**Industrial Training Report**

**(Subject Code: 15CS390L)**

**Virtual Personal Assistant**

**Submitted By-**

Shreeansh Bhatnagar

RA1711003030324

**Underwent At-**

****

**In Partial Fulfilment of-**

Bachelor of Technology (Computer Science and Engineering)

**Submitted to-**

**Department of Computer Science and Technology**

**Faculty of Engineering and Technology**

**SRM Institute of Science and Technology**

Modinagar,Delhi-NCR-201204

3rd Year(July-2019)

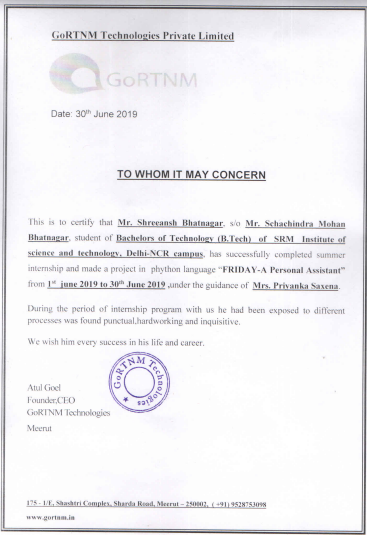
**Declaration**

*I hereby declare that this Industrial Training Report on the “Virtual Personal Assistant ”, submitted to the SRM Institute of Science & Technology, Delhi-NCR Campus for the partial fulfilment of the requirements for the award of the degree of Bachelor of Technology in Computer Science & Engineering, is my original work and has not formed the basis for the award of any other degree, diploma, fellowship or any other similar titles.*

*Shreeansh Bhatnagar*

***(RA1711003030324)***

**Company Certificate**

****

**BONAFIDE CERTIFICATE**

**Reg. No.: RA1711003030324**

*Certified to be the Bonafide record of work done by* ***Shreeansh Bhatnagar*** *of 5th semester 3rd year of B. TECH degree course in SRM INSTITUTE OF SCIENCE & TECHNOLOGY, DELHI-NCR Campus for the Department of* ***Computer Science & Engineering,*** *in* ***Industrial Training*** *during the academic year* ***2019-2020.***

**Faculty in Charge Head of the Department**

Mr. Naresh Sharma Dr .R.P Mahapatra

Assistant Professor H.O.D

Computer Science and Engg. Computer Science and Engg.

**ACKNOWLEDGEMENT**

*I take this opportunity to express my heart-felt gratitude to Subject Coordinators for their constant encouragement, able guidance and support throughout the course of this semester.*

*I sincerely thank Mr. Naresh Sharma for his advice and support during the course of this work.*

*I take this opportunity to thank the Head of Department, Computer Science & Engineering,*

*Dr. Rajendra Prasad Mahapatra & express my gratitude towards our parents, colleagues and friends for their kind support during the completion of work.*

*I would like to thank* *Mrs. Priyanka Saxena, Director, GORTNM Innovations and Technologies for guiding me during the course of the training.*

**INDEX**

* *Declaration*
* *Bonafide Certificate*
* *Company Certificate*
* *Acknowledgement*

1. About Company……………………………………………………………………………………. 7
2. Project Introduction.……………………………………………………………………………… 8
3. Project Functionality……………………………………………………………………………… 9
4. Python Libraries Used…………………………………………………………………………… 12
5. Source Code………………………………………………………………………………………….. 15
6. Module Descriptions……………………………………………………………………………… 24
7. GUI Explanation………………………………………………………………………………………. 30
8. Project Applications…………………………………………………………………………………. 31
9. Output………………………………………………………………………………………………………32
10. Result………………………………………………………………………………………………. 33

**About Gortnm**



*GoRTNM Technologies is software company (IT Company) offering Best Secure Websites & Web Services & Software Services in Meerut & Delhi NCR. Services: Websites & Web Applications, Backend & FrontEnd Services, Desktop Applications, Mobile Applications. Features: Free one year subscription of Server and domain. Technologies: NodeJS, MonogDB, Amazon AWS, PHP, MySQL, PosGreSQL, Android, HTML5, CSS, JavaScript, JQuery, ElectronJS, C++, QT, Python, CSharp, React, Ruby on Rails, Java, Joomla, JSON, Drupal, Magento, WordPress, IOS, Swift, Ionic. GoRTNM Technologies is a privately owned software company offering BackEnd Services, Amazon Web Services, FrontEnd Services, ECommerce Platforms, Web Solutions, Desktop Applications, Mobile applications, Graphic Design and Digital Marketing in the field of software development since 2017. Our software outsourcing services and solutions are designed with a focus on secure, scalable, expandable and reliable business systems. Through our low cost, high quality and reliable software services, we serve our clients giving them value for money and thus client satisfaction which is our first motive.*

**Project Introduction**

*The basic idea behind this project is to create a simple stand-alone application that helps less tech savvy people in the world to use the computer without feeling ignorant or computer illiterate.  
Computers have became a very important devices and as well as less expensive over time. The application works same like Siri/ Google Assistant etc. But the application deals with the computer itself mainly. The U.I of the application is self-explanatory and minimal. All the functions of the software are represented in the GUI using images that clearly explains the functionality of the buttons.   
 It takes speech as input as it makes it simple for most of the people. The input is taken by the computer using the microphone as the inbuilt microphone for computer may induce extensive sound in the inputs. Once The application gets the speech input, it processes the speech and after analysing the speech commands performs the required operation. The application analyses the speech input by first converting it into text format and the analyzing the obtained strings and finds keywords and executes commands.*

*The application is made completely using inbuilt python libraries only and no machine learning algorithm is implied in the project. All different functionalities of the application are also implemented by using inbuilt python libraries only.*

**Project Functionality**

*The Desktop application provides all the below stated functions :-*

*-* ***Opening and closing an application****, The Desktop Application can be used directly to open any application present on the desktop screen by giving a speech command through the mic by using “Open Application ” Keyword and then speaking the application name afterwards.*

*-* ***Opening a webpage****, The Desktop Application can be used open any website through the browser, for obtaining this functionality we apply various web scraping algorithms. To open any webpage we use keyword “Open Webpage” followed by the name of the site as a speech input, the desktop application detects the Keyword and opens the webpage in the new tab of the browser window.*

***- Translate input text to a desired language,*** *The Desktop Application can also be used as an Desktop version of the Google Translate as we can give speech input in form of speech and get required translated output as speech or text. The translation function can be accessed by clicking the mic button and saying “Translate”, the software will detect the keyword and run the translation module or we can* directly click on *the translation button to run the translation module. Once the translation* *module is running user can give the input for the language to which the speech is to be translated to.*

***- Provides weather details at any location of the world,*** *The Desktop Application can also be used to get the weather details of any location around the world. For accessing this function of the application, user can click on the mic icon on the application and say “Find Weather at <Location>”, or user can access this functionality by directly clicking on the weather icon and say the location name. The application provides output in form of a speech telling the user about the temperature, humidity, wind speed ect.*

***- Two player tic-tac-toe game,*** *A module for Small game of Tic-Tac-Toe has also been designed within the project that can be run clicking on the mic and giving speech command “Play Game” or we can access the game by directly clicking on the Game icon on the Application GUI screen.*

*-* ***Takes voice commands to play music,*** *The Application can be used to play music from the internet or play the music off an predefined directory. To play the music, user can click on the mic icon on the application and say “Play <song name>” if the song name exists in the directory then music plays from there otherwise the music plays on online Google Music API.*

*-* ***Takes voice commands to give information from Wikipedia,*** *The application can fetch information from Wikipedia for any particular topic or person, to access this functionality we can click on the mic icon on the desktop application and give voice command as “<Topic> Wikipedia” once this command is given to the application, the application provides output as the speech for the first five lines of the Wikipedia result of the topic.*

**Libraries Used**

*The project was mainly built using the Python programming language from scratch according to the required functionalities to attain lightning speed and smooth execution.*

*The python libraries used in the program are:*

* ***Tkinter(for GUI)***

*Python offers multiple options for developing GUI (Graphical User Interface). Out of all the GUI methods, tkinter is most commonly used method. It is a standard Python interface to the Tk GUI toolkit shipped with Python. Python with tkinter outputs the fastest and easiest way to create the GUI applications.**Importing* *tkinter is same as importing any other module in the python code. Note that the name of the module in Python 2.x is ‘Tkinter’ and in Python 3.x is ‘tkinter’.*

* ***Requests(for accessing weather API)***

*Requests is a Python module that you can use to send all kinds of HTTP requests. It is an easy-to-use library with a lot of features ranging from passing parameters in URLs to sending custom headers and SSL Verification.*

* ***Speech\_recognition(for speech to text conversion)***

*Speech\_recognition is a python module that you can use to access the google speech recognition API and you can create a object for the module to take inputs from user in speech format and the object will recognize the speech and convert it into text string.*

* ***Pyttsx3(python text to speech library)***

***pyttsx****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.However,*pyttsx *supports only Python 2.x. Hence, we will see pyttsx3 which is modified to work on both Python 2.x and Python 3.x with the same code.*

* ***Os(for accessing the operating system functions )***

*The OS module in python provides functions for interacting with the operating system. OS, comes under Python’s standard utility modules. 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.*

* ***Datetime(accessing date and current time)***

*Python datetime module is used for accessing the present date and time for the current time zone. The date time module also have some member functions that add to the functionality of the module.*

* ***Webbrowser(to access the web browser for web scraping purposes )***

*The****[webbrowser](https://docs.python.org/2/library/webbrowser.html" \l "module-webbrowser" \o "webbrowser: Easy-to-use controller for Web browsers.)****module provides a high-level interface to allow displaying Web-based documents to users. Under most circumstances, simply calling the*[***open()***](https://docs.python.org/2/library/webbrowser.html#webbrowser.open)*function from this module will do the right thing.*

* ***Translate(for translation purposes)***

*Translate is a simple but powerful translation tool written in python with with support for multiple translation providers. By now we are integrated with Microsoft Translation API and Translated MyMemory API. The biggest reason to use translate is make translations in a simple way without the need of much effort and can be used as a translation tool like command line.*

**Source Code**

1. ***File Name : Griffin.py***

import tkinter as tk

from tkinter.ttk import\*

import requests

from speaker import \*

import datetime

import os

import wikipedia

import webbrowser

import random

import weather

from translate import Translator

HEIGHT = 512

WIDTH = 435

def wish\_user():

hour=int(datetime.datetime.now().hour)

if hour>=0 and hour<12:

speak("Good Morning!")

elif hour>=12 and hour<18:

speak("Good Afternoon!")

elif hour>=18 and hour<24:

speak(" Good Evening! ")

else:

speak("Good Night!")

speak(" welcome! ")

def tr():

print("Enter the language in which you want to convert your text:")

speak("Enter the language in which you want to convert your text:")

lang=input\_speech()

print("Do you want to enter data or speak data or do you want to exit ? ")

speak("Do you want to enter data or speak data or do you want to exit ? ")

flag=input\_speech()

t=Translator(to\_lang=lang)

if flag=="enter data":

dat=input("Enter data.....")

translation=t.translate(dat)

speak(translation)

elif flag=="exit":

speak("bye")

elif flag=="speak data":

speak("speak the data to be translated")

data=''

speak("Speak Sir")

data=input\_speech()

translation=t.translate(data)

speak(translation)

else:

speak("please enter the right choise")

tr()

def play():

speak("Let us play Tic tAC toe")

os.startfile("tictac.py")

def Griffin():

query=input\_speech()

query=query.lower()

if 'wikipedia' in query:

speak("Finding information in wikipedia")

query=query.replace("wikipedia","")

results=wikipedia.summary(query,sentences=2)

speak("with refrence to wikipedia")

speak(results)

elif 'open youtube' in query:

speak("opening youtube")

webbrowser.open("youtube.com")

elif "exit" in query:

exit()

elif 'play game' in query:

speak("Let us play Tic tAC toe")

os.startfile(r"C:\Users\admin\AppData\Local\Programs\Python\Python37-32\tictac.py")

elif 'open google' in query:

speak("opening google")

webbrowser.open("google.com")

elif "translate" in query:

translate()

elif query.find('open')!=-1:

q=query.replace("open","")

q=q.strip()

print(q)

speak("opening"+q+".com")

webbrowser.open(q+".com")

elif 'bye' in query or 'end' in query:

exit()

else :

speak("you said"+query)

def know\_me():

speak("i am marion do you want to know about me??")

query=input\_speech()

if query=="yes":

speak("i was created by a team of students, and i was named after the protagonist of the famous novel the invisible man and i can perform all basic operations through your voice commands i am always happy to help")

else:

speak("ok you can know about me later")

if \_\_name\_\_=="\_\_main\_\_":

root = tk.Tk()

root.title("Assistant")

root.resizable(0,0)

canvas = tk.Canvas(root, height=HEIGHT, width=WIDTH)

canvas.pack()

background\_image = tk.PhotoImage(file='pic.png')

background\_label = tk.Label(root, image=background\_image)

background\_label.place(relwidth=1, relheight=1)

intro=tk.PhotoImage(file="intro.png")

button1 = tk.Button(root, fg="black", bg="white", borderwidth=0, font=80,image= intro, command=lambda: know\_me())

button1.place(relx=0.42, rely=0.15, relheight=0.15, relwidth=0.2)

mic=tk.PhotoImage(file="mic.png")

button2 = tk.Button(root, bg="white", fg="black", borderwidth=0, font=80, image= mic, command=lambda: atmaram())

button2.place(relx=0.166, rely=0.42, relheight=0.18, relwidth=0.22)

weather1=tk.PhotoImage(file="weather.png")

button3 = tk.Button(root, bg="white", fg="black", borderwidth=0, image=weather1, font=80, command=lambda: weather.weather())

button3.place(relx=0.39, rely=0.695, relheight=0.18, relwidth=0.25)

trans=tk.PhotoImage(file="trans.png")

button4 = tk.Button(root, bg="white", fg="black", borderwidth=0, image=trans, font=80, command=lambda: tr())

button4.place(relx=0.08, rely=0.75, relheight=0.15, relwidth=0.15)

tictac=tk.PhotoImage(file="tictac.png")

button5 = tk.Button(root, bg="white", fg="black", borderwidth=0, image=tictac, font=80, command=lambda: play())

button5.place(relx=0.763, rely=0.8, relheight=0.15, relwidth=0.15)

rem=tk.PhotoImage(file="rem.png")

button6 = tk.Button(root, bg="white", fg="black", borderwidth=0, image=rem, font=80, command=lambda: exit())

button6.place(relx=0.07, rely=0.04, relheight=0.13, relwidth=0.15)

wish\_user()

1. ***File Name : Weather.py***

import requests, json

import Griffin

# enter your api key here

api\_key = "b73770542be8e14c0c9fb12173dd6198"

def weather():

Griffin.speak("tell me the city in which you want to know the weather")

city\_name=Griffin.input\_speech()

#city\_name=input("Enter city name.....")

complete\_url="http://api.openweathermap.org/data/2.5/weather?q="+city\_name+"&APPID="+api\_key

response=requests.get(complete\_url)

x=response.json()

print(x)

if x['cod'] != '404':

y=x["main"]

w=x["wind"]

z=x["weather"]

current\_temperature = y["temp"]

current\_humidity = y["humidity"]

weather\_description = z[0]["description"]

Griffin.speak("there is "+weather\_description+" in "+city\_name)

Griffin.speak("the average temprature is "+str(int(current\_temperature-273.15))+" degree celsius")

Griffin.speak("the humidity is "+str(current\_humidity)+" per cent")

else:

Griffin.speak("City not found!!!!")

1. ***File Name:Speaker.py***

import speech\_recognition as sr

import pyttsx3

import pyaudio as p

engine=pyttsx3.init("sapi5")

voices=engine.getProperty('voices')

engine.setProperty('voice',voices[0].id)

engine.setProperty('rate',150)

def speak(q):

engine.say(q)

engine.runAndWait()

def input\_speech():

r=sr.Recognizer()

with sr.Microphone() as source:

print("Listening....")

r.pause\_threshold=1

audio=r.listen(source)

try:

print("Recognising.....")

query=r.recognize\_google(audio,language='en-in')

print("you said......"+query)

except Exception as e:

print(e)

print("Sorry Say tha Again...")

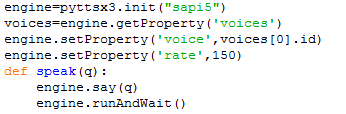
return input\_speech()

return query

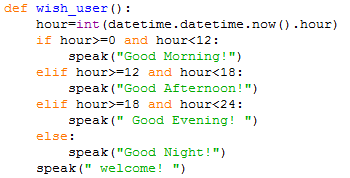
**Module Descriptions**

*The following modules are made for obtaining the required functionality of the project:*

* ***speak(query):***

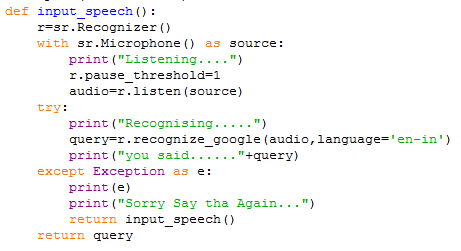
*This module uses the python library “pyttsx3” and takes a string as an input and then produces the speech equivalent of that text as an output . In the module Pyttsx3’s ‘sapi 5’ engine is used which is produced by Microsoft. the code of the module is shown below.*

* ***wish\_user():***

*this module simply wishes the user “good morning”, ”good afternoon” or “good evening” depending upon the time at which it is called. It also welcomes the user as the application is opened. It uses python predefined module datetime for knowing current time and uses speak module for greeting user. The code for this module is shown below,*

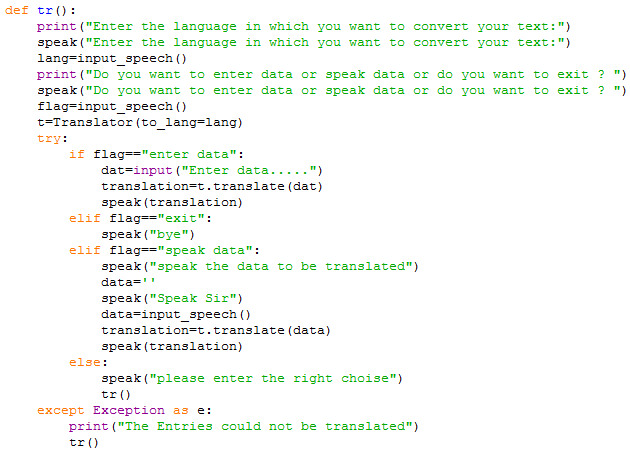
* ***input\_speech():***

*this module uses the module “speech\_recognition” of python to recognizes the speech input provided by the user and produce a string as an output. This module also utilizes the Google Speech Recognition API to obtain its functionality. The code for this module is shown below,*

**

* ***tr():***

*This module uses the “translator” module of python to take an speech input and also takes the language to translate the text to and then produces the speech output of the translated input. The code for this module is shown below,*

******

* ***mic():***

*This the module which is used when an direct speech input is given to the program by clicking on the mic icon in the GUI. It allows to do web scraping and open Wikipedia and other functions also. The programming for this module is shown below,*

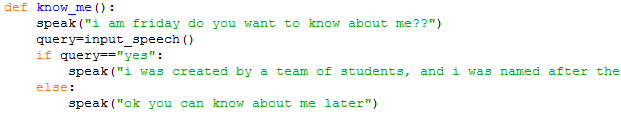
* ***Main module:***

*The main module is the first to get executed in any code of python. This module only contains the GUI programming of the module. Alsothe GUI programming of the application is done using Tkinter and all the other modules are imported into this file for obtaining the complete functionality. The coding of the module is shown below*,

**

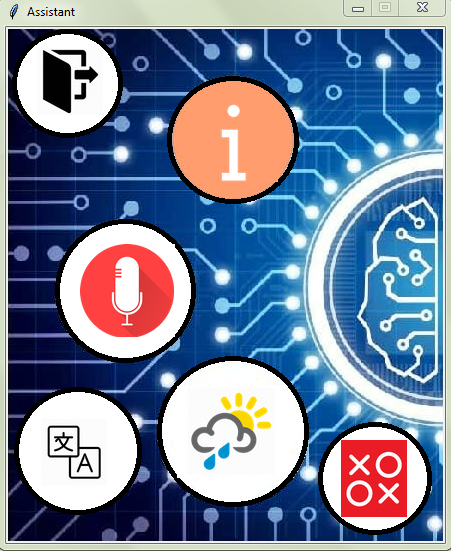
* ***know\_me():***

*This module allows the program to tell the user about itself.when this module is called by the user, the program Explains its functionality and also provides information regarding its date of creation. The code for this module is shown below,*

****

**GUI Explanation**

*The following diagram Explains the Functioning of the Project’s Output GUI.*

****

**The Information Button:** *On clicking this button user can get basic information about the Application*

**The Mic Button:** *This button allows user to give commands using various keywords.*

**Exit Button:**

*This button Exits the application*

**The Translator Button:** *This Button allows user to speak any string in English and translates that string to any given language.*

**The Game Button:** *Using this button user can start a game of two player Tic-Tac – Toe on computer.*

**The Weather Button:** *On clicking this button user can get the weather conditions of any location of world by speaking the location*

**Project Applications**

*Maryion was built to help the people with limited computer knowledge but it is also important to note that the other class of users might find some specific functionalities such as system logging useful.*

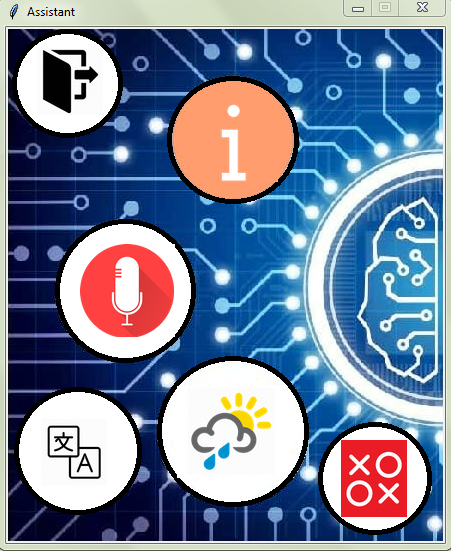
*The people who are not familiar getting around on their own on a computer can use this application as it abstracts away all the steps and presents only most important. Want to open and close an application, file? Just say it as you would to a person, it will do the work for you.*

*It can be used for playing music and knowing weather of a city without even touching the keyboard once. Applications like these can be extended to further applications.*

*for the people who are willing to make a computer as their real assistant( not like fake so called assistants like Siri, Cortana etc ), just like Tony Stark?. Even if anyone uses it, you get complete stats about what he/she was doing, what did they connected etc.*

**Output**

*The following image displays the final output screen obtained in the project “Virtual Personal Assistant”*

****

**Result**

