



A Project Report for Fulfillment of Course –
Project Based Learning
First Year Engineering Program of SPPU, Semester II
(2020-2021)

Submitted by:

DIKSHA PRAJAPATI	(FE20Comp072)
JATIN OZA	(FE20Comp077)
VAIBHAV UPGANLAWAR	(FE20Comp083)
SONAL GUPTA	(FE20Comp092)
ASHLESHA MEMANE	(FE20Comp075)

Under the Guidance of
MRS. VRUSHALI UTTARWAR

CERTIFICATE

This is to certify that the Project Report on “JARVIS: The Personal Voice Assistant” submitted by Diksha Prajapati, Jatin Oza, Vaibhav Urganlawar, Ashlesha Memane and Sonal Gupta of First year Engineering Program for **PROJECT BASED LEARNING** has been successfully completed.

MRS. VRUSHALI UTTARWAR

Project Guide

Prof. HNNIE WILLIAM

Academic Coordinator

DR. SANJAY K. BABAR

Dean-FE Section

DR. VIJAY M. WADHAI

Principal

SYNOPSIS

ON

JARVIS: THE PERSONAL VOICE ASSISTANT



INTRODUCTION

What is a Voice Assistant?

A voice assistant is a digital assistant that uses voice recognition, language processing algorithms, and voice synthesis to listen to specific voice commands and return relevant information or perform specific functions as requested by the user. The long-term vision for voice assistants is to act as a smart bridge between humans and the vast knowledge and capacities which the internet delivers. Taking away the need to use any device or screen to interact with the internet, technology or other humans in different locations, soon we'll be able to do it all with our voices only.

What is Jarvis?

Jarvis is the virtual assistant like Cortana for Windows, Siri for ios and Google developed "Google Voice Search" generally called Google Assistant which is use for android MOBILES. When we start commanding to our Jarvis, It will first wish you according to the time. As a personal assistant our Jarvis can assists end-users with day to day activities like searching queries in Google, watching videos on

YouTube, sending mails, searching programming related queries on the stack flow, playing color game etc.. The user's statements/commands are analyzed with the help of machine learning. User can be interactive with Jarvis by Voice chat. Jarvis help in saving time.

METHODOLOGY

We used python to make our personal A.I. assistant.

First of all, we created the "wish" function which greets according to our A.I. system time. After wish function we created the "takecommand" function which will helped our A.I. to take command from the user and it is also responsible for returning user's query in string format. By using various modules we developed code logic for doing tasks like sending an email, playing music, Wikipedia search, opening any websites, opening code editor or IDE and various other tasks using voice command. Some of the modules used are pyttsx3, speech Recognition, Wikipedia, web browser, date time, os, smtplib and pyjokes.

We also added some other smaller tasks and games to make the assistant more interactive and attractive.

FEATURES OF JARVIS

- 1) Wishes and greets the user according to the time like Good Morning, Good Afternoon or Good Evening.
- 2) The user can enjoy a number of platforms by just speaking few words. Jarvis can: Open YouTube, Open Google , Play Music, Tell the time, Search in Wikipedia, Open code of the voice bot program, Send Email Open color game etc.

OUTCOMES OF VOICE ASSISTANT

In today's world the impact of Voice assistant is quite huge and will keep on increasing. Through Voice assistant, one can do anything from playing music to stop your alarm. You just have to give command and Yes! It will follow your orders. You ask for what is the weather? It will give you the weather information in details and help you with many more tasks!

FUTURE OF VOICE-BOTS:

Some of the study forecasts that nearly 92.3 percent smartphone users will be using voice assistants by 2023 and by the end of 2030 there is estimation of more than 125 billion voice controlled devices. Huge market brands such as Amazon, Google are continuing to fuel this growing trend to set up a point/statement in this Developing Era.

The ambitions of some companies show what voice assistants could successfully do in the future. As an example, Google has developed Google Duplex, which is a technology that can conduct natural conversations and carry out "real world" tasks over the phone.

Some more field in which future of Voice Assistant is bright:

- 1) Mobile App Integration
- 2) In Healthcare sector
- 3) Voice Cloning
- 4) Smart Devices
- 5) In Gaming Industry

CONCLUSION

Jarvis is a voice assistant. Through Jarvis, we have automated various tasks. It performs tasks like sending mail, searching Google, opening YouTube, Wikipedia, sending mails etc. We aim to make this project a complete server assistant and make it smart to act as a replacement for a general server administration



REFERENCES

- [1] https://www.ripublication.com/irph/ijert_spl17/ijertv10n1spl_80.pdf
- [2] <http://www.diva-portal.org/smash/get/diva2:531796/FULLTEXT02.pdf>
- [3] <https://clearbridgemobile.com/7-key-predictions-for-the-future-of-voice-assistants-and-ai/>
- [4] <https://www.datadab.com/blog/voice-assistants/>
- [5] <https://www.smartsheet.com/voice-assistants-artificial-intelligence>

ACKNOWLEDGEMENT

Our first experience of project has been successful, thanks to the support staff of many friends & colleagues with gratitude. We wish to acknowledge all of them. However, we wish to make special mention of the following.

First of all, we are thankful of our project guide Mrs. VRUSHALI UTTARWAR ma'am under whose guideline we were able to complete our project. We are wholeheartedly thankful to her for giving us her valuable time and attention and for providing us a systematic way for completing our project in time.

We are also very thankful to respective principal DR. VIJAY M. WADHAI sir who gave us an opportunity to present this project.

INDEX

CHAPTER NUMBER	TITLE	PAGE NO
1	INTRODUCTION	10
2	LITERARTURE REVIEW	11-13
3	DESIGN AND DEVELOPMENT	14-27
4	CONCLUSION	28
5	REFERENCES	29



JARVIS: THE PERSONAL VOICE ASSISTANT

CHAPTER 1

INTRODUCTION

WHAT IS A VOICE ASSISTANT?

The scenario out of science fiction movies, where we'd come home and just start speaking to our personal home computer or digital assistant which would take care of our every need, is not too far from being realised anymore.

Enjoying the luxury of always having an assistant at our side, who can support us in our daily tasks 24/7, is making its way into our lives at a rapid pace of adoption. Granted their functionalities are still limited, but with time voice assistants will be able to help us with more and more activities.

The beginnings of the work done in this area reaches back to the 1970ies. When voice recognition of computers was not much more than a grand vision. It took until Apple's voice assistant Siri entered the stage to gain a significant leap forward because it was the first personal voice assistant that was broadly available to the public as part of the iPhone.

In general voice assistants react to voice commands and give the user relevant information about his inquiry. Presently voice assistants are already able to process orders of products, answer questions, perform actions like playing music or start a simple phone call with a friend. The basics of the technology currently exist and the next few years will be used to develop these artificially intelligent assistants even further, enabling them to have more complex capabilities.

The long-term vision for voice assistants is to act as a smart bridge between humans and the vast knowledge and capacities which the internet delivers. Taking away the need to use any device or screen to interact with the internet, technology or other humans in different locations. Soon we'll be able to do it all with our voices only.

• FEATURES OF PYTHON



- Python is a very high-level programming language, yet it is effortless to learn. Anyone can learn to code in Python in just a few hours or a few days. Mastering Python and all its advanced concepts, packages and modules might take some more time. However, learning the basic Python syntax is very easy, as compared to other popular languages like C, C++, and Java. Python code looks like simple English words. There is no use of semicolons or brackets, and the indentations define the code block. You can tell what the code is supposed to do simply by looking at it.
- Python is developed under an OSI-approved opensource license. Hence, it is completely free to use, even for commercial purposes. It doesn't cost anything to download Python or to include it in your application. It can also be freely modified and re-distributed. Python can be downloaded from the official **Python website**.
- Python has an extensive standard library available for anyone to use. This means that programmers don't have to write their code for every single thing unlike other programming languages. There are libraries for image manipulation, databases, unit-testing, expressions and a lot of other functionalities. In addition to the standard library, there is also a growing collection of thousands of components, which are all available in the **Python Package Index**.
- When a programming language is interpreted, it means that the source code is executed line by line, and not all at once. Programming languages such as C++ or Java are not interpreted, and hence need to be compiled first to run them. There is no need to compile Python because it is processed at runtime by the interpreter.
- Python is portable in the sense that the same code can be used on different machines. Suppose you write a Python code on a Mac. If you want to run it on Windows or Linux later, you don't have to make any changes to it. As such, there is no need to write a program multiple times for several platforms.
- A programming language is object-oriented if it focuses design around data and objects, rather than functions and logic. On the contrary, a programming

language is procedure-oriented if it focuses more on functions (code that can be reused). One of the critical Python features is that it supports both object-oriented and procedure-oriented programming.

- Many programming languages need to declare the type of the variable before runtime. With Python, the type of the variable can be decided during runtime. This makes Python a dynamically typed language
- Python is a high-level programming language because programmers don't need to remember the system architecture, nor do they have to manage the memory. This makes it super programmer-friendly and is one of the key features of Python.
- One of the key aspects of any programming language is support for GUI or Graphical User Interface. A user can easily interact with the software using a GUI. Python offers various toolkits, such as Tkinter, wxPython and JPython, which allows for GUI's easy and fast development.
- Python contains several advanced programming features such as generators and list comprehensions. Python also has automatic memory management eliminating the need to manually allocate and free memory in the code.

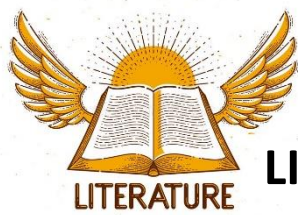
What is JARVIS?

“Jarvis” was the main character of Tony’s Stark’s life assistant in the Movies of Iron Man. The movie version of Jarvis was an intelligent computer that conversed with Stark, monitored his household and helped to build and program his superhero suits.

In our project we mainly use voice as communication means so Jarvis is basically the Speech recognition application. The basic concept of voice assistant encompasses two technologies:

- 1) Synthesizer
- 2) Recognizer

A synthesizer takes input and produces an audio stream as output, while speech recognizer takes an audio stream as input and outputs it as text. Speech recognition is a topic that is very useful in many applications and environments in our daily life. Generally speech recognizer is a machine which understands humans and their spoken word in some way and can act thereafter. One may be very familiar with the voice assistants present in the market like Google Assistant, Siri by Apple, Alexa by Amazon, Cortana by Microsoft and many others. A different aspect of speech recognition can be to facilitate for people with functional disability or other kinds of handicap. To make their daily chores easier, voice control could be helpful. In our project it can perform some basic tasks like browsing various sites, weather information, and play music for you, etc.



CHAPTER 2

LITERATURE REVIEW



Consumers buying behavior can be influenced by voice assistant for various service sectors as adoption of voice technology is both opportunity and challenge. Firms can use voice assistant as a new communication channel and touch point which will also uplift the brand image of these firms as an innovative service provider.

The term Voice Assistant means conversational agent who does tasks, whether functional or social, with or for a person and who is capable to improve his/her understanding of the speaker and the background. Incorporated in smart objects, this app utilizes a combination of AI techniques, including automated speech recognition (ASR), text-to-speech synthesis (TTS), and natural language understanding (NLU) to communicate with normal interactions with people by conversation.

With more industries taking over voice assistants, Financial Service companies will need to begin integrating these voice-based technologies into customer acquisition and retention strategies, marketing, sales and customer experience. Financial Service companies can use their internal voice assistants to increase their employees' and sales executives' productivity and effectiveness.

Although speakers can help organize and manage daily routines of customers, they can act as a channel and offer voice-based services which go beyond personalized support for the provision of banking services. Customers have long been using financial apps for personal banking because they can carry out banking tasks without personal interactions on mobile devices. Customers are provided with the convenience and luxury of Voice-enabled digital banking which allows them to carry out banking transactions only via voice.

Early vision of voice support services from Bank of America led to investments in Erica, aimed at helping customers manage financial lives using predictive analytics and cognitive message. Erica took customer relationships into the next level of

personalization, modulated on the personal banker. (Tearsheet,2017) Erica has all the information about customers

In the Education sector also voice assistant is contributing in educating kids and adults. Problem Pal is proposed in (Trivedi,2018), an Alexa Skill that enables teachers to produce content from voice commands automatically. The capacity of Wikipedia, Wolfram Alpha and the Khan Academy can generate practical questions about any subject. Speakers are exceptionally good at mathematics and spelling words, so they can be used to monitor and check their performance in mathematics by primary school students. In a first and second grade sample of primary students (Selak. 2017), students have regularly verified their mathematical results and have not requested their teacher's assistance

Alexa may be useful for an appointment in health care Plan, patient information, pharmacy, and pharmacy information facts. Anybody can use Alexa to track the Doctor's availability and we can prepare our Offering of rooms, appointment. Alexa may be used to take the patient's concentration Information of patients such as medications, and diet.

Boston Children's has checked Alexa's abilities to support hospital employees in regular activities. One of the Intensive Care Unit (ICU) skills gives nurses detailed information on medical dosage, unique procedures, contact information for staff and other related data points. It also helps users to decide where a medical device or prescription is stored in a large storage room to save their time.

CHAPTER 3

Design and Development/ Methodology Used to collect Data and Its analysis

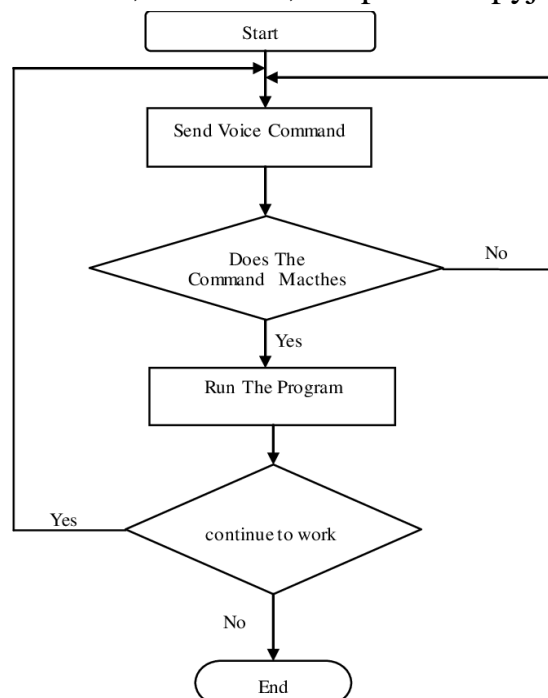
We used Python to make our "JARVIS" as a personal A.I assistant. Python is widely used for AI and ML. First of all, we created the "wish" function which greets the users according to the A.I system time.

After the wish function, we created the "takecommand" which helps Jarvis to take command from the users.

By using various modules we developed code logic for doing various tasks like sending emails, playing music, weather forecast, opening Google, games, tell the time, opening any websites and various other tasks using the voice command.

Voice recognition works based on the premise that a person voice are unique to different speakers. Users were provided multiple questions which they are supposed to ask Jarvis.

Some of the modules used are pytsx3,speech recognition, Wikipedia, Web browser, date time, smtplib and pyjokes.



ALL THE TASKS THAT OUR JARVIS CAN PERFORM

First of all, Jarvis greets us according to the current time, like good morning, good afternoon and good evening. Then we can ask it to perform a number of tasks like:

- Open YouTube
- Open Google
- Open Amazon
- Open Netflix
- Play Music
- Tell the time
- Open code
- Email me
- Can tell us jokes
- Tell us the weather of the city we want
- Search the specific topic we mention in Google
- Play songs we mention on YouTube
- Search the product we say on Amazon
- Search any topic directly on YouTube as well
- Open games like:
 - snake game
 - color game
 - pong game
 - tic tac toe
- Can Translate any sentence from English to Spanish and vice versa

- And also, you can terminate the program by just saying “exit” or “quit”!!
- Can also search any topic in wikipedia

In order to make our project we used different modules. Here are the modules used and its need for use is defined as:

```
import pyttsx3
import datetime
import speech_recognition as sr
import wikipedia
import webbrowser
import os # It is used to open files in the computer
import smtplib # Use to send email
import pyjokes
# import colour # another file for game in same directory
import requests # Used for weather forecasting
from googletrans import Translator
from wikipedia import exceptions # module for starting file
import pywhatkit as kit
```

1. PYTQ5 for live GUI

- PytQ5 is one of the most used modules in building GUI in python.
- It is the most important python bindings.

PYTHON LIBRARIES:

- pyttsx3: it is a python module which convert text to speech.
- Speech Recognition: it is a python module which converts speech to text.
- Datetime: it is a python module which gives actual date and time.
- Wikipedia: it is a python module for searching anything on Wikipedia.
- Smtplib: it is a simple mail transfer protocol that allows us to send mails.
- Pyjokes: it is python libraries which contains lots of interesting and funny jokes.

- Webbrowser: it provides for accessing the world wide web.
- os: it represents the operating system We used the pytsx3 module for speech recognition and defined a speak function that we will use for Jarvis to reply in voice command.

```
# sapi5 used to take voices which is inbuilt in windows
engine = pyttsx3.init('sapi5')
voices = engine.getProperty('voices')
# print(voices)
# Male voice and if voices[1] then female
engine.setProperty('voice', voices[1].id)

def speak(audio): # for it to speak the function is created
    engine.say(audio) # saying the audio that is passed
    engine.runAndWait()
```

This is the wishMe() function, that will wish the user according to the current time!

```
def wishMe(): # will wish me in time
    hour = int(datetime.datetime.now().hour) # will take current time is pc
    if hour >= 0 and hour <= 12:
        speak("Good Morning")

    elif hour > 12 and hour <= 18:
        speak("Good Afternoon")

    else:
        speak("Good Evening")

    speak("I am Jarvis . Please tell me how may I help you")
```

Then we created the takeCommand() function that takes voice input from the user and returns string output.

```

def takeCommand():
    # It takes voice input from user and return string output
    r = sr.Recognizer()
    with sr.Microphone() as source:
        print("Listening...")
        r.pause_threshold = 1 # can change different parameters also
        # seconds of non-speaking audio before a phrase is considered complete
        audio = r.listen(source)

    try: # will run if there is no error
        print("Recognizing....")
        query = r.recognize_google(audio, language="en-in")
        # its using google engine to recognize audio there are different also present like bing,etc
        print(f"User said: {query}\n")

    except Exception as e: # will run if program encounters any error
        print(e)
        # If you don't want to see error in console comment above line
        print("Say that again please...")
        return "None" # Returning none string and not python one
    return query

```

Screenshots of some of tasks that JARVIS can perform with the help of different modules :

```

if "wikipedia" in query:
    speak("Searching Wikipedia")
    query = query.replace("wikipedia", "")
    result = wikipedia.summary(query, sentences=2)
    # returns 2 sentences from wikipedia
    speak("According to wikipedia")
    print(result)
    speak(result)

elif "open youtube" in query:
    webbrowser.open("youtube.com")

elif "open google" in query:
    webbrowser.open("google.com")

elif "open netflix" in query:
    webbrowser.open("https://www.netflix.com/in/")

elif "play music" in query:
    music_dir = r"D:\Songs"
    songs = os.listdir(music_dir)
    print(songs)
    os.startfile(os.path.join(music_dir, songs[0]))

```

```

elif "the time" in query:
    strTime = datetime.datetime.now().strftime("%H:%M:%S")
    speak(f"Madam, the time is {strTime}")

elif "open code" in query:
    code_path = r"C:\Users\Diksha\PycharmProjects\PBL Project\Jarvis2.py"
    os.startfile(code_path)
    speak("The code for this program has been opened ")

elif "email me" in query:
    try:
        speak("What should I say")
        content = takeCommand()
        to = "dikshaprajapati42@gmail.com"
        sendEmail(to, content)
        speak("Email has been sent")
    except exceptions as e:
        print(e)
        speak("Sorry Diksha, I am not able to send this email")

elif "colour game" in query:
    speak("Opening Game")
    os.system("Game.py")
    exec(open("Game.py").read())

```

```

elif "jokes" in query:
    jokes()

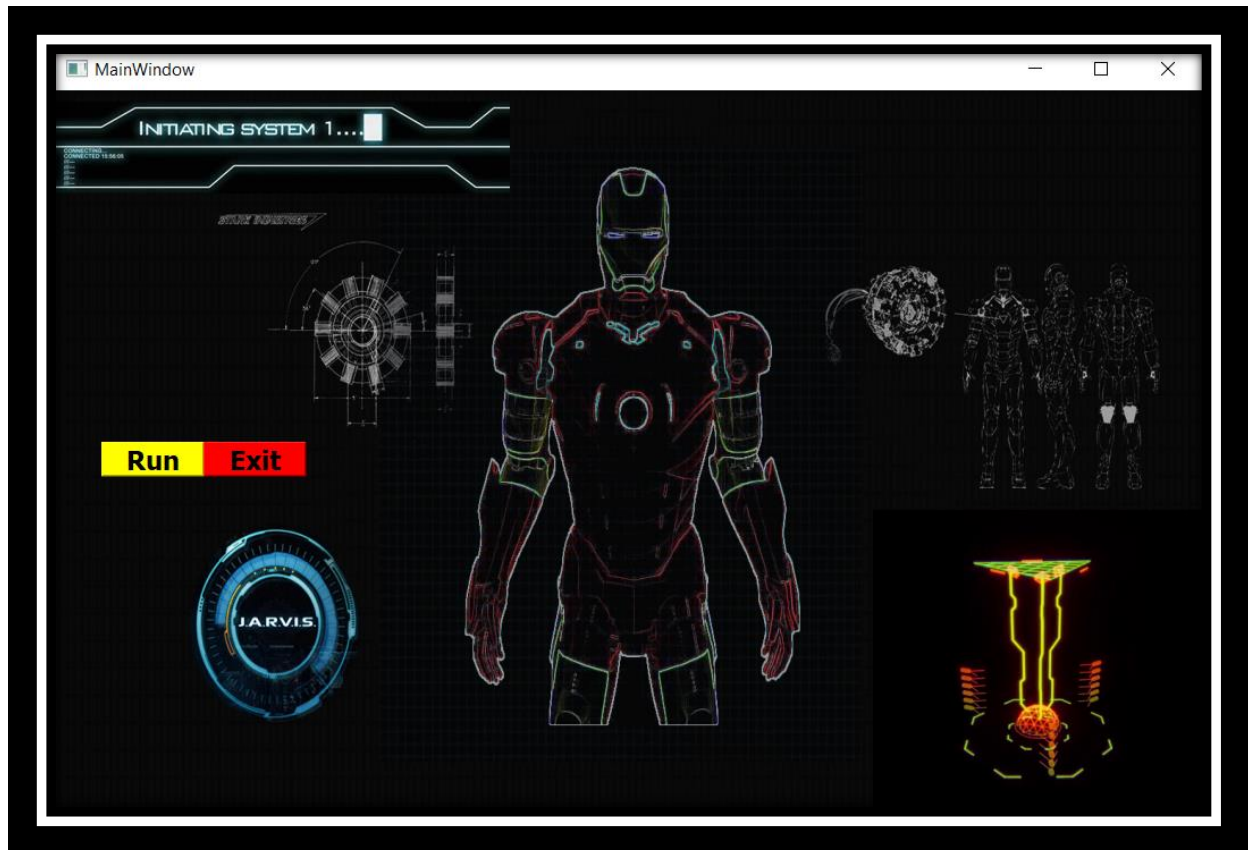
elif "weather" in query:
    weather()

elif "translator" in query:
    translator()

elif "search google" in query:
    speak("what should i search on google")
    cm = takeCommand().lower()
    webbrowser.open(
        "https://www.google.com/search?q=" + cm)

```

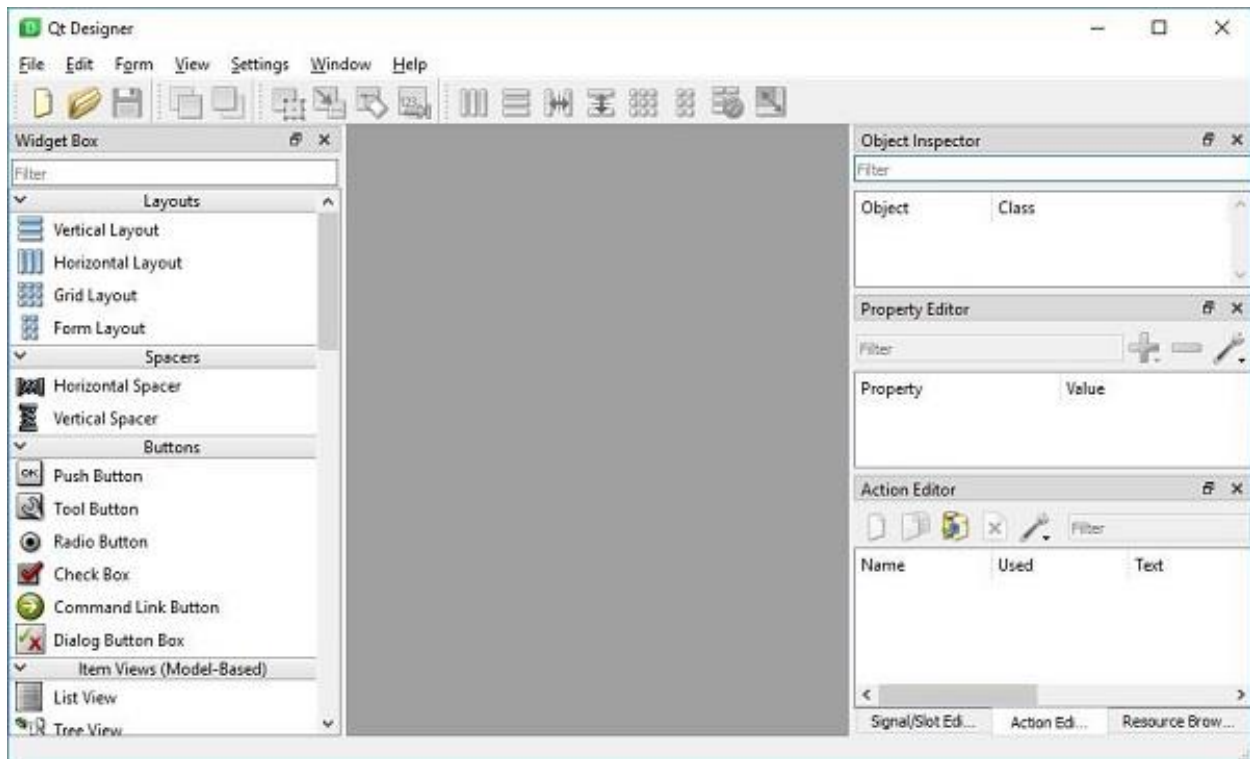
CREATING THE GUI FOR JARVIS



The GUI for our JARVIS Project is made with the help of module pyqt5.

The PyQt installer comes with a GUI builder tool called **Qt Designer**. Using its simple drag and drop interface, a GUI interface can be quickly built without having to write the code. It is however, not an IDE such as Visual Studio. Hence, Qt Designer does not have the facility to debug and build the application.

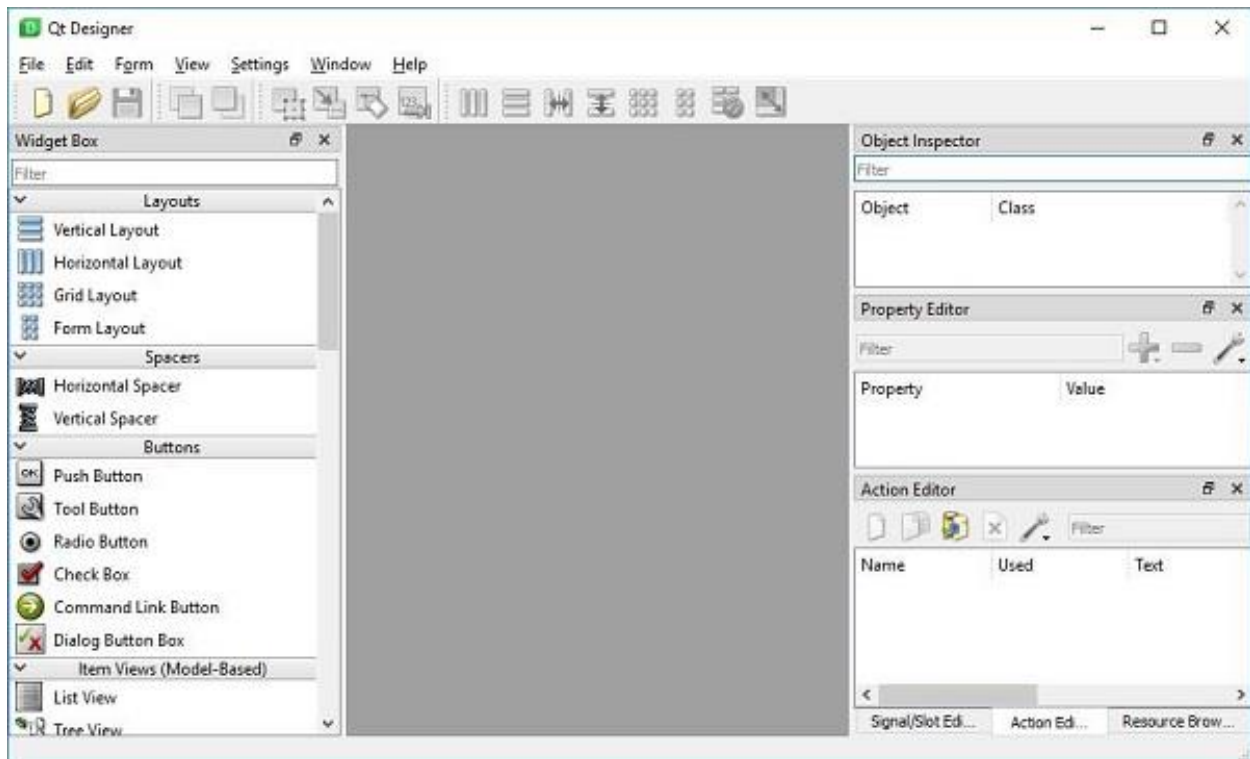
Start Qt Designer application which is a part of development tools and installed in scripts folder of the virtual environment.



You can then drag and drop required widgets from the widget box on the left panel. You can also assign value to properties of widget laid on the form. The designed form is saved as demo.ui. This ui file contains XML representation of widgets and their properties in the design. This design is translated into Python equivalent by using pyuic5 command line utility. This utility is a wrapper for uic module of Qt toolkit. The usage of pyuic5 is as follows –

pyuic5 -x demo.ui -o demo.py

In the above command, -x switch adds a small amount of additional code to the generated Python script (from XML) so that it becomes a self-executable standalone application.



```
jarvis.py > ...
224
225 startExecution = MainThread()
226
227 class Main(QMainWindow): #QMainWindow has been inherited
228     def __init__(self):
229         super().__init__()
230         self.ui = Ui_JarvisUI()
231         self.ui.setupUi(self)
232         self.ui.pushButton.clicked.connect(self.startTask)
233         self.ui.pushButton_2.clicked.connect(self.close) #.close already defined
234
235     def startTask(self):
236         self.ui.movie = QtGui.QMovie("C:/Users/Lenovo/Downloads/7LP8.gif")
237         self.ui.label.setMovie(self.ui.movie)
238         self.ui.movie.start()
239
240         self.ui.movie = QtGui.QMovie("C:/Users/Lenovo/Downloads/T8bahf.gif")
241         self.ui.label_2.setMovie(self.ui.movie)
242         self.ui.movie.start()
243
244         self.ui.movie = QtGui.QMovie("gif/giphy.gif")
245         self.ui.label_3.setMovie(self.ui.movie)
246         self.ui.movie.start()
```



```

jarvis.py > ...
248         timer = QTimer(self)
249         timer.timeout.connect(self.showTime)
250         timer.start(1000)
251         startExecution.start()
252
253     def showTime(self):
254         current_time = QTime.currentTime()
255         current_date = QDate.currentDate()
256         label_time = current_time.toString('hh:mm:ss')
257         label_date = current_date.toString(Qt.ISODate)
258         self.ui.textBrowser.setText(label_date)
259         self.ui.textBrowser_2.setText(label_time)
260
261
262
263     app = QApplication(sys.argv)
264     jarvis = Main() # object creation
265     jarvis.show()
266     exit(app.exec_())
267

```

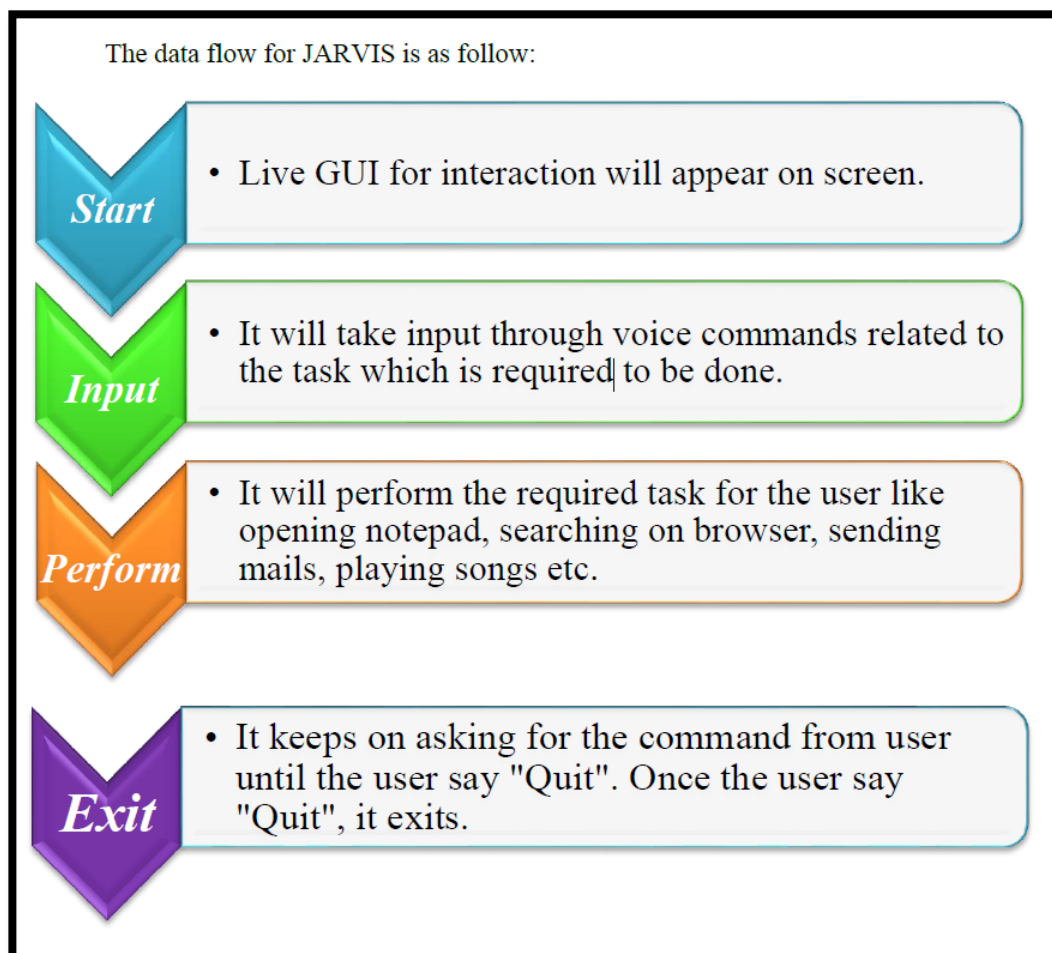
We have added a lot of games also in our Jarvis: these include Color game, pong game, tic tak toe and snake game. Here is the libraries that we used in color game:

COLOR GAME LIBRARIES:

- **tkinter:** Tkinter is **the standard GUI library for Python**. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit. Import the Tkinter module.
- **random:** The random module is a built-in module **to generate the pseudo-random variables**. It can be used perform some action randomly such as to get a random number, selecting a random element from a list, shuffle elements randomly, etc.

- **Pytttsx3:** pytttsx is a cross-platform text to speech library which is platform independent. The major advantage of using this library for text-to-speech conversion is that it works offline.
- **Datetime:** Datetime module **supplies classes to work with date and time.** These classes provide a number of functions to deal with dates, times and time intervals. Date and datetime are an object in Python, so when you manipulate them, you are actually manipulating objects and not string or timestamps.
- **Speech_recognition:** Speech recognition **helps us to save time by speaking instead of typing.** It also gives us the power to communicate with our devices without even writing one line of code.
- **Wikipedia:** Wikipedia is a Python library that makes **it easy to access and parse data from Wikipedia.** Search Wikipedia, get article summaries, get data like links and images from a page, and more. Wikipedia wraps the MediaWiki API so you can focus on using Wikipedia data, not getting it.
- **Web browser:** The webbrowser module provides a high-level interface **to allow displaying Web-based documents to users.**
- **os:** The OS module in Python **provides functions for interacting with the operating system.** 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.

- **Smtplib:** 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.**
- **Colour:** Python's colour module is a simple, yet powerful library **to store, convert and manipulate colors.** Using this library, we can store colors in one of the many formats, namely, RGB, hex codes, HSL, etc. We can also convert colors from one format to the other.
- **Hence finally the dataflow of JARVIS is:**



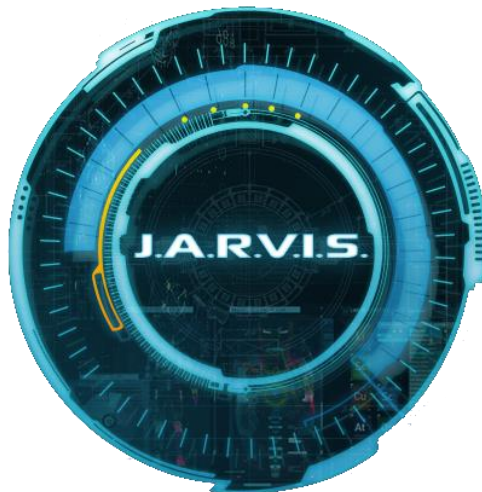


CHAPTER 4

CONCLUSION

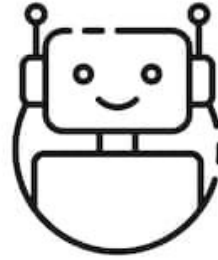
Voice controlled devices use natural language processing to process the language spoken by the human and understand the query and process it to respond to humans with the results. The understanding of devices means Artificial Intelligence needs to be integrated with the devices so that the devices can work in a smart way and can also control IoT applications and devices can also respond to query.

These are designed in general to minimize the human efforts and control the device with just human voice. The future of these assistance includes integrating it with mobile using react native to provide a synchronized experience between the two connected devices. The devices can also be designed to accept commands in bilingual language and to help visually impaired people. The devices can also be designed to interact with other intelligence voice controlled devices like IoT applications and devices. The accuracy of the devices is increasing exponentially in the last decade. We have automated various tasks using python here. It eases most of the tasks like searching on web, getting weather forecast details, playing songs, sending mail, games, jokes etc.





REFERENCES



- [1] <https://www.researchgate.net>
- [2] <https://iimk.ac.in/research/markconf20/Proceedings/220.pdf>
- [3] <http://www.diva-portal.org/smash/get/diva2:531796/FULLTEXT02.pdf>
- [4] <https://www.geeksforgeeks.org/voice-assistant-using-python/>
- [5] <https://www.pwc.com/us/en/services/consulting/library/consumer-intelligence-series/voice-assistants.html>
- [6] <https://www.customsoftwarelab.com/5-predictions-for-the-future-of-voice-assistants-and-ai-in-software-development/>
- [7] <https://www.futureofworkhub.info/explainers/2021/4/19/future-of-voice-assistants-how-the-va-might-overtake-the-pa>
- [8] <https://iimk.ac.in/research/markconf20/Proceedings/220.pdf>