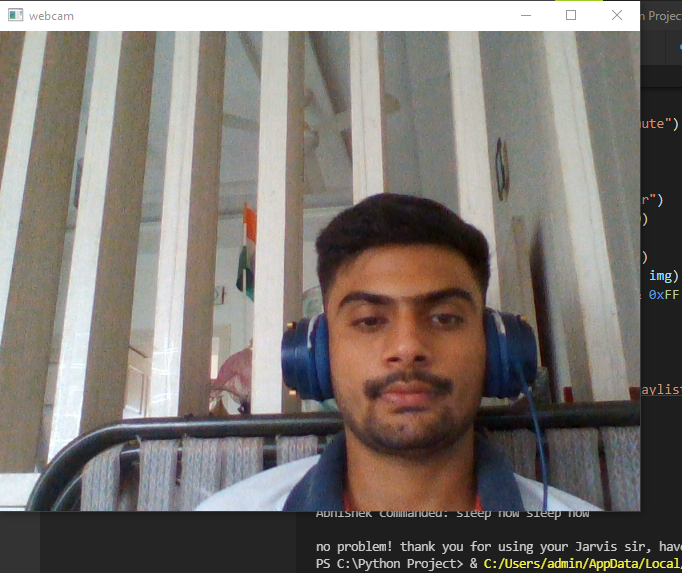
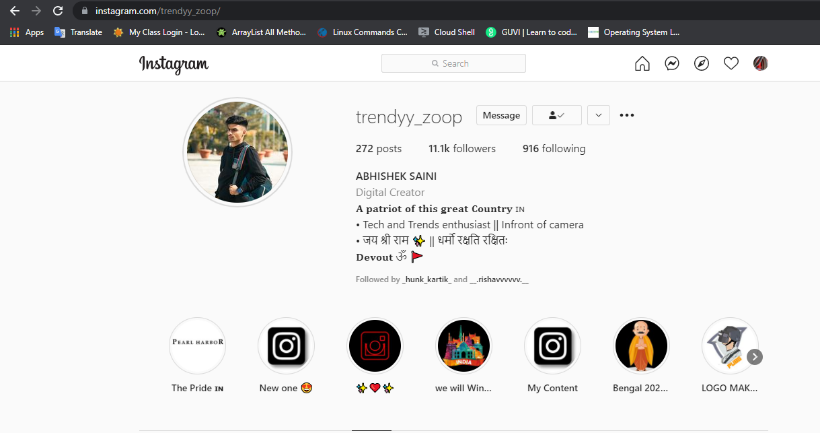


* With command “Hello Jarvis” it will speak and print some random conversation.

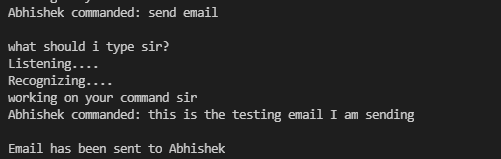


* It will again ask for next task and as here user given “Open Notepad” command, so it will open notepad for the user.



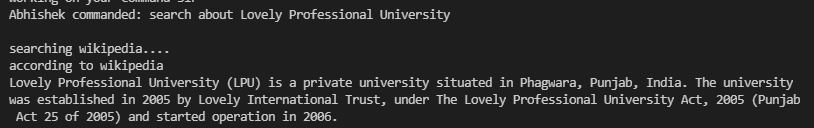
 

* Here user commanded “Open Camera” it will automatically open the camera.
* Here user commanded “Open instagram Profile” it will ask to type the username and it will find out results for the user.

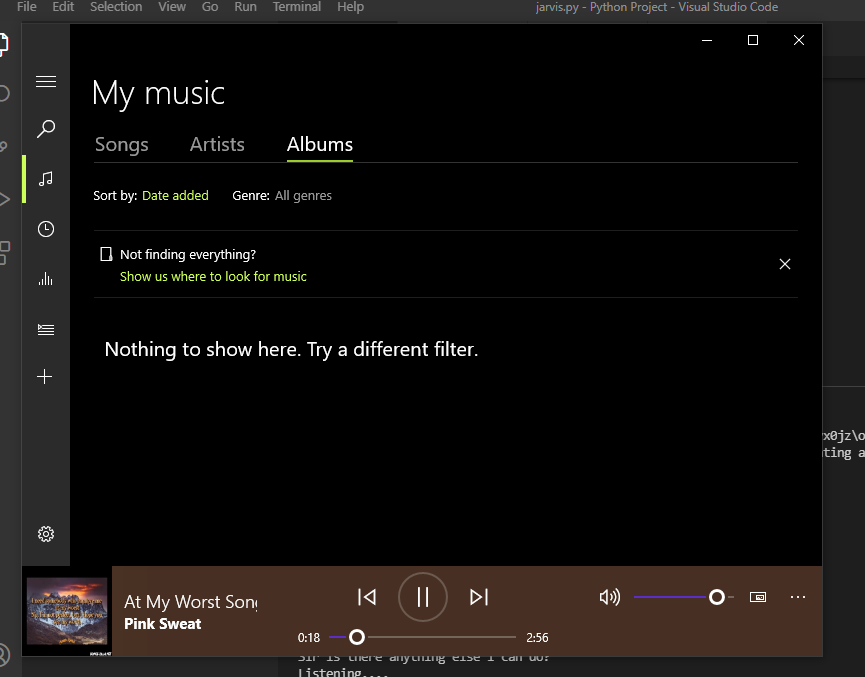




* User will command “Send Email” it will ask what should I type sir? Then it will recognize the text and send it to the mail id defined in the function.

****

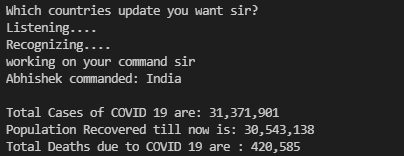
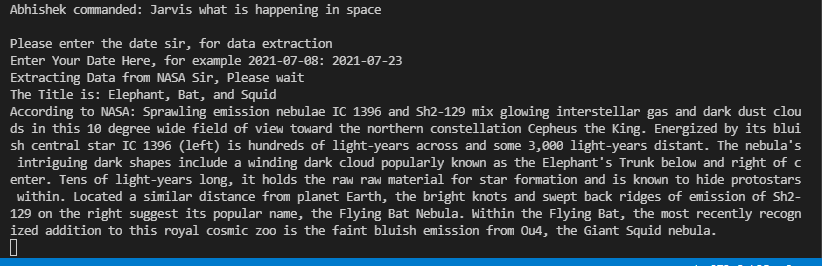
* Wikipedia Input / Output. It will speak and display the result at the same time.

****

Software will play the music, stop the music Volume up, Volume Down by just giving Jarvis a command.

Example: “I want to listen my playlist”, “Volume Up / down / Mute”, “Stop the music”, “change the song” etc.

* **Some more Commands and outputs in the screenshots:**

****

**Text

Description automatically generatedText

Description automatically generatedText

Description automatically generatedText

Description automatically generated**

**Text

Description automatically generatedText

Description automatically generatedText

Description automatically generated**

**Text

Description automatically generatedText

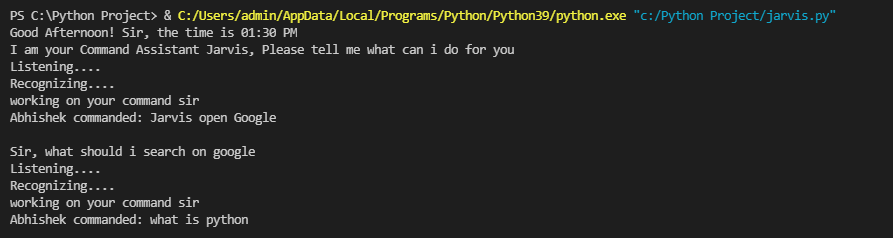
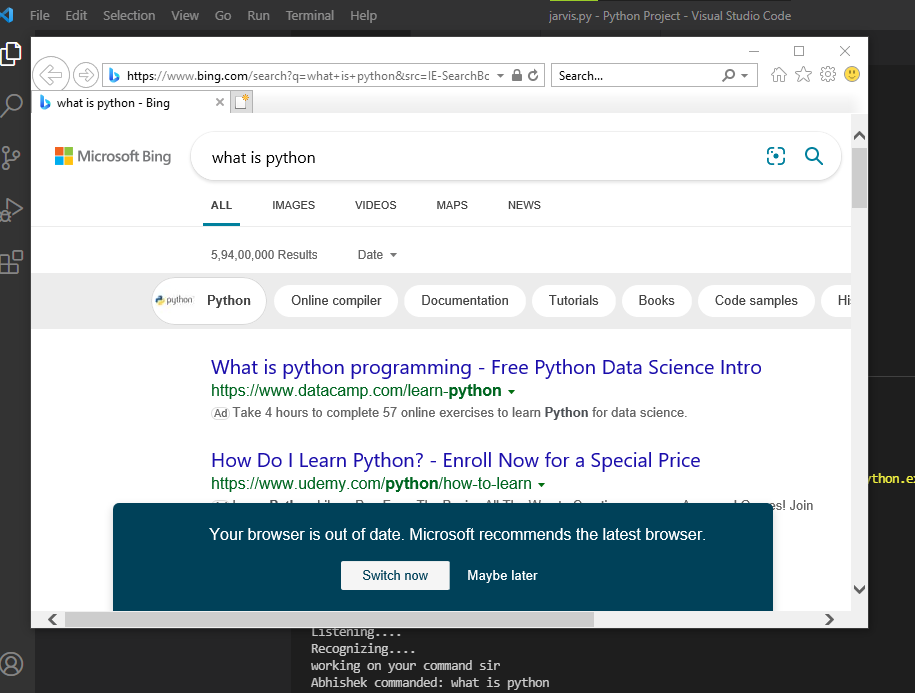
Description automatically generatedText

Description automatically generated**

**System testing**

The system testing is done on fully integrated system to check whether the requirements are matching or not. The system testing for JARVIS desktop assistant focuses on the following four parameters:

1. **FUNCTIONALITY**

****In this we check the functionality of the system whether the system performs the task which it was intended to do. To check the functionality each function was checked and run, if it is able to execute the required task correctly then the system passes in that particular functionality test. For example to check whether JARVIS can search on Google or not, as we can see in the image below user said “Open Google”, then Jarvis asked, ”What should I search on Google, sir?” then user said, “What is Python”, Jarvis open Google and searched for the required input.

**Tested Ok**

1. **USABILITY**

Usability of a system is checked by measuring the easiness of the software and how user friendly it is for the user to use, how it responses to each query that is being asked by the user. It makes it easier to complete any task as it automatically do 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. The desktop 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. The main application of it can be its multitasking ability. It can ask for continuous instruction one after other until the user “QUIT” it. 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.

1. **SECURITY**

The security testing mainly focuses on vulnerabilities and risks. As JARVIS is a local desktop application, hence there is no risk of data breaching through remote access. The software is dedicated to a specific system so when the user logs in, it will be activated.

1. **STABILITY**

Stability of a system depends upon the output of the system, if the output is bounded and specific to the bounded input then the system is said to be stable. If the system works on all the poles of functionality then it is stable.

Final Chapter- **CONCLUSION**

JARVIS 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**

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. JARVIS cannot be called externally anytime like other traditional assistants like Google Assistant can be called just by saying, “Ok Google!”

* **SCOPE FOR FUTURE WORK**

1. Make JARVIS to learn more on its own and develop a new skill in it.

2. JARVIS android app can also be developed.

3. Make more Jarvis voice terminals.

4. Voice commands can be encrypted to maintain security.

5. As I am going to learn Machine Learning in upcoming days, I will make it more advance with ML concepts (Example: Face Detection, Face lock unlock, Home Automation etc.).

6. Make GUI for the software (I will do for sure).

**REFERENCES**

The project titled **“Jarvis” – a voice command assistant** was designed by me individually. From installing of all the packages, importing, creating all the necessary functions was all done by me individually. I, myself have done all the research before making this project, designed the requirement documents for the requirements and functionalities, wrote synopsis and all the documentation, code and made the project in such a way that it is deliverable at each stage. I have written the complete code in Python language and in **VS Code** IDE from where it was very easy to install the packages and libraries, I have created the functions like **takeCommand(),** **wishMe()** etc. which has the following functionalities, like **takeCommand()** which is used to take the command as input through microphone of user and returns the output as string, **wishMe()** that greets the user according to the time like Good Morning, Good Afternoon and Good Evening and taskexecution() which contains all the necessary task execution definition like **sendEmail()** and many conditions in if condition like “open Google”, “open notepad”, “search on Wikipedia” ,”play music” and “open command prompt” etc. While making this project I realized that with the advancement **JARVIS** can perform any task with same effectiveness or can say more effectively than us. I have gone through various resources for each and individual concept of python.

As what I have learnt on **E - Box**, how to apply those concepts in my project. I used **YOUTUBE** and **GOOGLE** on which various people have shared their ideas, so with the mixture of different sources and my own ideas, I have successfully created this wonderful Python project. At last, I have updated my report and completed it by attaching all the necessary screen captures of inputs and outputs, mentioning the limitations and scope in future of this project.