Speaking Assistant

Submitted in fulfillment of the requirements for the award of degree of
BACHELOR OF ENGINEERING
IN
COMPUTER SCIENCE & ENGINEERING

Submitted to:
Mentor name

Submitted By:
AYUSH RANA
19BCS1143
PRAJWAL SHARMA
19BCS1144

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
Chandigarh University, Gharuan
September 2021

CERTIFICATE

This is to certify that the work embodied in this Project Report entitled **"Speaking Assistant"** being submitted by **"Ayush Rana, Prajwal Sharma" -** UID **"19BCS1143, 19BCS1144 " ,** 5th Semester for partial fulfillment of the requirement for the degree of **" Bachelor of Engineering in Computer Science & Engineering "** discipline in " **Chandigarh University** " during the academic session July-Dec 2021 is a record of bonafide piece of work, carried out by student under my supervision and guidance in the **" Department of Computer Science & Engineering ", Chandigarh University.**

APPROVED & GUIDED BY: Geet Kiran Kaur Ma'am

DECLARATION

I, student of **Bachelor of Engineering** in **Computer Science & Engineering, 5th Semester**, **session: August – December 2021, Chandigarh University,** hereby declare that the work presented in this Project Report entitled **"Speaking Assistant"** is the outcome of my own work, is bona fide and correct to the best of my knowledge and this work has been carried out taking care of Engineering Ethics. The work presented does not infringe any patented work and has not been submitted to any other university or anywhere else for the award of any degree or any professional diploma.

Student details

1. Ayush Rana (19BCS1143) CSE 1C

2. Prajwal Sharma(19BCS1144) CSE 1C

APPROVED & GUIDED BY: **Geet Kiran Kaur Ma'am**

ACKNOWLEDGEMENT

I would like to express my gratitude towards **Geet Kiran Kaur Ma'am** for guiding me throughout the project. I also feel thankful and express my kind gratitude towards our college management for allowing me to conduct "**Speaking Assistant"** project. The mentioned project was done under the supervision of **Geet Kiran Kaur Ma'am** I thank all participants for their positive support and guidance.

Finally, I must say that no height is ever achieved without some sacrifices made at some end and it is here where we owe our special debt to our parents and our friends for showing their generous love and care throughout the entire period of time.

1. Ayush Rana (19BCS1143) CSE 1C

2. Prajwal Sharma(19BCS1144) CSE 1C

GUIDED BY – **Geet Kiran Kaur Ma'am**

INTRODUCTION

An **intelligent virtual assistant** (**IVA**) or **speaking assistant is** a software that can perform tasks or services for an individual based on commands or questions. The term "chatbot" is sometimes used to refer to virtual assistants generally or specifically accessed by online chat. In some cases, online chat programs are exclusively for entertainment purposes. Some virtual assistants are able to interpret human speech and respond via synthesized voices. Users can ask their assistants questions, control home automation devices and media playback via voice, and manage other basic tasks such as email, to-do lists, and calendars with verbal (spoken?) commands. A similar concept, however with differences, lays under the dialogue systems.

As of 2017, the capabilities and usage of virtual assistants are expanding rapidly, with new products entering the market and a strong emphasis on both email and voice user interfaces. Apple and Google have large installed bases of users on smartphones. Microsoft has a large installed base of Windows-based personal computers, smartphones and smart speakers. Amazon has a large install base for smart speakers. Convers Ica has over 100 million engagements via its email and SMS interface Intelligent Virtual Assistants for business.

Brief History

The 1990s digital speech recognition technology became a feature of the personal computer with IBM, Philips and Lemout & Hauspie fighting for customers. Much later the market launch of the first smartphone IBM Simon in 1994 laid the foundation for smart virtual assistants as we know them today.

In 1997 Dragon's Naturally Speaking software could recognize and transcribe natural human speech without pauses between each word into a document at a rate of 100 words per minute. A

version of Naturally Speaking is still available for download and it is still used today, for instance, by many doctors in the US and the UK to document their medical records.

In 2001 Colloquis publicly launched SmarterChild, on platforms like AIM and MSN Messenger. While entirely text-based SmarterChild was able to play games, check the weather, look up facts, and converse with users to an extent.

The first modern digital virtual assistant installed on a smartphone was Siri, which was introduced as a feature of the iPhone 4S on 4 October 2011. Apple Inc. developed Siri following the 2010 acquisition of Siri Inc. a spin-off of SRI International, which is a research institute financed by DARPA and the United States Department of Defense. Its aim was to aid in tasks such as sending a text message, making phone calls, checking the weather or setting up an alarm. Over time, it has developed to provide restaurant recommendations, search the internet, and provide driving directions.

In November 2014, Amazon announced Alexa alongside the Echo.

In April 2017 Amazon released a service for building conversational interfaces for any type of virtual assistant or interface.

FEASIBILITY STUDY

Feasibility study can help you determine whether or not you should proceed with your project. 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 Jerry, 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, it 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.

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.

METHODOLOGY

This project contains three steps:

1. Development of GUI Interface for speaking assistant
2. Coding of functions to be deployed for the working of speaking assistant
3. Insertion of queries for the functioning of speaking assistant

MODULE & TEAM MEMBER WISE WORK DISTRIBUTION

This project is divided into two modules:

1. Module-1(GUI Interface): It will be done by **Prajwal Sharma** with the help of **Pyqt5(Python)**
2. Module-2(Functioning): It will be done by **Ayush Rana** with the help of **Python**

HARDWARE AND SOFTWARE REQUIREMENTS

The software is designed to be light-weighted so that it doesn't be a burden on the machine running it. This system is being build keeping in mind the generally available hardware and software compatibility. Here are the minimum hardware and software requirement for speaking assistant.

Hardware:

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

Software:

- Windows 7(32-bit) or above.
- Python 3.0 or later

PROJECT DESIGN

Introduction

This project is basically developed to execute the commands of speaker by listening and processing them if spoken command is already included within the program which includes opening google search engine and its applications, mailing, opening local drives, camera, music, videos, etc.
It is divided in three python files:

1. Commands.py
2. Jerry.py
3. jerry_UI.py

Module 1(GUI Interface**):** It is the first module of this project and will be developed with the help of **pyqt5** (QT Designer) a python tool used to develop python GUI's. Following is the code of the Module 1

Below is the code written in python file jerry_UI.py and is part of module 1

```
# -*- coding: utf-8 -*-

# Form implementation generated from reading ui file 'jerry_UI.ui'

#

# Created by: PyQt5 UI code generator 5.15.4

#

# WARNING: Any manual changes made to this file will be lost when pyuic5 is

# run again.  Do not edit this file unless you know what you are doing.


from PyQt5 import QtCore, QtGui, QtWidgets


class Ui_Jerry(object):
```

```
    def setupUi(self, Jerry):
        Jerry.setObjectName("Jerry")
        Jerry.resize(1344, 832)
        self.centralwidget = QtWidgets.QWidget(Jerry)
        self.centralwidget.setObjectName("centralwidget")
        self.label = QtWidgets.QLabel(self.centralwidget)
        self.label.setGeometry(QtCore.QRect(0, 0, 1341, 831))
        self.label.setText("")
        self.label.setPixmap(QtGui.QPixmap("jerry.gif"))
        self.label.setScaledContents(True)
        self.label.setObjectName("label")
        self.label_2 = QtWidgets.QLabel(self.centralwidget)
        self.label_2.setGeometry(QtCore.QRect(130, 430, 241, 251))
        self.label_2.setStyleSheet("background-color: rgb(0, 0, 0);")
        self.label_2.setText("")
        self.label_2.setPixmap(QtGui.QPixmap("open.gif"))
        self.label_2.setScaledContents(True)
        self.label_2.setObjectName("label_2")
        self.pushButton = QtWidgets.QPushButton(self.centralwidget)
        self.pushButton.setGeometry(QtCore.QRect(1090, 600, 93, 41))
        font = QtGui.QFont()
        font.setPointSize(14)
        font.setBold(True)
        font.setWeight(75)
        self.pushButton.setFont(font)
        self.pushButton.setStyleSheet("color: rgb(0, 0, 0);\n"
"background-color: rgb(0, 255, 0);")
```

```
        self.pushButton.setObjectName("pushButton")
        self.pushButton_2 = QtWidgets.QPushButton(self.centralwidget)
        self.pushButton_2.setGeometry(QtCore.QRect(1210, 600, 93, 41))
        font = QtGui.QFont()
        font.setPointSize(14)
        font.setBold(True)
        font.setWeight(75)
        self.pushButton_2.setFont(font)
        self.pushButton_2.setStyleSheet("background-color: rgb(255, 0, 0);")
        self.pushButton_2.setObjectName("pushButton_2")
        self.textBrowser = QtWidgets.QTextBrowser(self.centralwidget)
        self.textBrowser.setGeometry(QtCore.QRect(30, 40, 291, 61))
        self.textBrowser.setObjectName("textBrowser")
        self.textBrowser_2 = QtWidgets.QTextBrowser(self.centralwidget)
        self.textBrowser_2.setGeometry(QtCore.QRect(1010, 40, 291, 61))
        self.textBrowser_2.setObjectName("textBrowser_2")
        Jerry.setCentralWidget(self.centralwidget)
        self.retranslateUi(Jerry)
        QtCore.QMetaObject.connectSlotsByName(Jerry)


    def retranslateUi(self, Jerry):
        _translate = QtCore.QCoreApplication.translate
        Jerry.setWindowTitle(_translate("Jerry", "MainWindow"))
        self.pushButton.setText(_translate("Jerry", "RUN"))
        self.pushButton_2.setText(_translate("Jerry", "EXIT"))
```

```
if __name__ == "__main__":
    import sys
    app = QtWidgets.QApplication(sys.argv)
    Jerry = QtWidgets.QMainWindow()
    ui = Ui_Jerry()
    ui.setupUi(Jerry)
    Jerry.show()
    sys.exit(app.exec_())
```

Below is the code written in python file Commands.py and is part of module1

```
class jerry_main(QMainWindow) :
    def __init__(self):
        super().__init__()
        self.ui = Ui_Jerry()
        self.ui.setupUi(self)
        self.ui.pushButton.clicked.connect(self.startTask)
        self.ui.pushButton_2.clicked.connect(self.close)

    def startTask(self) :
        self.ui.movie = QtGui.QMovie("jerry.gif")
        self.ui.label.setMovie(self.ui.movie)
        self.ui.movie.start()
        self.ui.movie = QtGui.QMovie("open.gif")
        self.ui.label_2.setMovie(self.ui.movie)
```

```
        self.ui.movie.start()
        timer = QTimer(self)
        timer.timeout.connect(self.showTime)
        timer.start(1000)
        start_jerry.start()

    def showTime(self) :
        now_time = QTime.currentTime()
        now_date = QDate.currentDate()
        label_time = now_time.toString('hh:mm:ss')
        label_date = now_date.toString(Qt.ISODate)
        self.ui.textBrowser.setText(label_date)
        self.ui.textBrowser_2.setText(label_time)
```

Module 2(Functioning): It is the second module of this project and will be developed with the help of python and various modules with the help of python import. Following is the code of the Module 2

Below is the code written in python file Jerry.py and is part of module 2

```
import pyttsx3 as p
import speech_recognition as sr

jerry = p.init('sapi5')
voices = jerry.getProperty('voices')
jerry.setProperty('voices', voices[0].id)
jerry.setProperty("rate", 173)
```

```
def Speak(audio) :
    jerry.say(audio)
    jerry.runAndWait()

def Listen() :
    command = sr.Recognizer()
    with sr.Microphone() as source :
        print("Listening...!")
        command.pause_threshold = 1
        audio = command.listen(source)
        try :
            print("\nRecognizing...!")
            print("It may take some time.")
            query = command.recognize_google(audio, language='en-in')
            print(f"\nYou said : {query}")
        except Exception as e :
            return "none"
        return query.lower()
```

Below is the code written in python file Commands.py and is part of module 2

```
import sys
from PyQt5 import QtWidgets, QtGui, QtCore
from PyQt5.QtCore import QTimer, QTime, QDate, Qt
from PyQt5.QtGui import QMovie
from PyQt5.QtCore import *
```

```
from PyQt5.QtGui import *
from PyQt5.QtWidgets import *
from PyQt5.uic import loadUiType
from pyttsx3 import speak
from jerry_UI import Ui_Jerry
from Jerry import Listen, Speak
import webbrowser as wb
import wikipedia
import datetime
import os
from ecapture import ecapture as ec
import cv2


class jerry_commands(QThread) :
    def __init__(self) :
        super(jerry_commands, self).__init__()

    def run(self) :
        self.take_commands()

    def take_commands(self) :
        Speak("Welcome boss")
        while True :
            self.query = Listen()
            if "wake up" in self.query or "jerry" == self.query :
```

```
                Speak("yes boss, i am awake, Jerry one point o reporting in your service boss, how are you.")

            elif "how are you" in self.query :

                Speak("i am fine sir, what about you")

            elif "fine" in self.query or "good" in self.query or "great" in self.query :

                Speak("it's good to know that you are fine.")

            elif "help" in self.query or "favour" in self.query :

                Speak("Yes boss, how may i help you.")

            elif "i am sad" in self.query or "not fine" in self.query or "not good" in self.query or "not well" in self.query :

                Speak("It's ok boss, sometime situation becomes worst but we have to be strong, so be positive and face your all problems with smile.")

            elif "thank you" in self.query :

                Speak("your welcome sir, i am glad that you are happy with my service")

            elif "made you" in self.query or "created you" in self.query or "your creater" in self.query :

                Speak("I was created by Mister Prajwal Sharma and Mister Ayush Rana.")

            elif "your name" in self.query :

                Speak("My name is jerry one point o, I am a virtual assistant.")

            # start elif from here

            elif "open google" in self.query :

                Speak("Opening google")

                wb.open('www.google.com')

            elif "open youtube" in self.query :

                Speak("Opening youtube")

                wb.open('www.youtube.com')

            elif 'wikipedia' in self.query:
```

```
        Speak('Searching Wikipedia...')
        self.query = self.query.replace("wikipedia", "")
        results = wikipedia.summary(self.query, sentences = 3)
        Speak("According to Wikipedia")
        print(results)
        Speak(results)
    elif 'open stack overflow' in self.query:
        Speak("Opening stackoverflow")
        wb.open("www.stackoverflow.com")
    elif 'the time' in self.query:
        strTime = datetime.datetime.now().strftime("% H:% M:% S")
        Speak(f"Sir, the time is {strTime}")
    elif 'play music' in self.query or "play song" in self.query:
        Speak("Playing music")
        music_dir = "C:\\Users\\ayush\\Music"
        songs = os.listdir(music_dir)
        print(songs)
        random = os.startfile(os.path.join(music_dir, songs[1]))
    elif 'open downloads' in self.query:
        Speak("Opening downloads")
        path=r"C:\\Users\\ayush\\Downloads"
        os.startfile(path)
    elif 'open documents' in self.query:
        Speak("Opening documents")
        path=r"C:\\Users\\ayush\\Documents"
        os.startfile(path)
    elif 'open code blocks' in self.query:
```

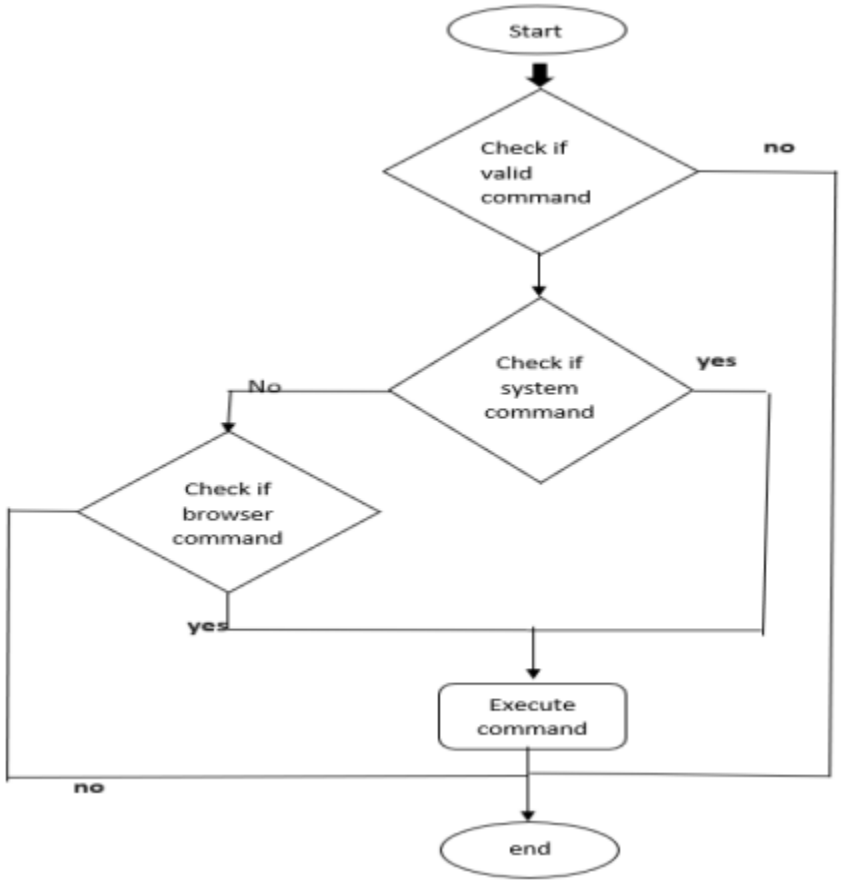
```
                Speak("Opening codeeblocks")
                path=r"C:\\Program Files\\CodeBlocks\\codeblocks.exe"
                os.startfile(path)
            elif "camera" in self.query or "take a photo" in self.query:
                videoCaptureObject = cv2.VideoCapture(0)
                result = True
                while(result):
                    ret,frame = videoCaptureObject.read()
                    cv2.imwrite("NewPicture.jpg",frame)
                    result = False
                videoCaptureObject.release()
                cv2.destroyAllWindows()
            elif 'search' in self.query:
                self.query = self.query.replace("search", "")
                wb.open(self.query)
            elif "exit" in self.query :
                Speak("Thanks boss for your time, Closing me, three, two, one, bye
boss.")
                exit()

start_jerry = jerry_commands()
```

Code written in Commands.py and part of module 1 must be placed here

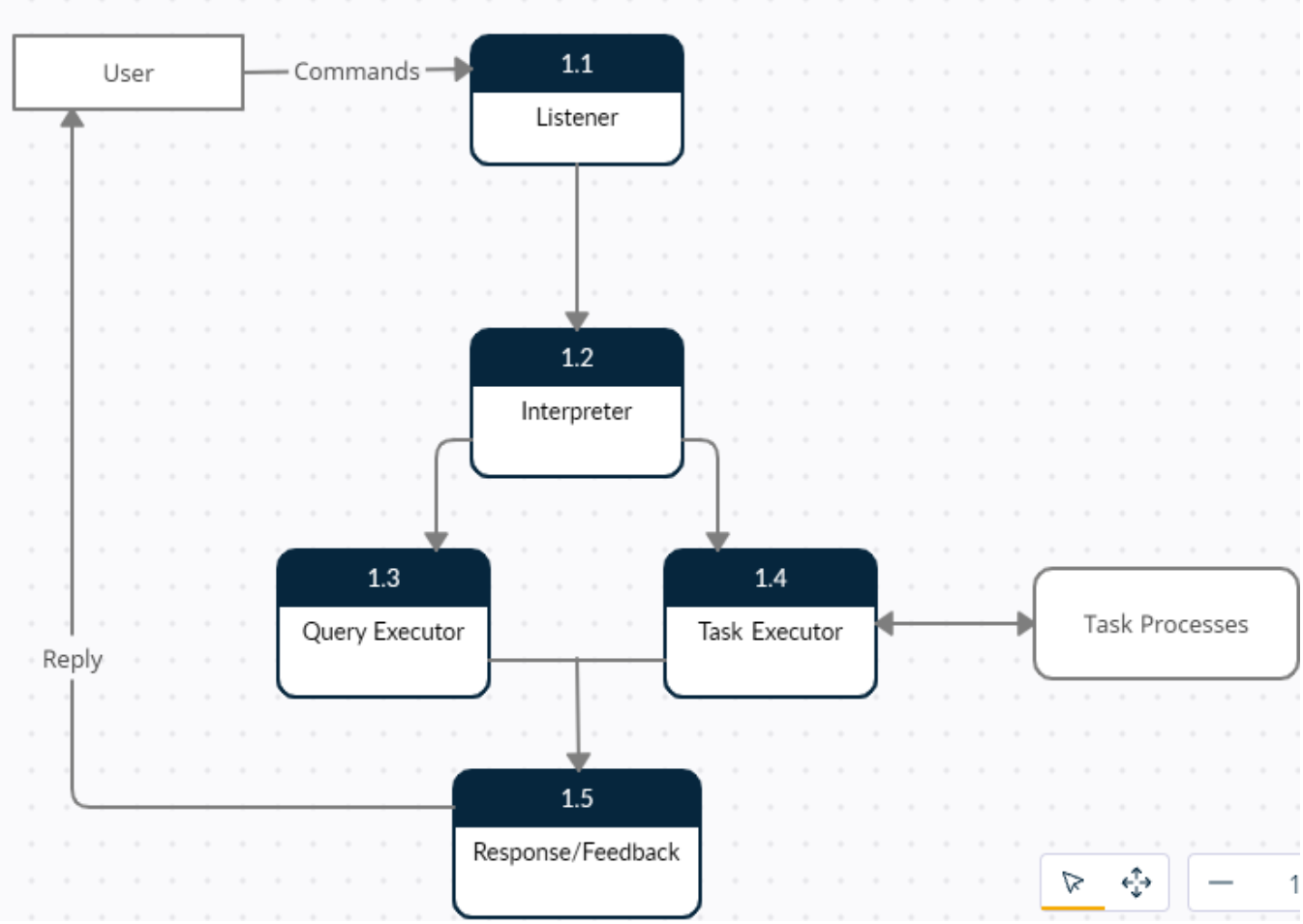
```
app = QApplication(sys.argv)
jerry = jerry_main()
jerry.show()
exit(app.exec_())
```

INNOVATION IN PROJECT

- **Data flowchart**

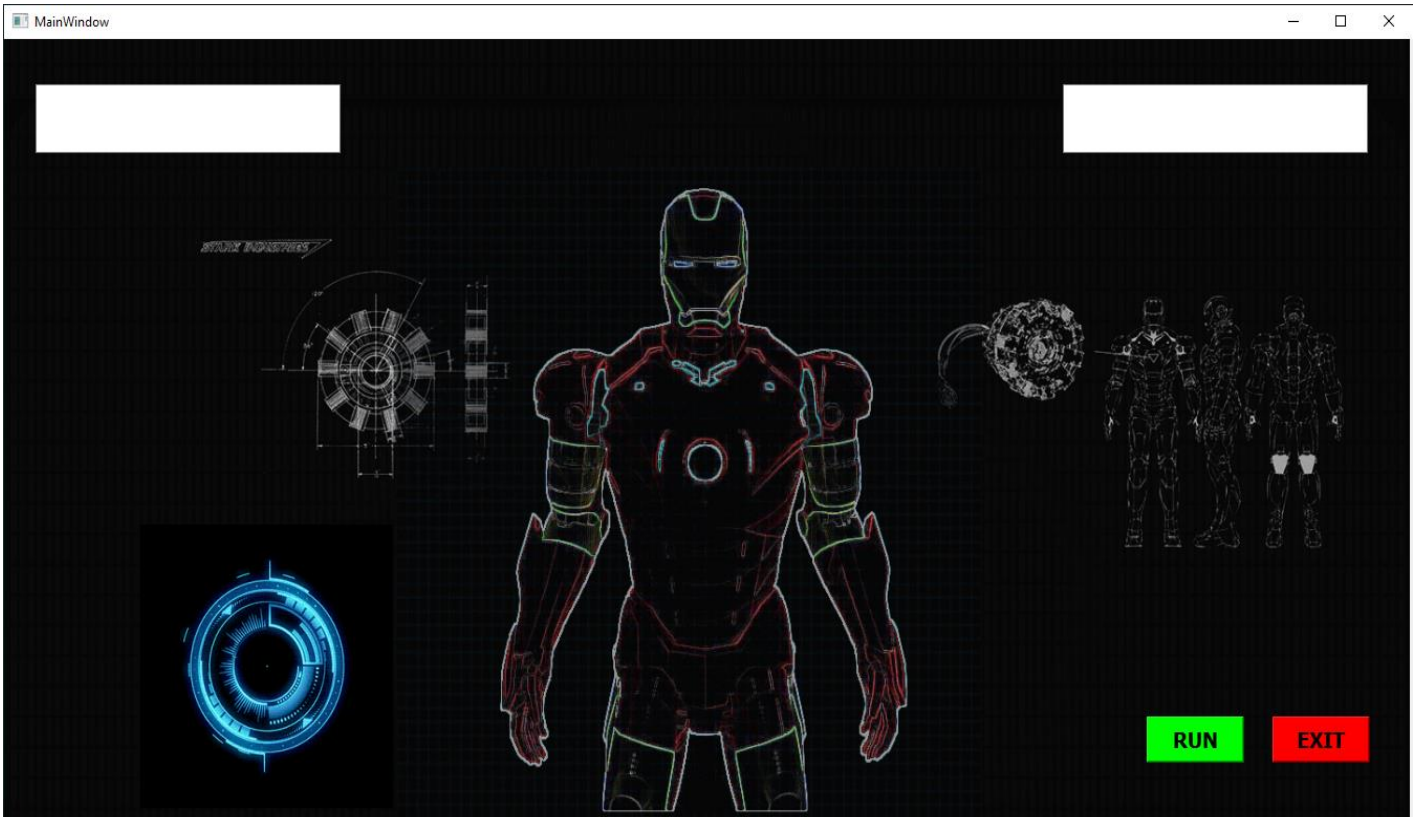


- **Data flow diagram**

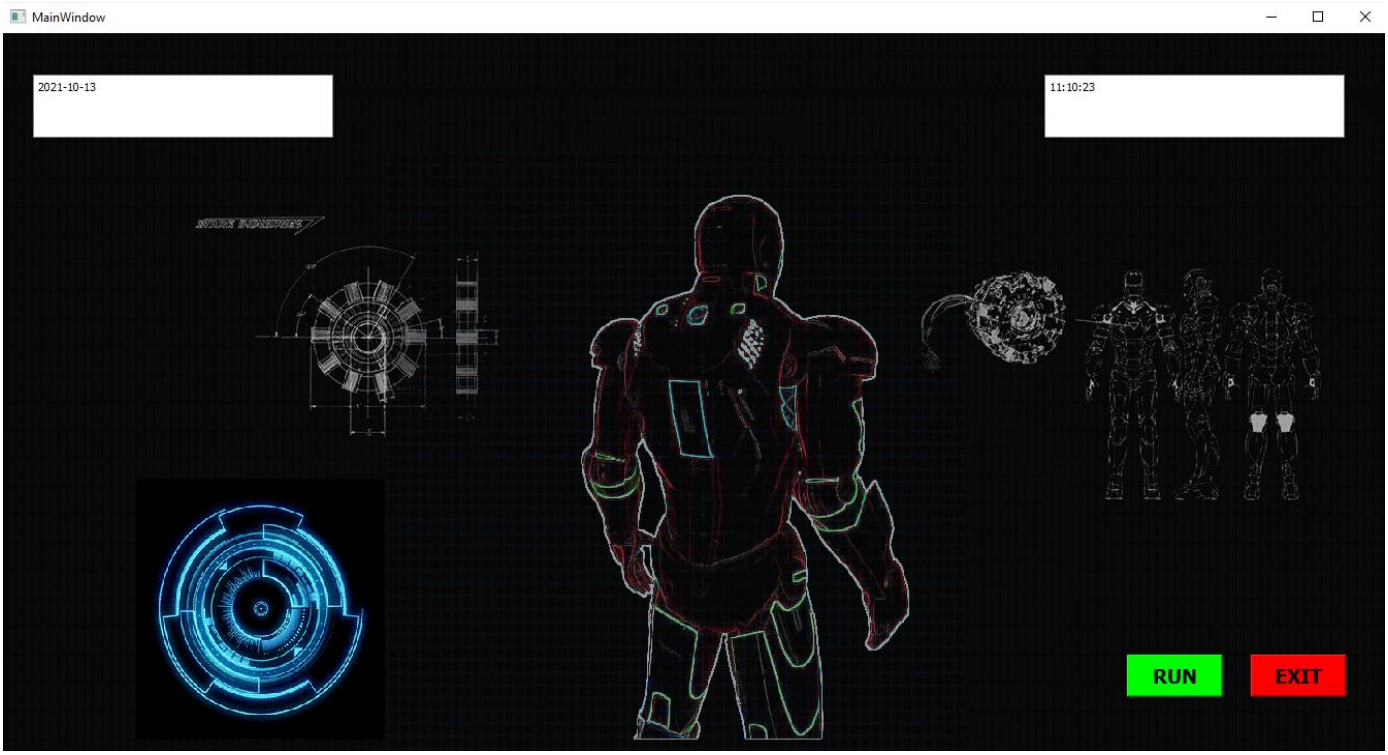


IMPLEMENTATION

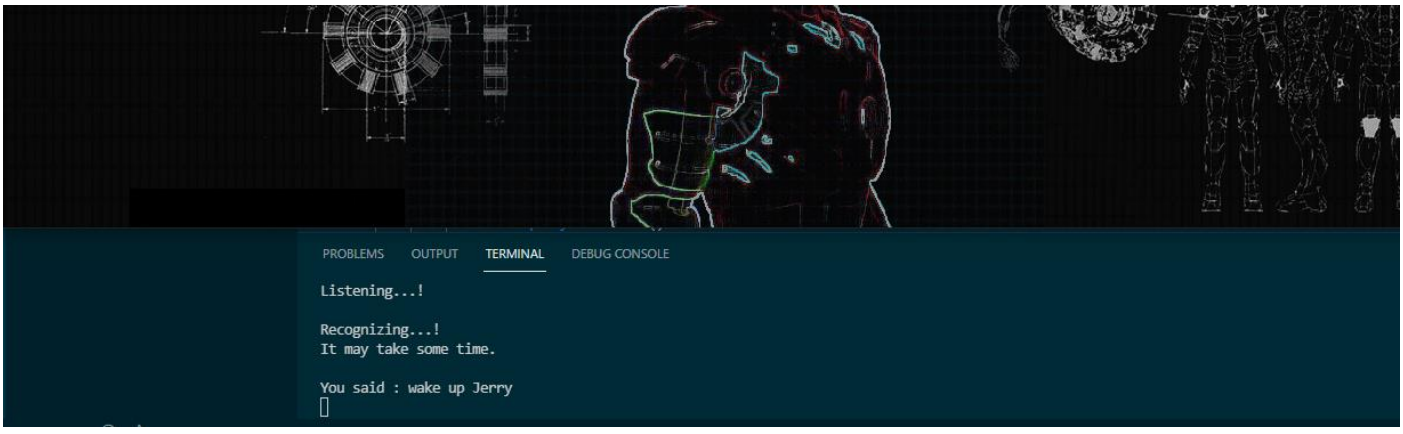
Executing the python file to run the project speaking assistant named jerry



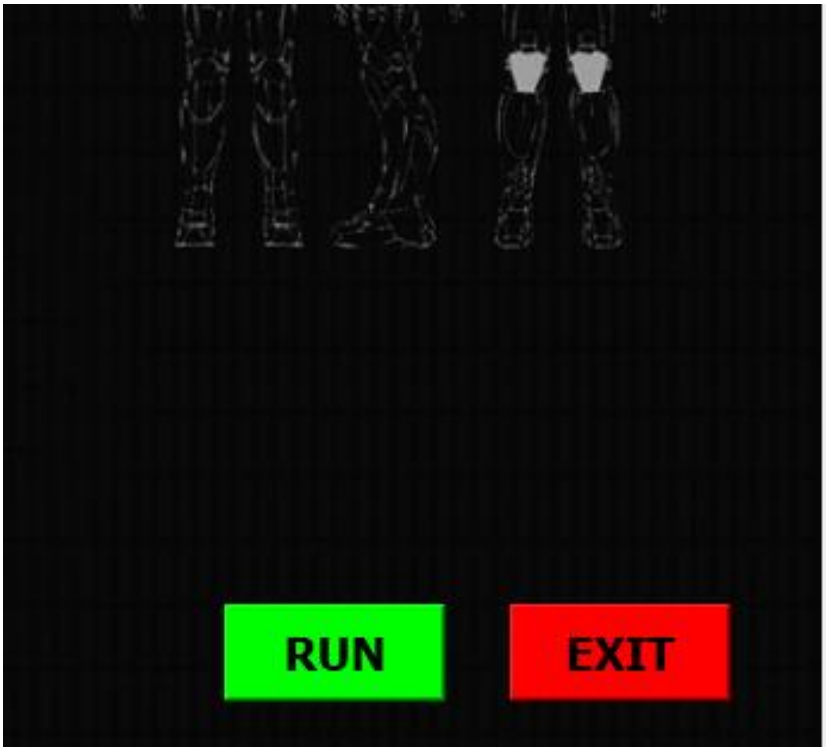
Now you can start the speaking assistant using run button



After that you can start talking with jerry by waking it up through command such as “wake up jerry” or by simply saying “jerry”

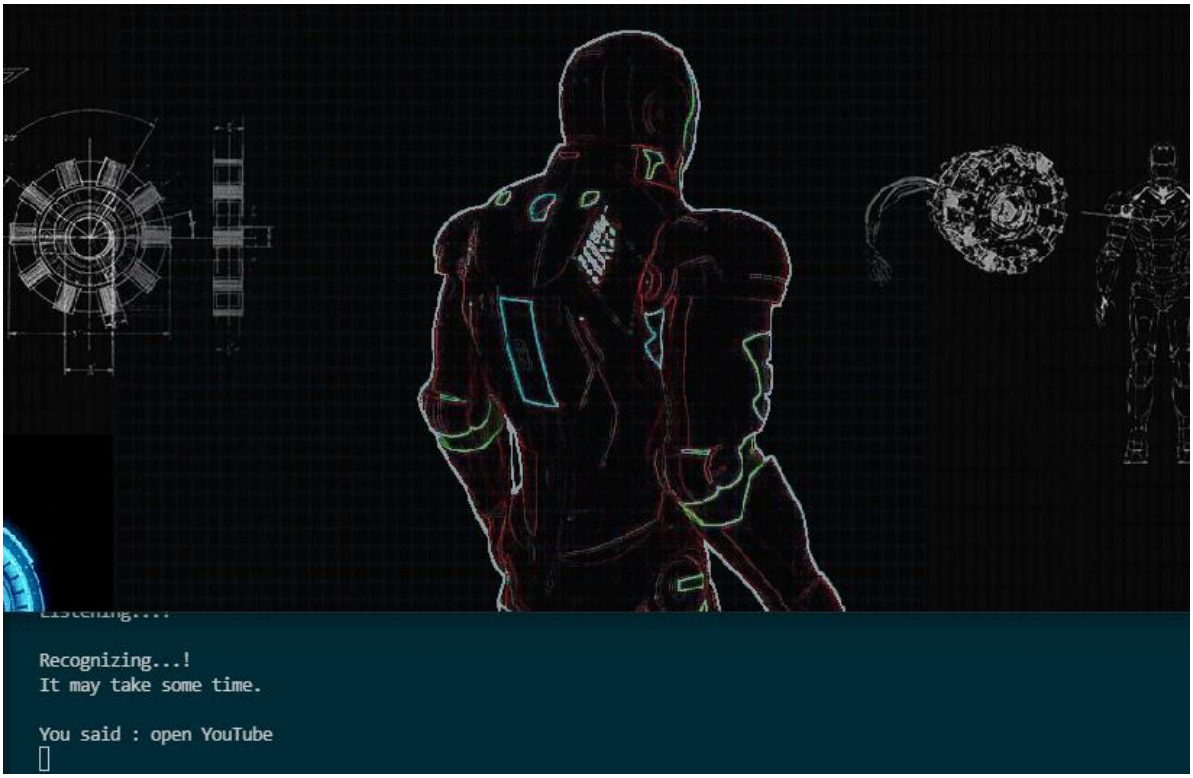


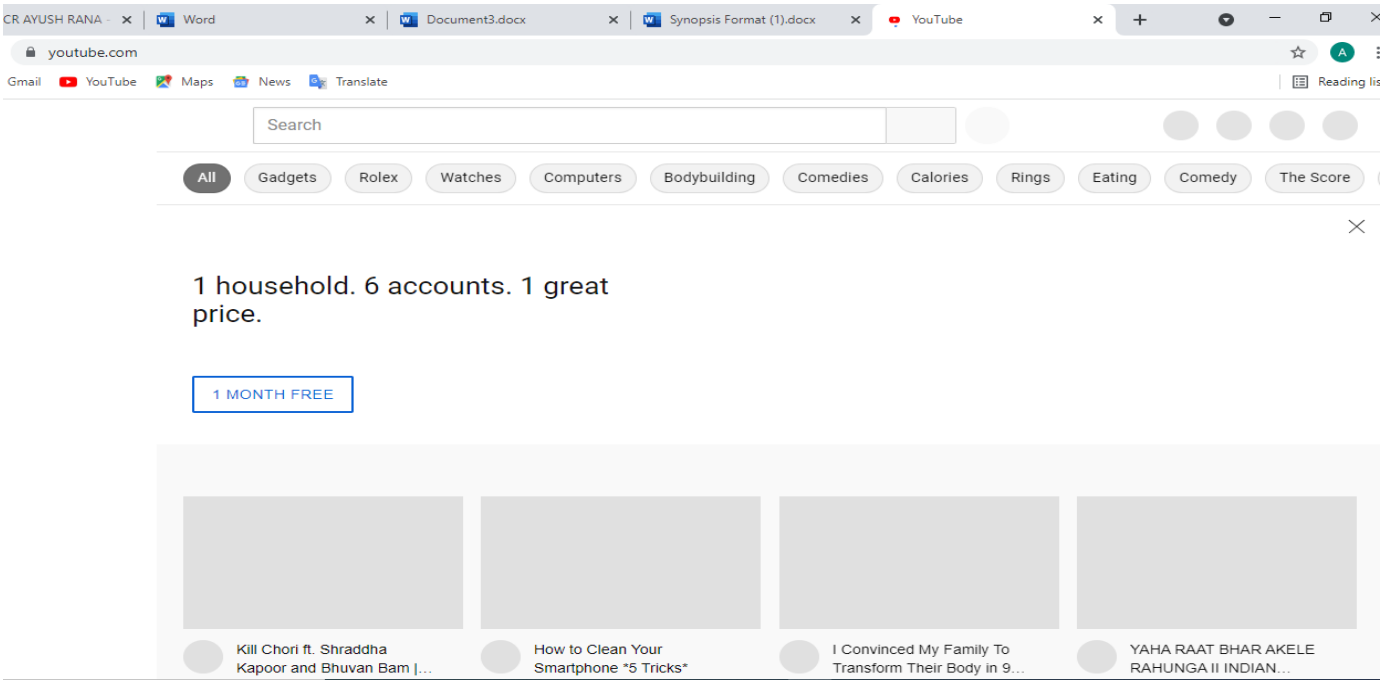
After waking up jerry you can speak with it and can give various tasks that are coded within it and jerry will complete those tasks (only the tasks that are coded in the program). And when you are done with it you can stop it and exit the program with the help of exit button



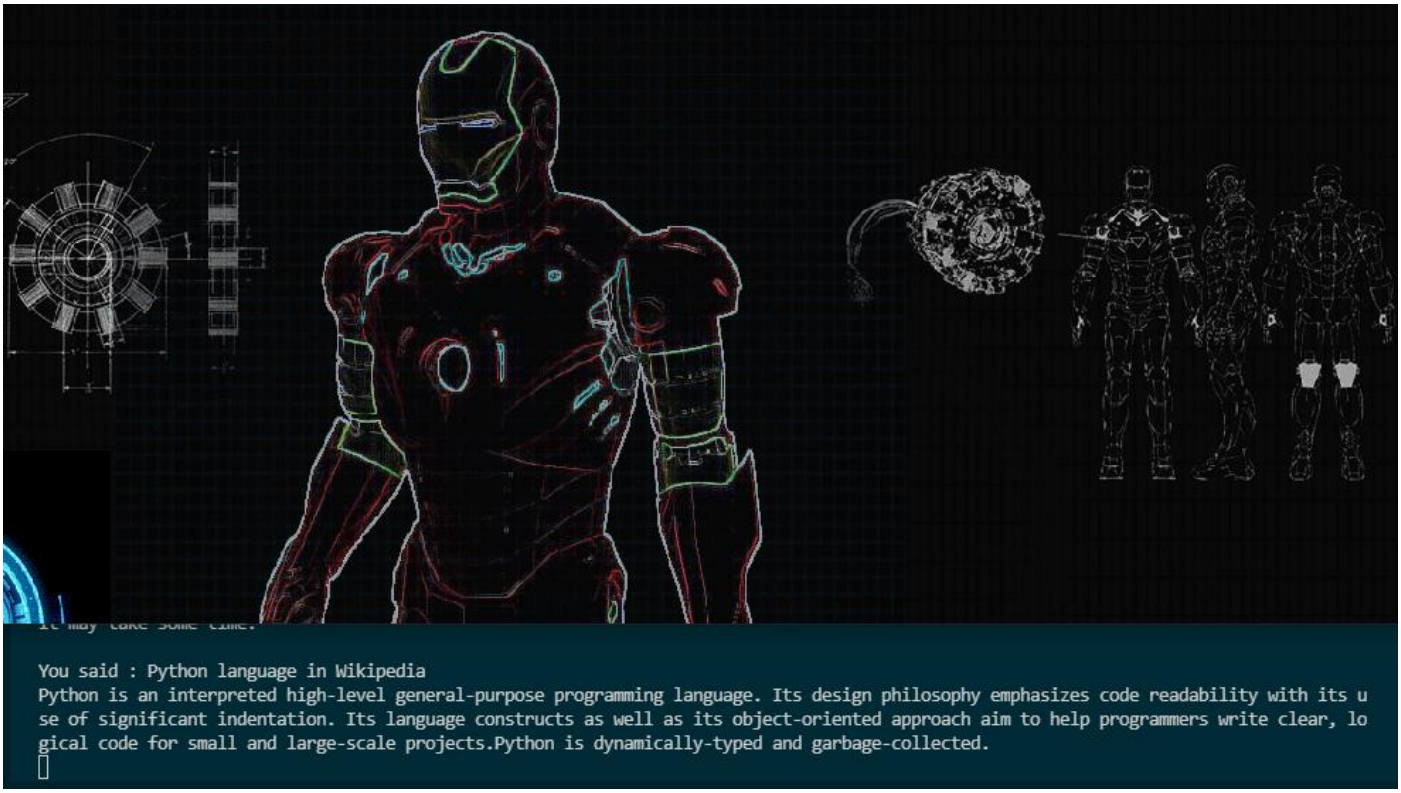
Below are some screenshots of tasks performed by jerry:

1. Opening YouTube in jerry

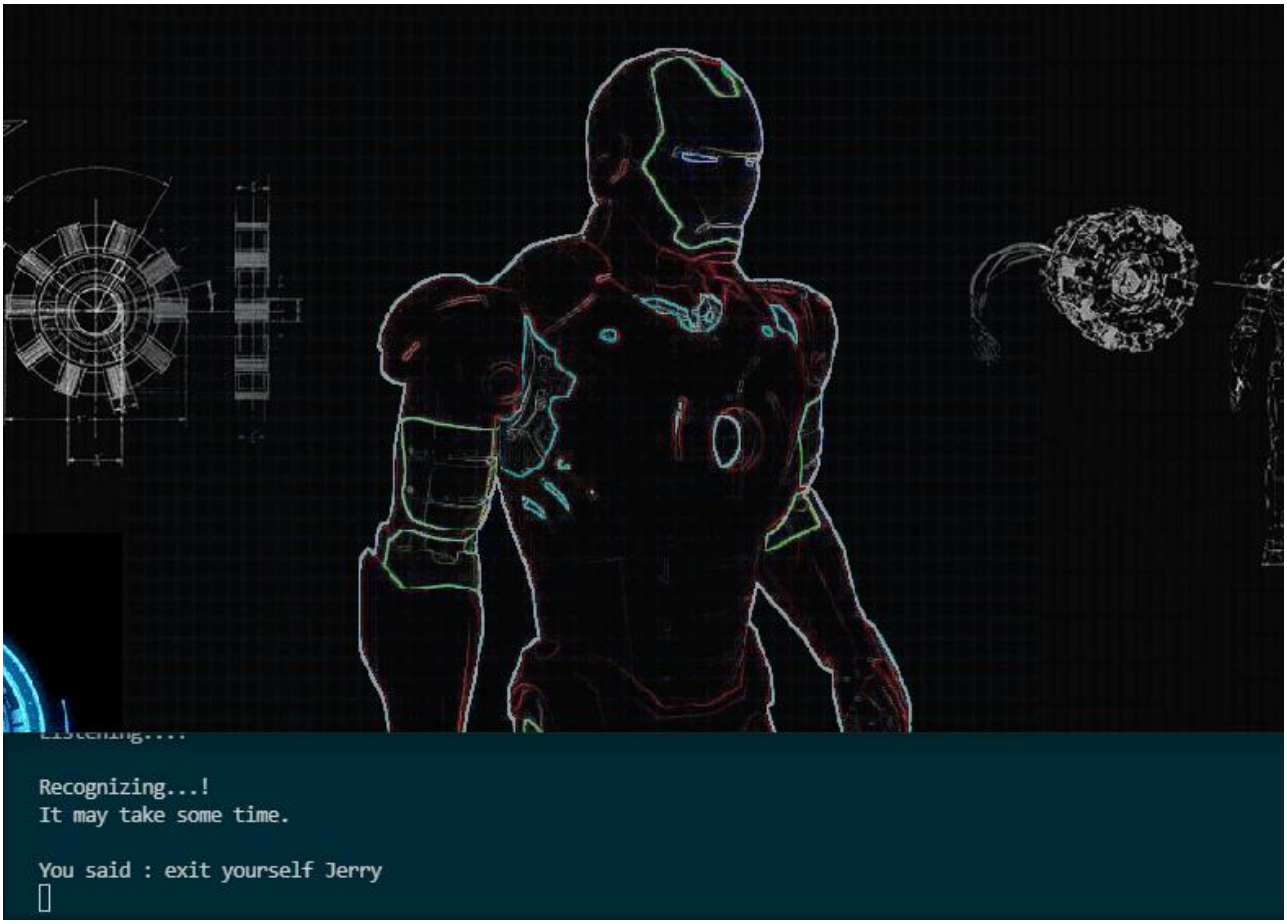

Recognizing...!
It may take some time.
You said : open YouTube


CR AYUSH RANA
Word
Document3.docx
Synopsis Format (1).docx
YouTube
youtube.com
Gmail
YouTube
Maps
News
Translate
Search
All
Gadgets
Rolex
Watches
Computers
Bodybuilding
Comedies
Calories
Rings
Eating
Comedy
The Score
1 household. 6 accounts. 1 great price.
1 MONTH FREE
Kill Chori ft. Shraddha Kapoor and Bhuvan Bam |...
How to Clean Your Smartphone *5 Tricks*
I Convinced My Family To Transform Their Body in 9...
YAHA RAAT BHAR AKELE RAHUNGA II INDIAN...

2. Searching in Wikipedia



3. Exiting the program



And there are many more functions such as opening google, stack overflow, files and folders, application (such as code blocks), searching in web browsers, taking photo and playing music that can be performed or executed by jerry.

REFERENCE AND BIBLIOGRAPHY

• Websites referred

 www.stackoverflow.com

 www.pythonprogramming.net

 www.tutorialspoint.com

 www.google.co.in

• Books referred

 Python Programming - Kiran Gurbani

 Learning Python - Mark Lutz

• YouTube Channels referred

 Code with Harry