CHAPTER 5

PARTIAL CODE

5.1 Code of the project:-

```
import pyttsx3
import speech_recognition as sr
import datetime
import wikipedia
import webbrowser
import os
import smtplib
import googlesearch
import bs4
from bs4 import BeautifulSoup as soup
from urllib.request import urlopen

# Code for using microsoft voice for Voice Assistant
engine = pyttsx3.init('sapi5')
voices = engine.getProperty('voices')
engine.setProperty('voice', voices[0].id)
def speak(audio):
    engine.say(audio)
    engine.runAndWait()

# Code for greeting while being started
def wishMe():
    hour = int(datetime.datetime.now().hour)

    if hour>=0 and hour<5:
        speak("It’s quite late night right now")

    elif hour>=5 and hour<12:
        speak("Good Morning!")
    elif hour>=12 and hour<18:
        speak("Good Afternoon!")

    else:
        speak("Good Evening!")

    speak("Hello, I am Garp. Always at your service. Please tell me how may I
help you.")

def takeCommand():
    #It takes microphone input from the user and returns string output

    r = sr.Recognizer()
    with sr.Microphone() as source:
        print("Listening...")
        r.pause_threshold = 1
        #r.energy_threshold = 300
        audio = r.listen(source)

    try:
        print("Recognizing...")
        query = r.recognize_google(audio, language='en-in')
        print(f"User said: {query}\n")

         except Exception as e:
        # print(e)
        print("Say that again please...")
        return "None"
    return query

def sendEmail(to, content):
    server = smtplib.SMTP('smtp.gmail.com', 587)
    server.ehlo()
    server.starttls()
    server.login('gauravprojects1@gmail.com', 'gmail@123')
    server.sendmail('gauravprojects1@gmail.com', to, content)
    server.close()

if __name__ == "__main__":
    wishMe()
    while True:
        query = takeCommand().lower()



# Code for executing tasks based on searching
        if 'wikipedia' in query:
            speak('Searching Wikipedia...')
            query = query.replace("wikipedia", "")
            results = wikipedia.summary(query, sentences=3)
            speak("Alright...")
            speak("According to Wikipedia")
            print(results)
            speak(results)

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

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

        elif ‘yahoo’ in query:
                speak(“opening yahoo”)
                webbrowser.open(“yahoo.com”)

        elif 'google about' in query:
            try:
                from googlesearch import search

                print("Please tell me what should I google about?")
                speak("Please tell me what should I google about?")
                to_search = takeCommand()
                speak(f"User said: {to_search}\n")
                print("Alright, please wait")
                speak("Alright, please wait")
                speak("According to Google, these are the websites where you can find
the results of your query.")

    for j in search(to_search, tld="com", num=10, stop=4, pause=2):
                    print(j)
                    speak(j)
            except Exception as e:
                print("I am sorry. I cannot find this at the moment.")
                speak("I am sorry. I cannot find this at the moment.")



# Code for opening online shopping sites
        elif 'amazon' in query:
            speak("opening amazon")
            webbrowser.open("amazon.com")

        elif 'flipkart' in query:
            speak("opening flipkart")
            webbrowser.open("flipkart.com")

        elif 'myntra' in query:
            speak("opening myntra")
            webbrowser.open("myntra.com")
```

```
       elif 'snapdeal' in query:
      speak("opening snapdeal")
      webbrowser.open("snapdeal.com")



    elif 'shopclues' in query:
      speak("opening shopclues")
      webbrowser.open("shopclues.com")



    elif 'jabong' in query:
      speak("opening jabong")
      webbrowser.open("jabong.com")

    elif 'ajio' in query or 'ajiyo' in query:
      speak("opening ajio")
      webbrowser.open("ajio.com")

# Code for opening social media sites
    elif 'facebook' in query:
      speak("opening facebook")
      webbrowser.open("facebook.com")

    elif 'instagram' in query:
      speak("opening instagram")
      webbrowser.open("instagram.com")

    elif 'linked' in query:
      speak("opening linkedin")
      webbrowser.open("linkedin.com")

    elif 'whatsapp' in query:
      speak("opening whatsapp")
      webbrowser.open("web.whatsapp.com")
    elif 'twitter' in query:
      speak("opening twitter")
      webbrowser.open("twitter.com")


# Code for using entertainment features
    elif 'music' in query:
      music_dir = 'C:\\Users\\Abhishek Gaurav\\Music'
      songs = os.listdir(music_dir)
      print(songs)
      speak("Alright please wait, playing music for you")
      os.startfile(os.path.join(music_dir, songs[0]))

    elif 'video' in query:
      video_dir = 'D:\\Videos\\Entertainment'
      videos= os.listdir(video_dir)
      print(videos)
      speak("Alright please wait, playing videos for you")
      os.startfile(os.path.join(video_dir, videos[0]))


# Code for asking the current time
    elif 'time' in query:
      strTime = datetime.datetime.now().strftime("%H:%M:%S")
      speak(f"Alright, the time is {strTime}")
        print(f"Alright, the time is {strTime}")

# Code for executing any app installed in system
    elif 'notepad++' in query:
      codePath = "C:\\Program Files\\Notepad++\\notepad++.exe"
        speak("Alright, opening notepad++ for you")
        print("Alright, opening notepad++ for you")
      os.startfile(codePath)
```

```
# Code for asking Voice Assistant to quit
        elif 'quit' in query or 'bye' in query or 'terminate' in query:
            print("Thank you very much for your time. Good Bye.")
            speak("Thank you very much for your time. Good Bye.")
            exit()



# Code for sending email
        elif 'email' in query or 'mail' in query:
            try:
                speak("To whom?")
                speak("Please type receipient address manually.")
                to = str(input("Receipient address:-"))
                speak("What should I say?")
                content = takeCommand()
                sendEmail(to, content)
                speak("Email has been sent successfully!")
                print("Email has been sent successfully!")
            except Exception as e:
                print(e)
                speak("Sorry Sir. I am not able to send this email. Please try again")

        elif 'news' in query:
            news_url = "https://news.google.com/news/rss"
            Client = urlopen(news_url)
            xml_page = Client.read()
            Client.close()
            speak("Please wait for a moment")

            soup_page = soup(xml_page, "xml")
            news_list = soup_page.findAll("item")

            speak("Alright, these are a few links to the some news. You can click on
anyone to open it in webbrowser.")

# Print news title, url and publish date
            for news in news_list:
                print(news.title.text)
                print(news.link.text)
                print(news.pubDate.text)
                print("-" * 10)
```