**VIRTUAL PERSONAL ASSISTANT**

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

**BACHELOR OF ENGINEERING**

**IN**

**COMPUTER SCIENCE & ENGINEERING**

****

**Submitted to:**

**ER.TANU DHIMAN**

**Submitted By:**

**DIVYANSH SINGH VIVEK KUMAR SINGH**

**18BCS1560 18BCS1557**

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**Chandigarh University, Gharuan**

**SEPTEMBER 2020**

**VIRTUAL PERSONAL ASSISTANT**

**(ORAK)**

Virtual Personal Assistant (VPA) is the next generation of carrier services for mobile users. VPA is believed to be the intelligent evolution of services to meet the ever increasing demand by the mobile professionals for mobility and connectivity. This new generation of services will allow mobile users to remotely access and manage information using speech recognition technology over telephones. VPA responds to conversational voice commands and delivers a single point of contact that seamlessly engages a wide range of information. The VPA controls the telephone calls, manages the personal activities through calendar, enables the user to access his task manager via voice interface, and includes all the functions of Unified Messaging. The VPA enables the user to optimize the user resources (time, cost), enhance his/her overall productivity, and minimize the interruptions to his regular workflow. The Virtual Personal Assistant (VPA) will enable the user to efficiently handle increasing demand of telephone calls, messages, meetings and other activities. The paper provides an overview of the VPA applications, and the expected features and future trends. The paper proposes as well a unified decision model based on a quantitative assessment of the importance of the requests and the availability of the user.

This system is voice-based personal assistant has always seemed a little out of place in the enterprise. It’s a useful tool for search, for reminders, and to write the note just by speaking it up. Window assistant is to create voice apps for the intelligent assistant. When user need to open any other application, he/she can use the command open. E.g. Open Notepad, File explorer, goggle chrome, this will help to open the applications. When user want to write the message can use command write. And to for searching purpose search command can be use. It will also restart and shutdown on the command. It will detect the speech and save in the database, and retrieve from the database and executive command and delete it from database. Interactions between a user and your Window assistant skill are mostly free-form, so assistant must understand language naturally and also in context. Window assistant determines what a user wants to do by identifying the user intent from spoken or textual input by utterance. The intent maps utterances to actions that Window assistant can take, such as invoking a dialog.

Why we need A Virtual Assistant –

Virtual assistants are everywhere these days. More and more people are turning to them and it looks like they’re here to stay. Here’s why you should hire one yourself.

**1. Better Use of Your Employees**

Say a mid-level or senior-level employee had to choose between these two options:

1. Playing an active role in spearheading growth strategies

2. Preparing routine reports on a daily basis

What would they choose?

It’s more interesting and seems more important.

But that doesn’t mean you can neglect these mundane tasks like emails and finances. Your business would crumble without this boring repetitive work.

So how do you keep your in-house team happy but still get the mundane tasks done?

With virtual assistants.

They can take up most of these mundane tasks that can be done remotely.

The result?

A full-time employee (especially those you are more senior or talented) can stay focused on more pressing matters that need to be done in-person at your office.

**2. Reduced Costs**

One of the main reasons why entrepreneurs and businesses outsource work is that it costs less.

Saving Salary Cost

Instead of hiring a full-time employee for a small job, you could choose to outsource it to someone at an hourly rate.

Lower pay isn’t the only way you can save money with virtual professionals though.

Saving Training Cost

A business owner can also save a lot when it comes to training costs.

Usually, companies have to train their in-house team on tasks and duties after they’ve cleared the hiring process.

All this costs time and money – that most companies can’t afford.

With VAs though, small business owners can hire professionals who already experts in their field.

This way, a business owner doesn’t have to waste time and money training them – they can get started right away.

**3. Saves Time**

Another reason why small business owners turn to VAs is that it saves them a LOT of time.

We all know that time is money – especially in business.

For example, let’s say you are a small business with no HR team.

Your employees shouldn’t spend hours posting job advertisements, doing preliminary resume screening, organizing an interview process – all on a hiring process for someone at entry-level position.

Why not skip all this and get a professional to work on specific tasks instead?

With virtual assistants, this is what you’ll get:

• You’ll have verified experts that you can hire in no time

• They’re time savers as you don’t have to waste time training them

• As they can stay focused only on tasks you’ve hired them for, they’ll get the work done quicker than someone juggling multiple demands

**PROJECT DESIGN**

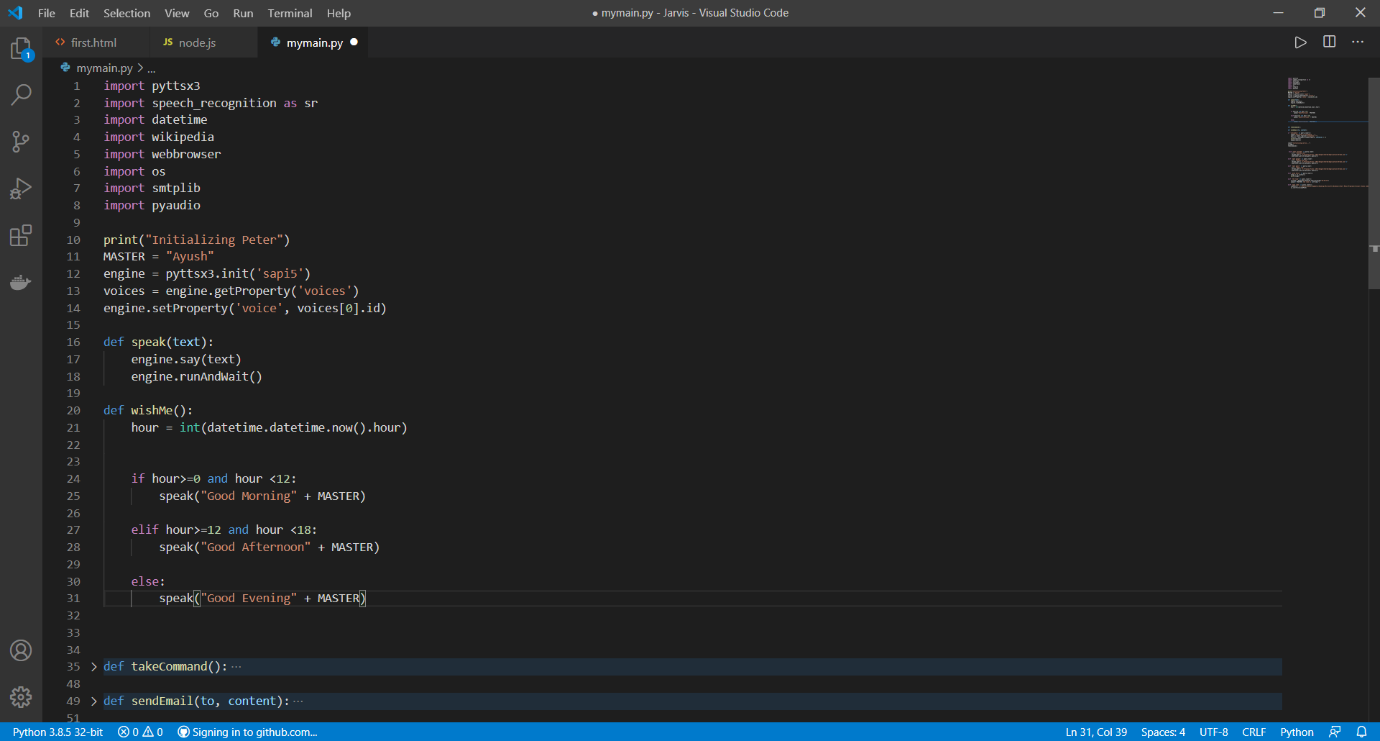
**1)Visual Studio Code**: Visual Studio Code is a free source-code editor made by Microsoft for Windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. Users can change the theme, keyboard shortcuts, preferences, and install extensions that add additional functionality.

Visual Studio Code's source code comes from Microsoft's free and open-source software VSCode project released under the permissive Expat License, but the compiled binaries are freeware for any use.

Visual Studio Code is a source-code editor that can be used with a variety of programming languages, including Java, JavaScript, Go, Node.js and C++. It is based on the Electron framework, which is used to develop Node.js Web applications that run on the Blink layout engine. Visual Studio Code employs the same editor component used in Azure DevOps.

Instead of a project system, it allows users to open one or more directories, which can then be saved in workspaces for future reuse. This allows it to operate as a language-agnostic code editor for any language. It supports a number of programming languages and a set of features that differs per language. Unwanted files and folders can be excluded from the project tree via the settings. Many Visual Studio Code features are not exposed through menus or the user interface, but can be accessed via the command palette.

Visual Studio Code can be extended via extensions, available through a central repository. This includes additions to the editor and language support. A notable feature is the ability to create extensions that add support for new languages, themes, and debuggers, perform static code analysis, and add code linters using the Language Server Protocol



**2)Python:**

As we know Python is a suitable language for script writers 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.

Speech recognition is the process of converting audio into text. This is commonly used in voice assistants like Alexa, Siri, etc. Python provides an API called **SpeechRecognition** to allow us to convert audio into text for further processing. In this article, we will look at converting large or long audio files into text using the SpeechRecognition API in python.



**Modules needed**

* **Subprocess:-** This module is used for getting system subprocess details which are used in various commands i.e. Shutdown, Sleep, etc. This module comes buit-in with Python.
* **Pyttsx3:-** This module is used for 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**

* **Wikipedia:-** As we all know Wikipedia is a great source of knowledge just like GeeksforGeeks we have used Wikipedia module to get information from Wikipedia or to perform 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. To install this module type the below command in the terminal.

**Pip install speechRecognition**

* **Web browser:-** To perform Web Search. This module comes buit-in with Python.
* **Datetime:-** Date and Time is used to showing Date and Time. This module comes built-int with Python.
* **Os:-** This module provides a portable way of using operating system dependent functionality. If you just want to read or write a file see open().
* **Smtplib:**- The smtplib module defines an SMTP client session object that can be used to send mail to any Internet machine with an SMTP or ESMTP listener daemon

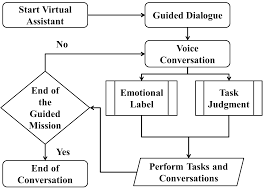
**INNOVATION IN MODEL/DESIGN/SOLUTION**

Advancements in artificial intelligence have [given rise to in-home virtual assistants](https://www.washingtonpost.com/news/the-switch/wp/2017/01/06/the-big-takeaway-from-this-years-ces/?tid=a_inl&utm_term=.4eb26b251f48&itid=lk_inline_manual_9), devices that listen and respond as we can command them to turn off the lights, purchase items online or order restaurant takeout. Amazon Echo and Google Home, two popular systems, can now be found in millions of homes.

ElliQ (pronounced L-E-Q) represents a new role for these technologies: proactively recommending ways in which humans could be living better lives, from getting more exercise to watching informational videos. Humans may not be taking direct orders from their technology, at least not yet, but it nevertheless suggests an emerging relationship where smart devices wield even greater influence over our decisions.

Indeed, machines prod humans all day. Your alarm rings to keep you from sleeping through a morning meeting. Your car beeps when you’ve started the engine but haven’t clipped your seat belt. Your Netflix account suggests movies to watch based on your viewing history.

Virtual assistant robots are different in that they have a broader view of our daily lives and are designed to help us accomplish tasks. They can already learn when we typically wake up and go to sleep, what we watch on television and what we purchase online. As the devices become capable of doing even more, they will store and analyze that information, too.



FLOWCHART OF THE PROJECT

The key is that we invite those technologies to nag us and that we have control over them. We set the alarm clock ourselves — and have the power to hit snooze.

“We have a whole set of words for talking about this in English: persuade, hint, advocate, encourage,” Mark said. “There’s all kinds of things that have a wide variety of implications and very different feelings that are generated by it.”

ElliQ monitors the user’s movements and learns their patterns to ensure its suggestions are well-timed, Skuler said. The user might prefer to take walks in the morning rather than after lunch or value quiet time in the evening over listening to music.

Currently, ElliQ is programmed with seven goals that the user can choose among, such as learning something new each day, being more physically active or communicating with family more often. The company sets one of the goals for you: developing a “positive affinity” for the robot.

Developing machines that can persuade people to act in a certain way is both a technological and psychological challenge, Mark said. Even humans struggle to know when advice will be well received and deliver it in a way that actually motivates the recipient.

**IMPLEMENTATION**

This Virtual assistant will be performing certain functions when we run our python code and performs specific functions according to the commands given by the user.  
Virtual assistant will wait till he is asked to perform some activity

It will wait for sometime to recognize the audio and if it is not clear or audible then the user will be asked to give a command again.70% of the work has been done and few innovations are left which are expected to be implemented soon.

There are variety functions and all the functions are called when we run the code and our Assistant is working successfully as shown below.

**CODE:-**

import pyttsx3import speech\_recognition as srimport datetimeimport wikipediaimport webbrowserimport osimport smtplibengine = pyttsx3.init('sapi5')voices = engine.getProperty('voices')# print(voices[1].id)engine.setProperty('voice', voices[0].id)def speak(audio): engine.say(audio) engine.runAndWait()def wishMe(): hour = int(datetime.datetime.now().hour) 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 orak ,and I made by a team of two student,Divyansh and Vivek,Please tell me how may I help You")def takeCommand(): # it take microphone from the user and returns string out put r = sr.Recognizer() with sr.Microphone() as source: print("Listening...") r.pause\_threshold = 1 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 querydef sendEmail(to, content): server = smtplib.SMTP('smtp.gmail.com', 587) server.ehlo() server.starttls() server.login('divyanshsinghhacked11@gmail.com', 'hello12')# as email will be not able to send because password were wrong server.sendmail('divyanshsinghhacked11@gmail.com', to, content) server.close()if \_\_name\_\_ == "\_\_main\_\_": wishMe() while True: # if 1: query = takeCommand().lower() # logic for executing tasks base on query if 'wikipedia' in query: speak("Searching wikipedia...") query = query.replace("wikipedia", "") results = wikipedia.summary(query, sentences=2) speak("According to wikipedia") print(results) speak(results) elif 'open google' in query: webbrowser.open("google.com") elif 'open stackoverflow' in query: webbrowser.open("stackoverflow.com") elif 'play music' in query: music\_dir = 'D:\\Non Critical\\songs\\Favorite 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"Sir,the time is {strTime} ") print(strTime) elif 'open code' in query: codePath = "C:\\Program Files\\Sublime Text 3\\sublime\_text.exe" os.startfile(codePath) elif 'email to divyansh' in query: try: speak("what would I say ?") content = takeCommand() # it will give string to = "divyanshsinghhacked11@gmail.com" sendEmail(to, content) speak("Email has been sent!") except Exception as e: print(e) speak("sorry sir divyansh and vivek ,I am not able to send this email")

**SNAPSHOOT OF CODE:-**

**PROJECT-AI base assistance by using python (tools & techic)**

**ORAK**

import pyttsx3  
import speech\_recognition as sr  
import datetime  
import wikipedia  
import webbrowser  
import os  
import smtplib  
  
engine = pyttsx3.init('sapi5')  
voices = engine.getProperty('voices')  
# print(voices[1].id)  
engine.setProperty('voice', voices[0].id)  
  
  
def speak(audio):  
 engine.say(audio)  
 engine.runAndWait()  
  
  
def wishMe():  
 hour = int(datetime.datetime.now().hour)  
 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 orak ,and I made by a team of two student,Divyansh and Vivek,Please tell me how may I help You")  
  
  
def takeCommand():  
 # it take microphone from the user and returns string out put  
  
 r = sr.Recognizer()  
 with sr.Microphone() as source:  
 print("Listening...")  
 r.pause\_threshold = 1  
 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('divyanshsinghhacked11@gmail.com', 'hello12')# as email will be not able to send because password were wrong  
 server.sendmail('divyanshsinghhacked11@gmail.com', to, content)  
 server.close()  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 wishMe()  
 while True:  
 # if 1:  
 query = takeCommand().lower()  
  
 # logic for executing tasks base on query  
 if 'wikipedia' in query:  
 speak("Searching wikipedia...")  
 query = query.replace("wikipedia", "")  
 results = wikipedia.summary(query, sentences=2)  
 speak("According to wikipedia")  
 print(results)  
 speak(results)  
 elif 'open google' in query:  
 webbrowser.open("google.com")  
  
 elif 'open stackoverflow' in query:  
 webbrowser.open("stackoverflow.com")  
  
 elif 'play music' in query:  
 music\_dir = 'D:\\Non Critical\\songs\\Favorite 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"Sir,the time is {strTime} ")  
 print(strTime)  
  
 elif 'open code' in query:  
 codePath = "C:\\Program Files\\Sublime Text 3\\sublime\_text.exe"  
 os.startfile(codePath)  
  
 elif 'email to divyansh' in query:  
 try:  
 speak("what would I say ?")  
 content = takeCommand() # it will give string  
 to = "divyanshsinghhacked11@gmail.com"  
 sendEmail(to, content)  
 speak("Email has been sent!")  
 except Exception as e:  
 print(e)  
 speak("sorry sir divyansh and vivek ,I am not able to send this email")