Abstract

“Jarvis” was main character of Tony’s Stark’s life assistant in Movies Iron Man. Unlike original comic in which Jarvis was Stark’s human butler, the movie version of Jarvis is an intelligent computer that converses with stark, monitors his household and help to build and program his superhero suit.

In this modern era, day to day life became smarter and interlinked with technology. We already know some voice assistance like google, Siri. etc. Now in our voice assistance system. This project works on voice input and give output through voice and displays the text on the screen. The main agenda of our voice assistance makes people smart and give instant and computed results. The voice assistance takes the voice input through our microphone and it converts our voice into computer understandable language gives the required solutions and answers which are asked by the user. This assistance connects with the world wide web to provide results that the user has questioned. Natural Language Processing algorithm helps computer machines to engage in communication using natural human language in many forms. The project aims to develop a personal-assistant for Windows -based systems. Jarvis draws its inspiration from virtual assistants like 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 or using keyboard input. As a personal assistant, Jarvis assists the end-user with day-to-day activities like general human conversation, searching queries in google, bing or yahoo, searching for videos, retrieving images, live weather conditions, word meanings, searching for medicine details, health recommendations based on symptoms and reminding the user about the scheduled events and tasks.

**Table of Content**

[**List of Figures**](#_Toc35974092)

[**List of Tables**](#_Toc35974093)

[**Abbreviations**](#_Toc35974093)

Chapter 1: Introduction

[1.2 Goal](#_Toc35974094)

[1.3 Objective](#_Toc35974095)

[1.4 Methodology](#_Toc35974096)

[1.5 Role](#_Toc35974097)

[1.6 Contribution of Project](#_Toc35974098)

[1.6.1 Market Potential:](#_Toc35974099)

[1.6.2 Innovativeness](#_Toc35974100)

[1.6.3 Usefulness](#_Toc35974101)

[1.7 Report Organisation](#_Toc35943756)

[Chapter 2:Requirement Engineering](#_Toc35974102)

2.1[User Roles & Responsibilities](#_Toc35974102)

[2.2 Requirement Collection](#_Toc35974103)

[2.1.1 Functional Requirements](#_Toc35974104)

[2.1.2 Non-Functional Requirements](#_Toc35974105)

[Chapter 3:Analysis and Design](#_Toc35974108)

[3.1 Use-case Diagrams](#_Toc35974106)

[3.2 ActivityDiagrams](#_Toc35974107)

[3.3 System Architecture](#_Toc35974113)

[3.4 Sequence Diagram](#_Toc35974113)

[3.5 Class Diagram](#_Toc35974113)

[3.6 E-R Diagram](#_Toc35974113)

[3.7 Database Design](#_Toc35974113)

[3.7.1 Schema Definitions](#_Toc35974113)

[3.7.2 Integrity Contraints](#_Toc35974113)

[Chapter 4: Methodology](#_Toc35974109)

[4.1 Dataset Discussion and Interpretation](#_Toc35974109)

[4.1 Proposed Algorithm](#_Toc35974109)

[4.1 Tools Required](#_Toc35974109)

[Chapter 5:Construction](#_Toc35974109)

[4.1 Implementation and Testing](#_Toc35974113)

[4.1.1 Implemented Classes](#_Toc35974110)

[4.1.2Implemented Functions](#_Toc35974110)

4.1.3Logical Structure of Data…………………………………………………………………

[4.2 Validation and Verification](#_Toc35974111)

[4.3 Testing Approach](#_Toc35974111)

[4.3.1 Unit Testing](#_Toc35974112)

[4.3.1.1 Test Cases](#_Toc35974112)

[4.3.2 Integration Testing](#_Toc35974112)

[4.3.2.1 Test Cases](#_Toc35974113)

Chapter5: Result and Discussion…………………………………………………………………………

Conclusion& Future Works………………………………………………………………………………

Appendix A (Project Synopsis)…………………………………………………………………………..

Appendix B (Guide Interaction Report)………………………………………………………………….

Appendix C (Project Snapshots)…..……………………………………………………………………..

References………………………………………………………………………………………………...

CHAPTER -1

INTRODUCTION

Today the development of artificial intelligence (AI) systems that can organize a natural human-machine interaction (through voice, communication, gestures, facial expressions, etc.) are gaining in popularity. One of the most studied and popular was the direction of interaction, based on the understanding of the machine by the machine of the natural human language. It is no longer a human who learns to communicate with a machine, but a machine learns to communicate with a human, exploring his actions, habits, behaviour and trying to become his personalized assistant.

This system is designed to be used efficiently on desktops. Personal assistants software improves user productivity by managing routine tasks of the user and by providing information from an online source to the user.

This project was started on the premise that there is a sufficient amount of openly available data and information on the web that can be utilized to build a virtual assistant that has access to making intelligent decisions for routine user activities.

Voice based personal assistants have gained a lot of popularity in this era of smart homes and smart devices. These personal assistants can be easily configured to perform many of your regular tasks by simply giving voice commands. Google has popularized voice-based search that is a boon for many like senior citizens who are not comfortable using the keypad/keyboard.

This Voice enabled personal assistant can be implemented byusing technologies like Speech-to-Text and Text-to-Speech,and can be integrated with other functionalities as well,depending on our requirement.

For building any voice based assistant you need two main functions. One for listening to your commands and another to respond to your commands. Along with these two core functions, you need the customized instructions that you will feed your assistant.

1.2 Goal:A goalrepresents the visions about yourself, your business and your future. Goals, however, require action to be achieved and when it is set correctly it forms an integral part of your business’s strategic plan.

Jarvis goals on following:

* ***Increase something*** – your income, nr of clients, customer satisfaction, effectiveness
* ***Reduce something*** – risks, expenses, turnaround times, competition
* ***Improve something*** – website, marketing campaigns, business processes
* ***Save something*** – time, money, space, energy

That’s why JARVIS is a skilled VIRTUAL ASSISTANT.

This Software aims at developing a personal assistant for Linux-based systems. The main purpose of the software is to perform the tasks of the user at certain commands, provided in either of the ways, speech or text. It will ease most of the work of the user as a complete task can be done on a single command. Jarvis draws its inspiration from Virtual assistants like Cortana for Windows and Siri for iOS. Users can interact with the assistant either through voice commands or keyboard input.

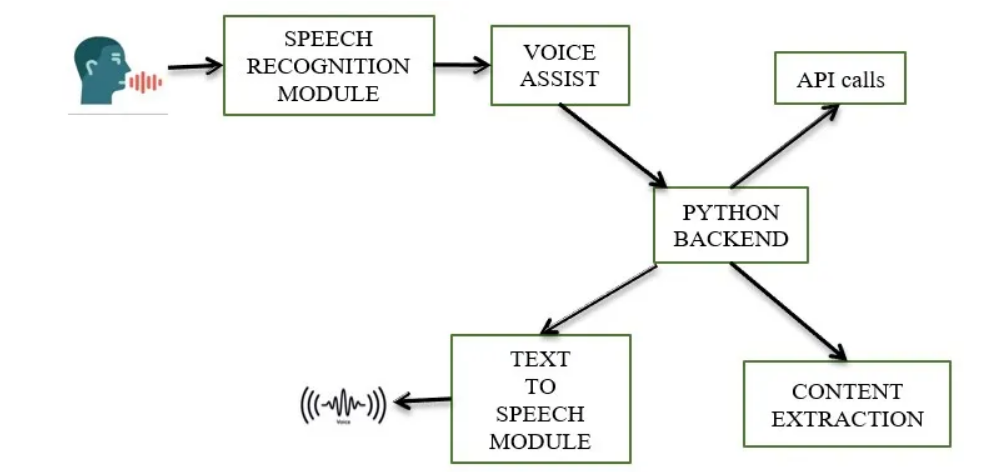
Currently, the project aims to provide the Linux Users with a Virtual Assistant that would not only aid in their daily routine tasks like searching the web, extracting weather data, vocabulary help and many others but also help in automation of various activities. In the long run, we aim to develop a complete server assistant, by automating the entire server management process - deployment, backups, auto-scaling, logging, monitoring and make it smart enough to act as a replacement for a general server administrator

**1.3 Objeective**:Main objective of building personal assistant software JARVIS (a virtual assistant) is using semantic data sources available on the web, user generated content and providing knowledge from knowledge databases. The main purpose of an intelligent virtual assistant is to answer questions that users may have. This may be done in a business environment, for example, on the business website, with a chat interface. On the mobile platform, the intelligent virtual assistant is available as a call-button operated service where a voice asks the user “What can I do for you?” and then responds to verbal input.

Virtual assistants can tremendously save you time. We spend hours in online research and then making the report in our terms of understanding. JARVIS can do that for you. Provide a topic for research and continue with your tasks while JARVIS does the research. Another difficult task is to remember test dates, birthdates or anniversaries. It comes with a surprise when you enter the class and realize it is class test today. Just tell JARVIS in advance about your tests and she reminds you well in advance so you can prepare for the test. One of the main advantages of voice searches is their rapidity. In fact, voice is reputed to be four times faster than a written search: whereas we can write about 40 words per minute, we are capable of speaking around 150 during the same period of time15. In this respect, the ability of personal assistants to accurately recognize spoken words is a prerequisite for them to be adopted by consumers.

The project aims to develop a personal-assistant for desktop. Jarvis draws its inspiration from virtual assistants like Google assistant for android, 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 or using keyboard input. As a personal assistant, Jarvis assists the end-user with day-to-day activities like general human conversation, searching queries in Google, bing or yahoo, searching for videos, sending massages, retrieving images, live weather conditions, word meanings, searching for medicine details, health recommendations based on symptoms and reminding the user about the scheduled events and task.

**1.4 Methedology:**



#### Speech Recognition module

The system uses Google’s online speech recognition system for converting speech input to text. The speech input Users can obtain texts from the special corpora organized on the computer network server at the information centre from the microphone is temporarily stored in the system which is then sent to Google cloud for speech recognition. The equivalent text is then received and fed to the central processor.

#### Python Backend:

The python backend gets the output from the speech recognition module and then identifies whether the command or the speech output is an API Call and Context Extraction. The output is then sent back to the python backend to give the required output to the user.

#### API calls

API stands for Application Programming Interface. An API is a software intermediary that allows two applications to talk to each other. In other words, an API is a messenger that delivers your request to the provider that you’re requesting it from and then delivers the response back to you.

#### Content Extraction

Context extraction (CE) is the task of automatically extracting structured information from unstructured and/or semi-structured machine-readable documents. In most cases, this activity concerns processing human language texts using natural language processing (NLP). Recent activities in multimedia document processing like automatic annotation and content extraction out of images/audio/video could be seen as context extraction TEST RESULTS.

#### Text-to-speech module

Text-to-Speech (TTS) refers to the ability of computers to read text aloud. A TTS Engine converts written text to a phonemic representation, then converts the phonemic representation to waveforms that can be output as sound. TTS engines with different languages, dialects and specialized vocabularies are available through third-party publishers.

**Pyttsx**

Pyttsx stands for Python Text to Speech. It is a cross-platform Python wrapper for textto-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.

**1.5 Role:**

Presently, Jarvis is being developed as an automation tool and virtual assistant. Among the Various roles played by Jarvis are:

1. Search Engine with voice interactions

2. Medical diagnosis with Medicine aid.

3. Reminder and To-Do application.

4. Vocabulary App to show meanings and correct spelling errors.

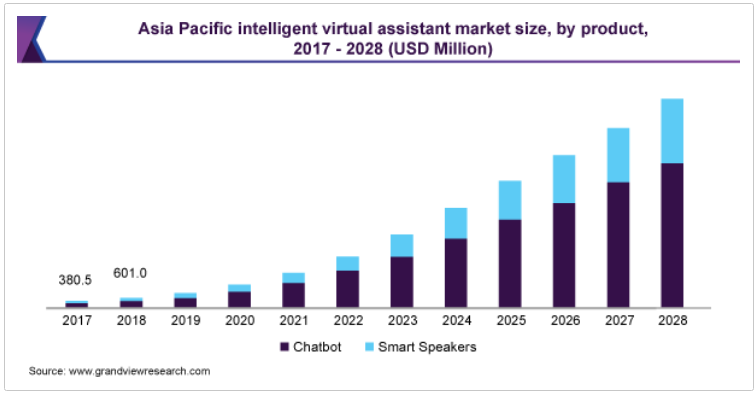
5. Weather Forecasting Application.

Jarvis Voice assistants will continue to offer more individualized experiences as they get better at differentiating between voices. However, it’s not just developers that need to address the complexity of developing for voice as brands also need to understand the capabilities of each device and integration and if it makes sense for their specific brand. They will also need to focus on maintaining a user experience that is consistent within the coming years as complexity becomes more of a concern. This is because the visual interface with voice assistants is missing. Users simply cannot see or touch a voice interface.

The use of Jarvisvirtual assistants can also enhance the system of IoT (Internet of Things). Twenty years from now, Microsoft and its competitors will be offering personal digital assistants that will offer the services of a full-time employee usually reserved for the rich and famous.

**1.6 Contribution Of Project**

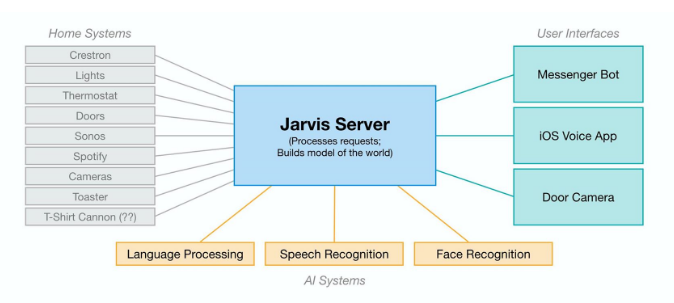
### 1.6.1 Market Potential:

****

The need for improved efficiency across service-based companies and the integration of [Artificial Intelligence](https://www.grandviewresearch.com/industry-analysis/artificial-intelligence-ai-market) (AI) powered virtual assistants among various devices such as tablets, computers, and smartphones, among others, is anticipated to boost the market growth. An Intelligent Virtual Assistant (IVA) is capable of fulfilling a wide range of customer service roles such as ease to navigate, offer product information, assist customers in paying bills, and carry out transfers or complete forms, and direct problem queries to human agents in customer service. Such factors are prompting banking and financial institution across the world to integrate virtual assistants, leading to industry growth.

“Jarvis” was main character of Tony’s Stark’s life assistant in Movies Iron Man. Unlike original comic in which Jarvis was Stark’s human butler, the movie version of Jarvis is an intelligent computer that converses with stark, monitors his household and help to build and program his superhero suit.

**1.6.2 Innovativeness:**



There already exist a number of desktop virtual assistants. A few examples of current virtual assistants available in market are discussed in this section along with the tasks they can provide and their drawbacks.

**SIRI from Apple** SIRI is personal assistant software that interfaces with the user thru voice interface, recognizes commands and acts on them. It learns to adapt to user’s speech and thus improves voice recognition over time. It also tries to converse with the user when it does not identify the user request. It integrates with calendar, contacts and music library applications on the device and also integrates with GPS and camera on the device. It uses location, temporal, social and task based contexts, to personalize the agent behavior specifically to the user at a given point of time.

Supported Tasks

• Call someone from my contacts list

• Launch an application on my Phone

• Send a text message to someone

• Set up a meeting on my calendar for 9am tomorrow

• Set an alarm for 5am tomorrow morning

• Play a specific song in my iTunes library

• Enter a new note Drawback SIRI does not maintain a knowledge database of its own and its understanding comes from the information captured in domain models and data models.

* This Software aims at developing a personal assistant for windows.
* The main purpose of the software is to perform the tasks of the user at certain commands, provided in either of the ways, speech or text.
* It will ease most of the work of the user as a complete task can be done on a single command.
* Jarvis draws its inspiration from Virtual assistants and Siri for iOS.
* Users can interact with the assistant either through voice commands

“Further, most appliances aren’t even connected to the internet yet. It’s possible to control some of these using internet-connected power switches that let you turn the power on and off remotely. But often that isn’t enough.” Some appliances actually required hardware modifications in order for them to work, such as a toaster from the 1950’s which he fitted with a “connected switch” to enable it to automatically start toasting when the power comes on.

1.6.3 Usefullness:

Presently, Jarvis is being developed as an automation tool and virtual assistant. Among the Various roles played by Jarvis are:

1. Search Engine with voice interactions

2. Medical diagnosis with Medicine aid.

3. Reminder and To-Do application.

4. Vocabulary App to show meanings and correct spelling errors.

5. Weather Forecasting Application.

1. Queries from the web: Making queries is an essential part of one’s life, and nothing changes even for a developer working on Linux. We have addressed the essential part of a netizen’s life by enabling our voice assistant to search the web. Here we have used Node JS and Selenium framework for extracting the result from the web as well as displaying it to the user. Jarvis supports a plethora of search engines like Google, Bing and Yahoo and displays the result by scraping the searched queries. In order to make queries from different search engines, the given format should be adopted: Jarvis supports Google, Bing and Yahoo, which should precede the desired query.

2. Accessing youtube videosVideos have remained as a main source of entertainment, one of the most prioritized tasks of virtual assistants. They are equally important for entertainment as well as educational purposes as most teaching and research activities in present times are done through Youtube. This helps in making the learning process more practical and out of the four walls of the classroom. Jarvis implements the feature through a subprocess module which is handled by the main Golang service. This service initiates the subprocess for Node JS which serves the Selenium WebDriver, and scraps the searched YouTube query. In order to access videos from youtube format is: youtube

3. Get weather for a location Getting live weather conditions about a place remains an important task of virtual assistants. It helps the user charter the course of their action. Jarvis addresses this issue with the help of Python. In order to access the live weather condition format is: Weather

4. Retrieve images Users could get images directly through the Jarvis interface. This implementation is done using the Selenium WebDriver. The images are derived as iframes from the entire web code received from Google images. These are formatted according to use and displayed in a compact manner in the Jarvis interface. In order to retrieve image format is: Image

5. Dictionary meaning One of the usages of the web is to find word meaning and its usage in our day to day life. Instead of going through the bulky books, our users can simply search for it using the voice assistant and get the meaning within a fraction of seconds. For retrieving the meaning of a word format is, meaning

6. Medicine Details One of the important issue Jarvis addresses is of healthcare, and medicine in general. The user can query either the medicine or the symptoms. The former lets you know the complete details of the medicine, like indications, contradictions, trade or brand names, dosage, the process of consumption, warning and precautions, storage conditions, etc. On the other hand, the symptom feature lets you query about the symptoms while Jarvis lists various diseases one is likely to be affected along with their medicine. This is helpful for people who are quite busy with their life and find trouble visiting the doctor immediately, thus relying on the web to find the best result for short term cause. Here we use Node JS framework along with Selenium to scrap the required data from the web and display it to the user. We have a huge database of various medicines and symptoms which helps Jarvis respond to the queries of the user at ease. The syntax to be used for querying the necessary are: In order to get details about medicine format is, Medicine In order to re-track the causes of symptoms format is, Symptoms

7. Set Reminders One of the main features of a voice assistant is to set a reminder for the user accordingly. Jarvis is no different when it comes to this. The user can set reminders to be notified about a task at a particular time. This will help users, especially developers to schedule their time and resources easily. All the user have to do is to input Set reminder to the assistant. A form will be displayed. Fill the form with the required details and click on set reminder button.

8. Sending Emails Integrating mailing features to Jarvis eases the job of mailing, which otherwise would have to be done by opening the concerned email address. With Jarvis, you do not need to go for another tab to do one of the major task of your day to day affairs. The user can send emails to the desired receiver. He should input Send mail, after which a form will be displayed. Fill the form with the required details and click on the send mail button.

Why to use Jarvis?

1. It fulfils the lack of a virtual assistant in Linux systems.

2. It has an easy to install and use interface.

3. It accepts inputs even through voice or keyboard.

4. It automates tedious tasks like deployment, unit testing through a single command.

5. It gives live weather updates.

6. It gives advice on health.

1.7 Report organization:

There already exist a number of desktop virtual assistants. A few examples of current virtual assistants available in market are discussed in this section along with the tasks they can provide and their drawbacks.

SIRI from Apple: SIRI is personal assistant software that interfaces with the user thru voice interface, recognizes commands and acts on them. It learns to adapt to user’s speech and thus improves voice recognition over time. It also tries to converse with the user when it does not identify the user request. It integrates with calendar, contacts and music library applications on the device and also integrates with GPS and camera on the device. It uses location, temporal, social and task based contexts, to personalize the agent behavior specifically to the user at a given point of time.

Supported Tasks

• Call someone from my contacts list

• Launch an application on my iPhone

• Send a text message to someone

• Set up a meeting on my calendar for 9am tomorrow

• Set an alarm for 5am tomorrow morning

• Play a specific song in my iTunes library

• Enter a newnote

Drawback

SIRI does not maintain a knowledge database of its own and its understanding comes from the information captured in domain models and data models.

ReQall

ReQall is personal assistant software that runs on smartphones running Apple iOS or Google Android operating system. It helps user to recall notes as well as tasks within a location and time context. It records user inputs and converts them into commands, and monitors current stack of user tasks to proactively suggest actions while considering any changes in the environment. It also presents information based on the context of the user, as well as filter information to the user based on its learned understanding of the priority of that information.

Supported Tasks

• Reminders

• Email

• Calendar, Google Calendar

• Outlook

• Evernote

• Facebook, LinkedIn

• News Feeds

Drawback

Will take some time to put all of the to-do items in – you could spend more time putting the entries in than actually doing the revision.

CHAPTER -2

REQUIREMENT ENGINEERING

As we know Python is a suitable language for scriptwriters and developers. Let’s write a script for Voice Assistant using Python. The query for the assistant can be manipulated as per the user’s need.   
**HARDWARE AND SOFTWARE REQUIREMENTS**

The software is designed to be light-weighted so that it doesn’t be a burden on themachine running it. This system is being build keeping in mind the generally availablehardware and software compatibility. Here are the minimum hardware and softwarerequirement for virtual assistant.

**Hardware:**

* Pentium-pro processor or later.
* RAM 512MB or more.

**Software:**

* Windows 7(32-bit) or above.
* Python 2.7 or later
* Chrome Driver
* Selenium Web Automation
* SQLite

**2.1 USER ROLE AND RESPONSIBILITIES:**

Usually, user needs to manually manage multiple sets of applications to complete one task. For example, a user trying to make a travel plan needs to check for airport codes for nearby airports and then check travel sites for tickets between combinations of airports to reach the destination. There is need of a system that can manage tasks effortlessly.

We already have multiple virtual assistants. But we hardly use it. There are number of people who have issues in voice recognition. These systems can understand English phrases but they fail to recognize in our accent. Our way of pronunciation is way distinct from theirs. Also, they are easy to use on mobile devices than desktop systems. There is need of a virtual assistant that can understand English in Indian accent and work on desktop system.

When a virtual assistant is not able to answer questions accurately, it’s because it lacks the proper context or doesn’t understand the intent of the question. Its ability to answer questions relevantly only happens with rigorous optimization, involving both humans and machine learning. Continuously ensuring solid quality control strategies will also help manage the risk of the virtual assistant learning undesired bad behaviors. They require large amount of information to be fed in order for it to work efficiently.

Virtual assistant should be able to model complex task dependencies and use these models to recommend optimized plans for the user. It needs to be tested for finding optimum paths when a task has multiple sub-tasks and each sub-task can have its own sub-tasks.

In such a case there can be multiple solutions to paths, and the it should be able to consider user preferences, other active tasks, priorities in order to recommend a particular plan.

2.2 REQUIREMENT COLLECTION:

**2.2.1 Functional Requirements:**

● Windows Distribution

● Proper Internet Connection

● Github Credentials

● Python 2.7

● Mplayer for voice support (Text-to-Speech)

● Chromium-based browser, like Chrome, Edge

**2.2.2 Non-Functional Requirements:**

The non-functional requirements of the system include:

● The system ensures safety, security and usability, which are observable during operation (at run time).

● The system is adaptable to different situations.

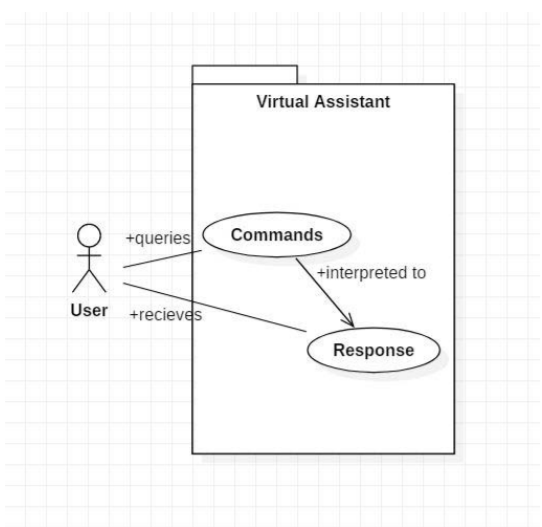
● The project has good and compact UI using AngularJS with responsive interface.

● The project is light on resources.

CHAPTER – 3

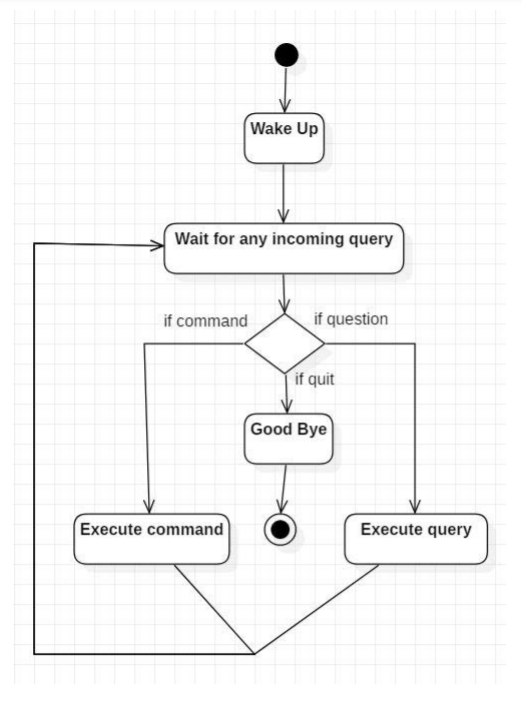
ANALYSIS AND DESIGN

3.1 Use Case Diagram:



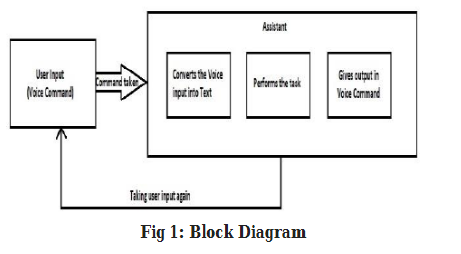
In this project there is only one user. The user queries command to the system. System then interprets it and fetches answer. The response is sent back to the user.

3.2 Activity Diagram:



Initially, the system is in idle mode. As it receives any wake up cal it begins execution. The received command is identified whether it is a questionnaire or a task to be performed. Specific action is taken accordingly. After the Question is being answered or the task is being performed, the system waits for another command. This loop continues unless it receives quit command. At that moment, it goes back to sleep.

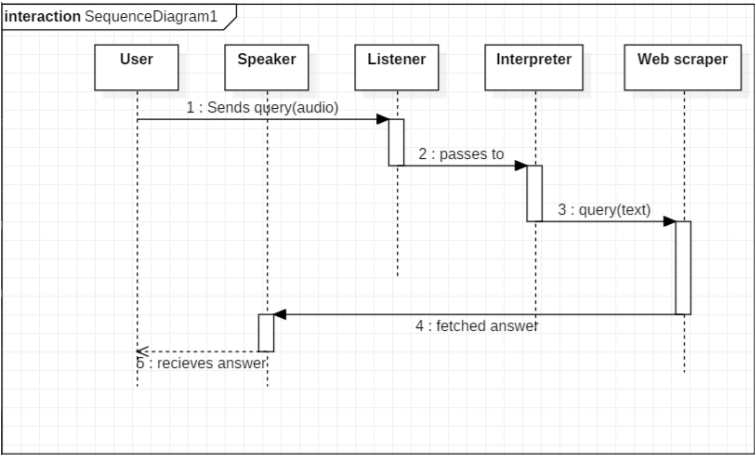
3.3 System Architecture:



This Assistant consists of three modules. First is, assistant accepting voice input from user. Secondly, analysing the input given by the user, and mapping it to the respective intent and function. And the third is, the assistant giving user the result all along with voice.

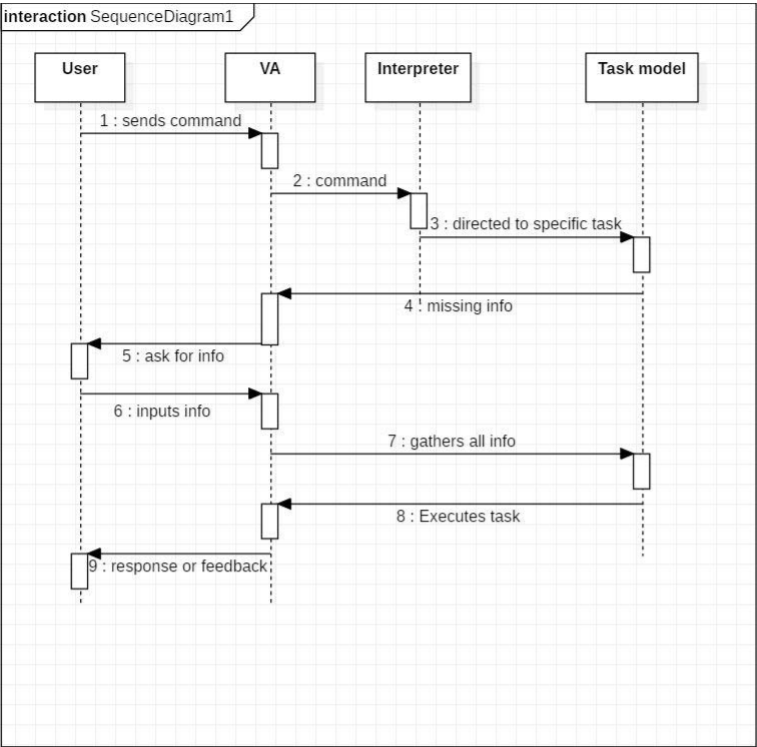
3.4 Sequence Diagram:

Sequence diagram for Query-Response



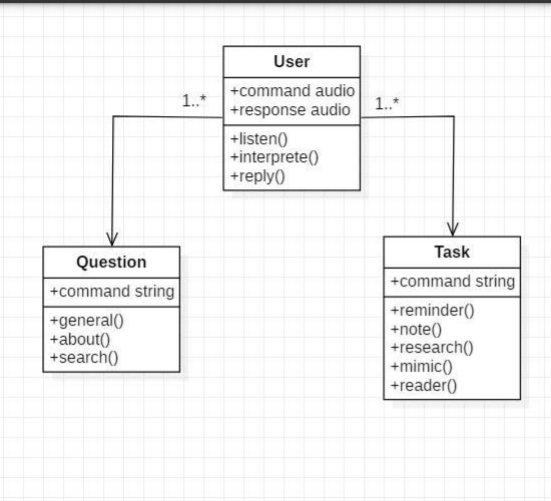
The above sequence diagram shows how an answer asked by the user is being fetched from internet. The audio query is interpreted and sent to Web scraper. The web scraper searches and finds the answer. It is then sent back to speaker, where it speaks the answer to user.

Sequence diagram for Task Execution



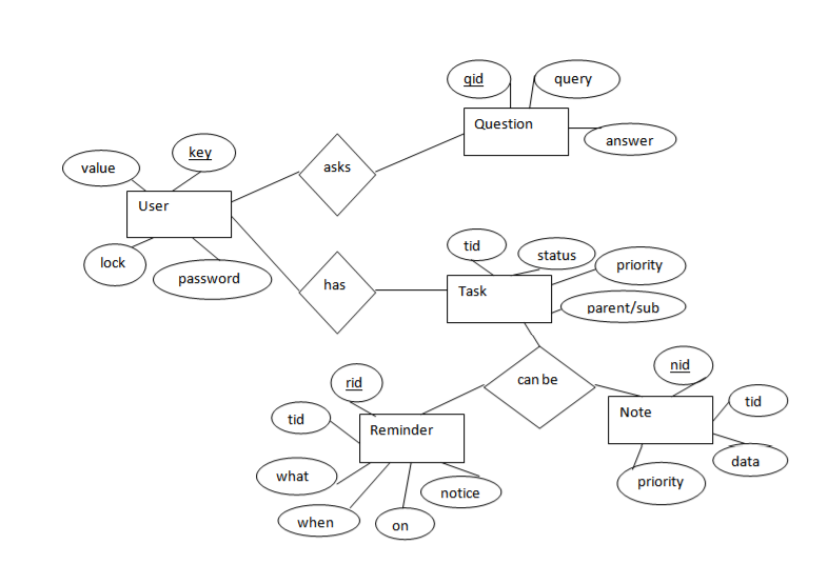
The user sends command to virtual assistant in audio form. The command is passed to the interpreter. It identifies what the user has asked and directs it to task executer. If the task is missing some info, the virtual assistant asks user back about it. The received information is sent back to task and it is accomplished. After execution feedback is sent back to user.

3.5 Class Diagram:



The class user has 2 attributes command that it sends in audio and the response it receives which is also audio. It performs function to listen the user command. Interpret it and then reply or sends back response accordingly. Question class has the command in string form as it is interpreted by interpret class. It sends it to general or about or search function based on its identification. The task class also has interpreted command in string format. It has various functions like reminder, note, mimic, research and reader.

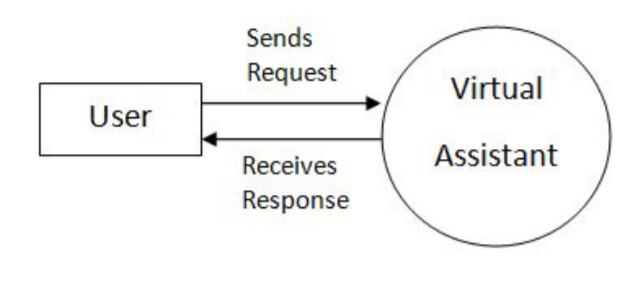
3.6 E-R Diagram:

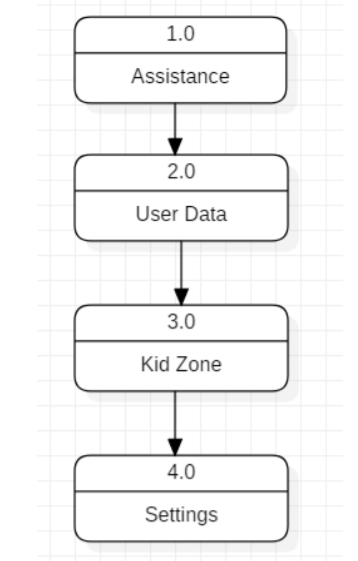


The above diagram shows entities and their relationship for a virtual assistant system. We have a user of a system who can have their keys and values. It can be used to store any information about the user. Say, for key “name” value can be “Jim”. For some keys user might like to keep secure. There he can enable lock and set a password (voice clip). Single user can ask multiple questions. Each question will be given ID to get recognized along with the query and its corresponding answer. User can also be having n number of tasks. These should have their own unique id and status i.e. their current state. A task should also have a priority value and its category whether it is a parent task or child task of an older task.

3.7 Database Design:

DATA FLOW DIAGRAM:





CHAPTER -4

METHEDOLOGY

4.1 Dataset Discussion and Interpretation:

It is essential to evaluate cost and benefit. It is essential to evaluate cost and benefit of the proposed system. Five types of feasibility study are taken into consideration.

1. Technical feasibility: It includes finding out technologies for the project, both hardware and software. For virtual assistant, user must have microphone to convey their message and a speaker to listen when system speaks. These are very cheap now a days and everyone generally possess them. Besides, system needs internet connection. While using JIA, make sure you have a steady internet connection. It is also not an issue in this era where almost every home or office has Wi-Fi.

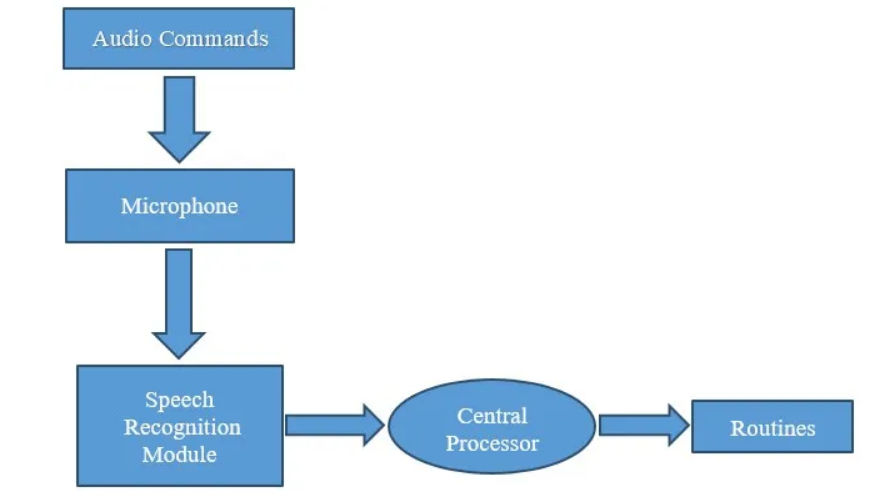
2. Operational feasibility: It is the ease and simplicity of operation of proposed system. System does not require any special skill set for users to operate it. In fact, it is designed to be used by almost everyone. Kids who still don’t know to write can read out problems for system and get answers.

3. Economical feasibility: Here, we find the total cost and benefit of the proposed system over current system. For this project, the main cost is documentation cost. User also would have to pay for microphone and speakers. Again, they are cheap and available. As far as maintenance is concerned, JIA won’t cost too much.

4. Organizational feasibility: This shows the management and organizational structure of the project. This project is not built by a team. The management tasks are all to be carried out by a single person. That won’t create any management issues and will increase the feasibility of the project.

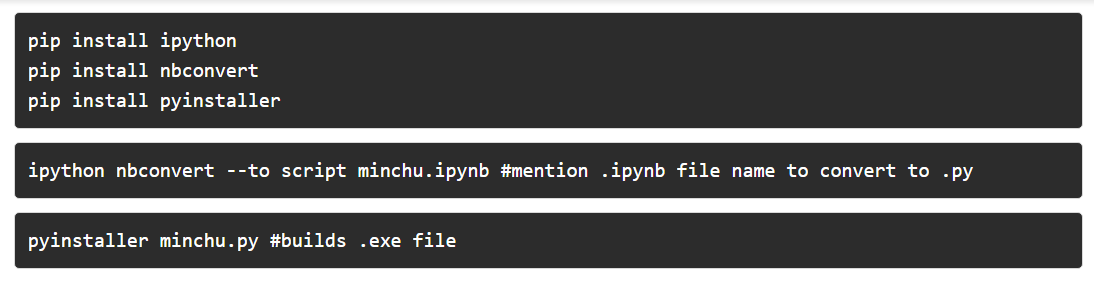
5. Cultural feasibility: It deals with compatibility of the project with cultural environment. Virtual assistant is built in accordance with the general culture. The project is named JIA so as to represent Indian culture without undermining local beliefs. This project is technically feasible with no external hardware requirements. Also it is simple in operation and does not cost training or repairs. Overall feasibility study of the project reveals that the goals of the proposed system are achievable. Decision is taken to proceed with the project.

The work started with analyzing the audio commands given by the user through the microphone. This can be anything like getting any information, operating a computer’s internal files, etc. This is an empirical qualitative study, based on reading above mentioned literature and testing their examples. Tests are made by programming according to books and online resources, with the explicit goal to find best practices and a more advanced understanding of Voice Assistant.

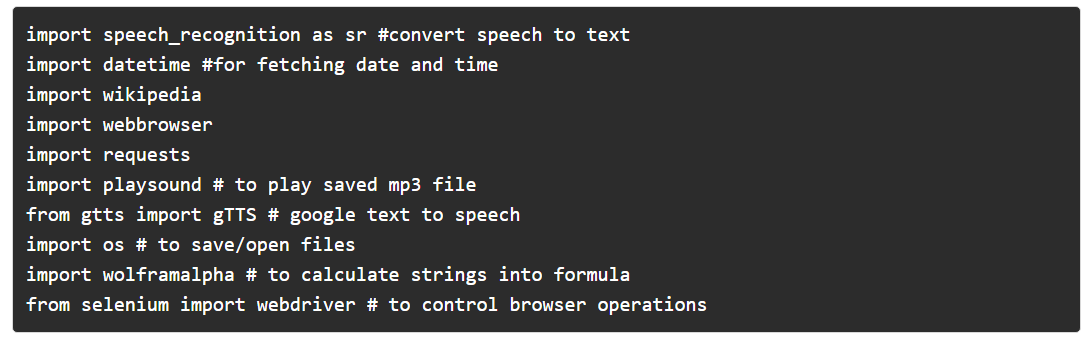


4.2 Proposed Algorithm:

To create an executable from the Python script you can use Pyinstaller. First, you have to convert the .ipynb Python file to a .py extension. For this use python and convert packages. Next, use Pyinstaller to create a .exe file for your .py file. All the following steps need to be performed in the command prompt from the location where Python is installed.



The .py file created should be located in the same folder where the .ipynb file is located. Once the build is complete, Pyinstaller creates two folders, *build and dist*. Navigate to the dist folder and execute the .exe file to run your personal desktop assistant. This application is portable and can be executed on any device.



4.3 Tools Required:

#### Speech Recognition module

The system uses Google’s online speech recognition system for converting speech input to text. The speech input Users can obtain texts from the special corpora organized on the computer network server at the information centre from the microphone is temporarily stored in the system which is then sent to Google cloud for speech recognition. The equivalent text is then received and fed to the central processor.

#### Python Backend:

The python backend gets the output from the speech recognition module and then identifies whether the command or the speech output is an API Call and Context Extraction. The output is then sent back to the python backend to give the required output to the user.

#### API calls

API stands for Application Programming Interface. An API is a software intermediary that allows two applications to talk to each other. In other words, an API is a messenger that delivers your request to the provider that you’re requesting it from and then delivers the response back to you.

#### Content Extraction

Context extraction (CE) is the task of automatically extracting structured information from unstructured and/or semi-structured machine-readable documents. In most cases, this activity concerns processing human language texts using natural language processing (NLP). Recent activities in multimedia document processing like automatic annotation and content extraction out of images/audio/video could be seen as context extraction TEST RESULTS.

#### Text-to-speech module

Text-to-Speech (TTS) refers to the ability of computers to read text aloud. A TTS Engine converts written text to a phonemic representation, then converts the phonemic representation to waveforms that can be output as sound. TTS engines with different languages, dialects and specialized vocabularies are available through third-party publishers.

**Pyttsx**

Pyttsx stands for Python Text to Speech. It is a cross-platform Python wrapper for textto-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.

**Subprocess:-** This module is used for getting system subprocess details which are used in various commands i.e Shutdown, Sleep, etc. This module comes built-in with Python. 

**WolframAlpha:-** It is used to compute expert-level answers using Wolfram’s algorithms, knowledgebase and AI technology. To install this module type the below command in the terminal.

pip install wolframaplha

**Pyttsx3:-** This module is used for the conversion of text to speech in a program it works offline. To install this module type the below command in the terminal.  
**pip install pyttsx3**

**Tkinter:-** This module is used for building GUI and comes inbuilt with Python. This module comes built-in with Python. 

**Wikipedia:-**  we have used the Wikipedia module to get information from Wikipedia or to perform a Wikipedia search. To install this module type the below command in the terminal.

pip install wikipedia

**Speech Recognition:-** Since we’re building an Application of voice assistant, one of the most important things in this is that your assistant recognizes your voice (means what you want to say/ ask). To install this module type the below command in the terminal.

pip install SpeechRecognition

**Web browser:-** To perform Web Search. This module comes built-in with Python. 

**Ecapture:-** To capture images from your Camera. To install this module type the below command in the terminal.

pip install ecapture

**Pyjokes:-** Pyjokes is used for collection Python Jokes over the Internet. To install this module type the below command in the terminal.  
**pip install pyjokes**

**Datetime:-** Date and Time is used to showing Date and Time. This module comes built-int with Python. 

**Twilio:-** Twilio is used for making call and messages. To install this module type the below command in the terminal.

pip install twilio

**Requests:** Requests is used for making GET and POST requests. To install this module type the below command in the terminal.  
**pip install requests**   
 **BeautifulSoup:** Beautiful Soup is a library that makes it easy to scrape information from web pages. To install this module type the below command in the terminal.

pip install beautifulsoup4

**CHAPTER -5**

**CONSTRUCTION**

**5.1 Implementation and Testing:**

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

**5.2 Implemented Classes:**

|  |  |
| --- | --- |
| **Voice Technology** | **Brain Technology** |
| Voice Activation | Voice Bio-metrics |
| Automatic Speech Recognition (ASR) | Dialog Management |
| (Teach-To-Speech (TTS) | Natural Language Understanding (NLU)  Named Entity Recognition NER) |

The class user has 2 attributes command that it sends in audio and the response it receives which is also audio. It performs function to listen the user command. Interpret it and then reply or sends back response accordingly. Question class has the command in string form as it is interpreted by interpret class. It sends it to general or about or search function based on its identification. The task class also has interpreted command in string format. It has various functions like reminder, note, mimic, research and reader.

**5.3 Test Cases:**

Test Case 1

Test Title: Response Time

Test ID: T1

Test Priority: High

Test Objective: To make sure that the system respond back time is efficient.

Description: Time is very critical in a voice based system. As we are not typing inputs, we are speaking them. The system must also reply in a moment. User must get instant response of the query made.

• Test Case 2

Test Title: Accuracy

Test ID: T2

Test Priority: High

Test Objective: To assure that answers retrieved by system are accurate as per gathered data.

Description: A virtual assistant system is mainly used to get precise answers to any question asked. Getting answer in a moment is of no use if the answer is not correct. Accuracy is of utmost importance in a virtual assistant system.

•Test Case 3

Test Title: Approximation

Test ID: t3

Test priority: Moderate

Test Objective: To check approximate answers about calculations.

Description: There are times when mathematical calculation requires approximate value. For example, if someone asks for value of PI the system must respond with approximate value and not the accurate value. Getting exact value in such cases is undesirable.

Note: There might include a few more test cases and these test cases are also subject to change with the final software development.

**CHAPTER -6**

**RESULT AND DISCUSSION**

Conclusion:

This is how simple it is to build your own voice assistant. You can add many more features such as play your favorite songs, give weather details, open email application, compose emails, restart your system, etc. You can integrate this application into your phone or tablet as well. Have fun exploring and developing your own Alexa/Siri/Cortana.

It not only works on human commands but also give responses to the user based on the query being asked or the words spoken by the user such as opening tasks and operations. It is greeting the user the way the user feels more comfortable and feels free to interact with the voice assistant. The application should also eliminate any kind of unnecessary manual work required in the user life of performing every task. The entire system works on the verbal input rather than the next one.

Future Works:

The mass adoption of artificial intelligence in users’ everyday lives is also fueling the shift towards voice. The number of IoT devices such as smart thermostats and speakers are giving voice assistants more utility in a connected user’s life. Smart speakers are the number one way we are seeing voice being used. Many industry experts even predict that nearly every application will integrate voice technology in some way in the next 5 years. The use of virtual assistants can also enhance the system of IoT (Internet of Things). Twenty years from now, Microsoft and its competitors will be offering personal digital assistants that will offer the services of a full-time employee usually reserved for the rich and famous.

APPENDIX-1

PROJECT SYNOPSIS

1. **Problem Domain**

* **Problem Identification**

The project aims to develop a personal-assistant for desktop. Jarvis draws its inspiration from virtual assistants like google assistant for android, 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 or using keyboard input. As a personal assistant, Jarvis assists the end-user with day-to-day activities like general human conversation, searching queries in Google, bing or yahoo, searching for videos, sending massages, retrieving images, live weather conditions, word meanings, searching for medicine details, health recommendations based on symptoms and reminding the user about the scheduled events and task.

* Problem in Existing System

We are all well aware about Siri, Google Assistant and many other virtual assistants which are designed to aid the tasks of users in IOS, Android platforms. But to our surprise, there’s no such virtual assistant available for the paradise of Developers i.e. windows platform.

1. **Solution Domain**

* Suggested Solutions
* We plan to Integrate Jarvis with mobile using react native, to provide a synchronized experience between the two connected devices.
* Further, in the long run, Jarvis is planned to feature auto deployment supporting elastic beanstalk, backup files, and all operations which a general Server Administrator does. The functionality would be seamless enough to replace the Server Administrator with Jarvis.
* **Purpose of the project / innovativeness and usefulness**

This Software aims at developing a personal assistant for windows.

* The main purpose of the software is to perform the tasks of the user at certain commands, provided in either of the ways, speech or text.
* It will ease most of the work of the user as a complete task can be done on a single command.
* Jarvis draws its inspiration from Virtual assistants and Siri for iOS.
* Users can interact with the assistant either through voice commands
* **An improvement over the existing system**

1. We plan to Integrate Jarvis with mobile, to provide a synchronized experience between the two connected devices. Further, in the long run, Jarvis is planned to feature auto deployment supporting elastic beanstalk, backup files, and all operations which a general Server Administrator does. The functionality would be seamless enough to replace the Server Administrator with Jarvis.
2. **System Domain**

* Required Resources

1. Software/Hardware requirement specification:-
2. Front End: python

Back End: mysql

1. Software requirements
2. Visual studio
3. Python IDE
4. Hardware requirements
5. RAM: 512 MB or more
6. Hard disk: 16 GB or more
7. Development requirements
8. Windows 7/10.
9. Python IDE
10. Visual studio

* **Methodology to be adopted / planning of work**

As this program includes the functions and services of: calling services, text message transformation, mail exchange, alarm, event handler, location services, music player service, checking weather, Google searching engine, Wikipedia searching engine.

Pip3 install SpeechRecognition

Pip3 install jsonlib

Pip3 install pyttsx3

Pip3 install PyAudio

Pip3 install geocoder

Pip3 install loc

Pip3 install Wikipeda

**4.Application Domain**

* The Scope of Project:

Presently, Jarvis is being developed as an automation tool and virtual assistant. Among the Various roles played by Jarvis are:

1. Search Engine with voice interactions

2. Medical diagnosis with Medicine aid.

3. Reminder and To-Do application.

4. Vocabulary App to show meanings and correct spelling errors.

5. Weather Forecasting Application. And so on

* **Impact of the work in the real world:**

GUI: I t is used to interact with the user. GUI reflects the basic appearance of the application.

Voice Recognition: For intelligent voice assistant application is done using Google Server. This process involves the conversion of acoustic speech into a set of words and is performed by software component.

Voice Input Manager: It manages the command given by user. It sends the Input given by user to the database manager,

Database Manager: It compares Input given by user that is in the form of voice with the Database which contains vocabulary of words. It sends response to the action performer.

Action Performer: It takes response from the database manager as Input and decides which action should be performed. Action can be in the form of text message or call.

1. Text message: Users are able to send the SMS to a specific person in the phonebook as well by giving a correct command which contains the messaging request keywords. The message should be sent to the destination immediately.

2. Calling service: The application should allow the users to make a call to the person in the contacts or by saying mobile number of the person to whom user wants to call. By giving a correct command with the calling request to a stored person, the Android phone should successfully direct to the number of the person requested.

3.Sending Emails: Integrating mailing features to Jarvis eases the job of mailing, which otherwise would have to be done by opening the concerned email address. With Jarvis, you do not need to go for another tab to do one of the major tasks of your day to day life. The user can send emails to the desired receiver. He should input send mail, after which a form will be displayed. Fill the form with the required details and click on the send mail button.

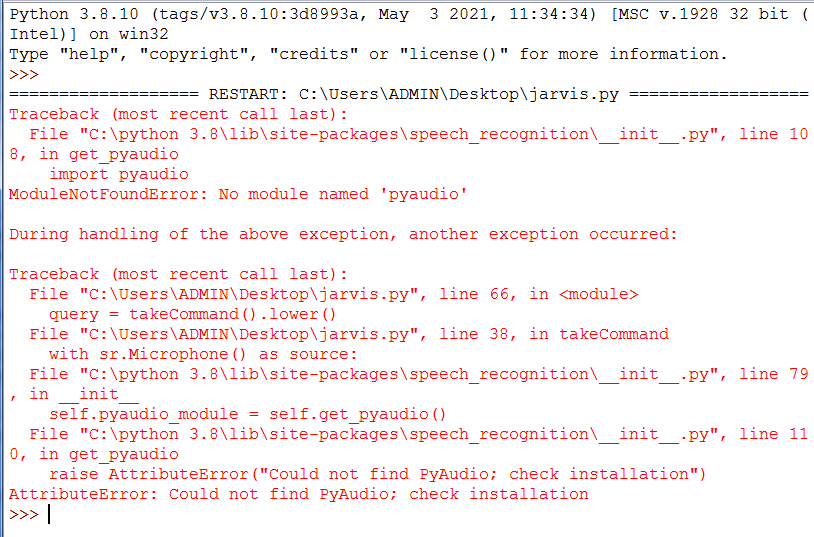
**5. Expected Outcomes/ Benefits:**

Through this voice assistant, we have automated various services using a single line command. It eases most of the tasks of the user like searching the web, retrieving weather forecast details, vocabulary help and medical related queries. We aim to make this project a complete server assistant and make it smart enough to act as a replacement for a general server administration. The future plans include integrating Jarvis with mobile using React Native to provide a synchronized experience between the two connected devices. ​Further, in the long run, Jarvis is planned to feature auto deployment supporting elastic beanstalk, backup files, and all operations which a general Server Administrator does. The functionality would be seamless enough to replace the Server Administrator with Jarvis.

**APPENDIX-2**

**GUIDE INTERACTION REPORT**

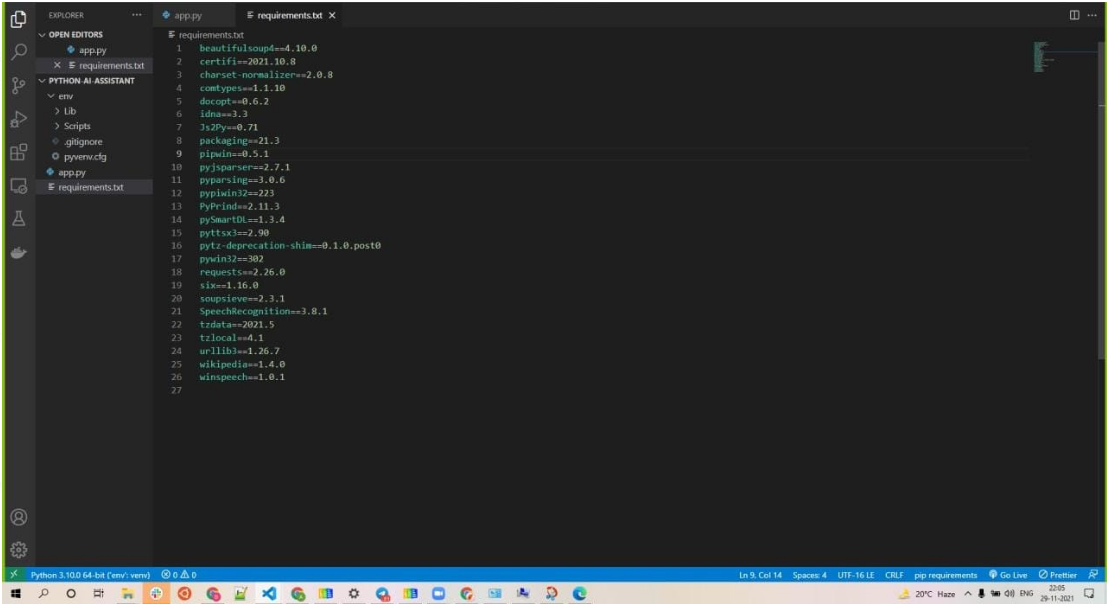
We have completed our project under the guidance of Prof.Mitushi Soni. She guided us in downloading the python modules. She cleared all the errors occurred during the project.

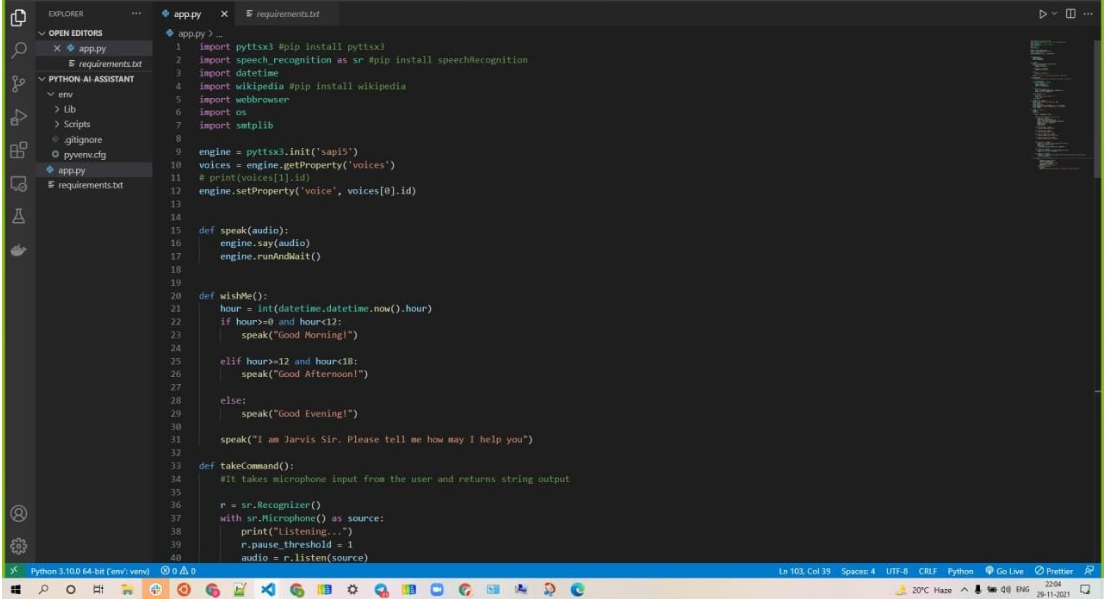
****

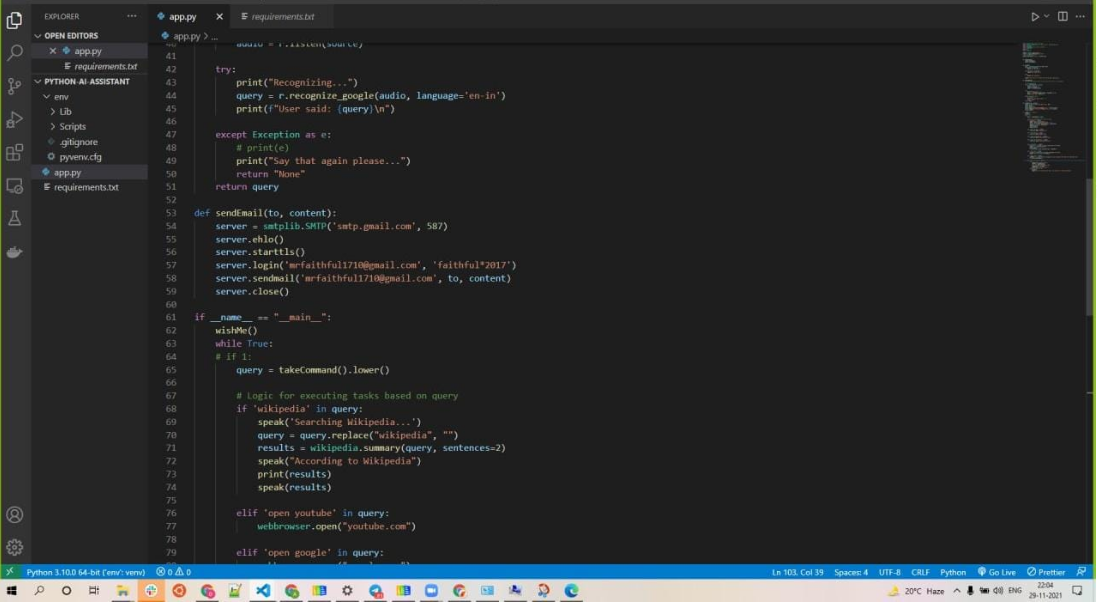
This was the major error occurred in our project and our guide helped us in clearing the error.

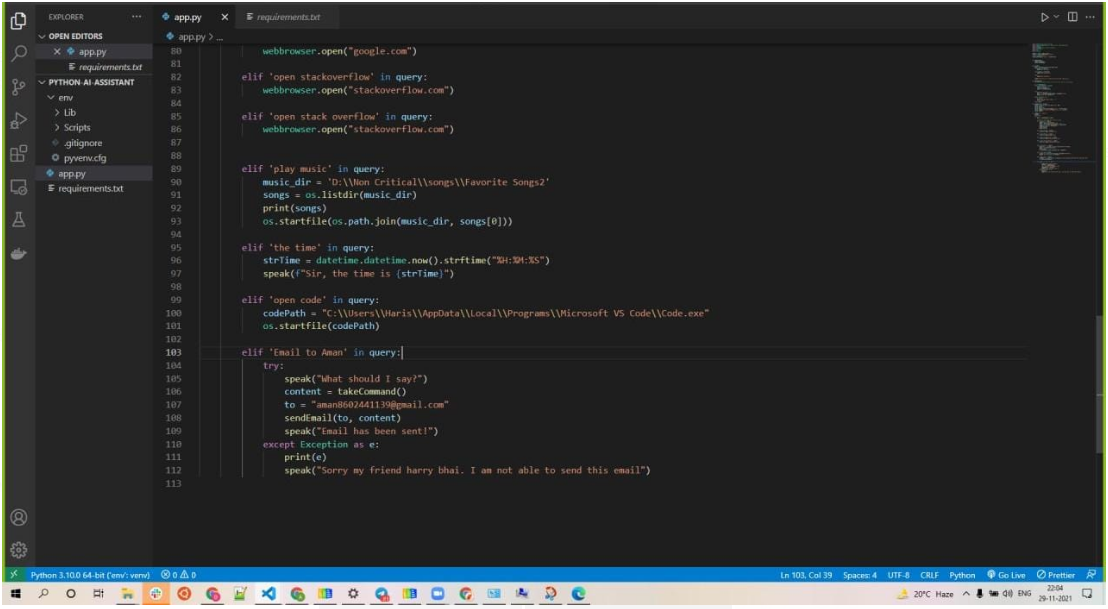
**APPENDIX:3**

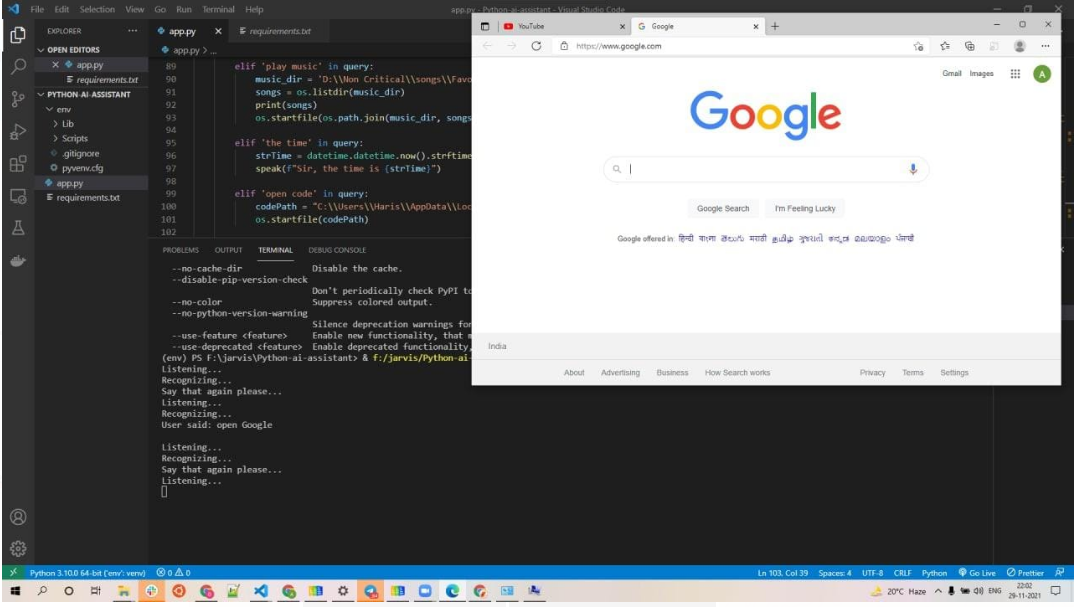
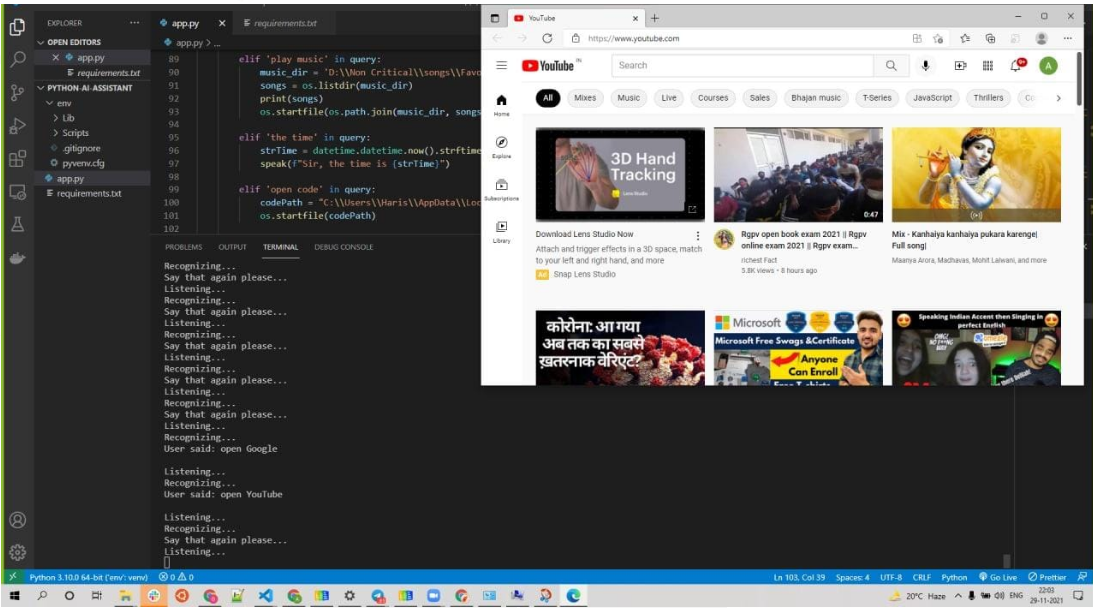
**PROJECT SNAPSHOTS**

****

****







REFERENCE

Websites referred

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

www.pythonprogramming.net

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

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

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

• Books referred

Python Programming - Kiran Gurbani

Learning Python - Mark Lutz

• YouTube Channels referred

CS Dojo

edureka!

• Documents referred

Designing Personal Assistant Software for Task Management using Semantic Web Technologies and Knowledge Databases - Purushotham Botla

Python code for Artificial Intelligence: Foundations of Computational Agents - David L. Poole and Alan K. Mackworth