

APPENDIX 1

**INT 404 ASSIGNMENTS**

**Artificial Intelligence**

**END TERM REPORT**

*Topic:-***voice Based  
virtual Assistance for  
windows**

**Project Section: - K18MS**

**NAME OF THE CANDIDATE: -**

**Name: - Abhi Raj**

**Submitted to: - Jasleen kaur (25340)**

**Assistant Professor**

Department of Intelligent Systems

School of Computer Science Engineering

Lovely Professional University, Jalandhar

## APPENDIX 2

### Student Declaration

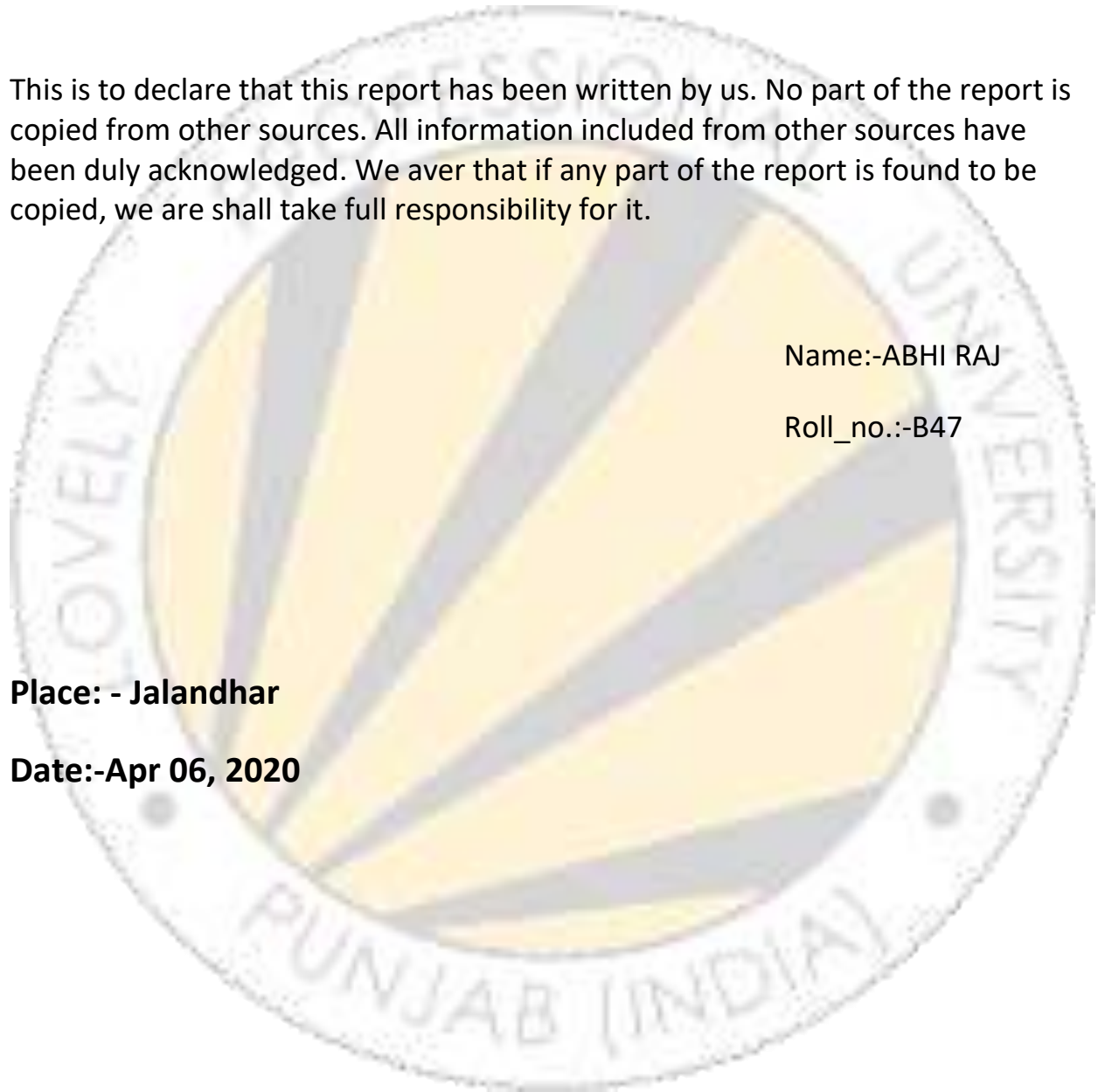
This is to declare that this report has been written by us. No part of the report is copied from other sources. All information included from other sources have been duly acknowledged. We aver that if any part of the report is found to be copied, we are shall take full responsibility for it.

Name:-ABHI RAJ

Roll\_no.:-B47

Place: - Jalandhar

Date:-Apr 06, 2020



## APPENDIX 3

### BONAFIDE CERTIFICATE

This is to certified that this “Abhi Raj” student of B.Tech (CSE) of Lovely Professional University has worked under my supervision and guidance for this project work and prepared a project Report with the title “voice based virtual assistance for windows”.

The project Report which these are submitting is his genuine and original work to Best of my knowledge.

Jasleen Kaur

Assistant Professor

25340

Department of Intelligent Systems

## TABLE OF CONTENTS

TITLE	PAGENO
1. Introduction and Objective.....	1
2 Technology Used .....	2
3.SWOT Analysis.....	(2-3)
4. codes of program.....	(4-7)

## **INTRODUCTION**

A voice assistant is a software agent that performs tasks or services assigned to it by the user through various commands. Voice assistants of the same domain are able to interpret human speech and respond to it. On the other hand, data storage is major issue in day-to-day life. Every smart device has a limited data storage capacity. After the storage is full, managing data is a tedious job. For this given issue, we make use of multiple hard drives which also have a limited capacity. Moving data from different storage devices becomes difficult and is time consuming. Once the surface of the disk is damaged by regular crashes, it leads to the loss of the data storage in the disk.

## **OBJECTIVE**

The objective of the project is to create a system which serve the purpose of voice personal assistant. Voice-controlled intelligent personal assistant is a software agent that can perform tasks or services for an individual. The system has no other advantages for the user using the agent. The integration of file storage system which can be accessed by heterogeneous client devices with voice-controlled intelligent personal assistant bot leads to a smart memory assistant device with processing capacities for the memory, this processing power for the memory adds up to the operations like user interactions, streaming, personal cloud storage.

The logo of Owens Professional University is a large, faint watermark in the background. It is a circular emblem with a yellow and grey striped design. The text "OWENS PROFESSIONAL UNIVERSITY" is written around the top half of the circle, and "INDIA" is at the bottom.

### **Technology used in this project**

- Natural Language processing
- virtual agents
- AI optimized hardware
- Decision Management
- .net framework
- python programming language
- speech recognition

### **SWOT ANALYSIS**

A SWOT analysis basically involves specifying the objective of a project or goal and identifying the internal and external factors that are favourable and unfavourable so you can overcome any obstacles that are in the way of reaching it.

It can be carried out for pretty much anything but I'm going to give you some examples further down so you can apply the analysis to different VA challenges you may encounter.

### **SWOT stands for:**

- Strengths
- Weaknesses
- Opportunities
- Threats

## **STRENGTH**

Reliability- it is one of the most important ones because sadly it seems to be the most common. A huge number of clients told me they'd been let down by flaky VAs who messed them about, didn't report back on their progress, or simply didn't deliver it at all

conversational-More natural conversation creates surprise and curiosity for a cutting-edge product.

## **WEAKNESS**

- ☐ Understand the question and provide the relevant answers is sometimes it is difficult
- ☐ Hard to adopt all skills. For this assistant it is hard to adopt the skills but very limited algorithms are integrated in this voice assistant so it can perform only limited tasks.
- ☐ No clear direction

## **Opportunities**

Users can talk freely and ask questions

## **Threats**

The domain can be unclear, users could ask the wrong questions

## Programs of Voice based virtual assistance for windows

```
import pyttsx3
import speech_recognition as sr
import datetime
import wikipedia
import webbrowser
import os
import smtplib
```

```
MASTER = "Sir"
print("Initializing Assistant")
```

```
engine = pyttsx3.init('sapi5')
voices = engine.getProperty('voices')
engine.setProperty('voice', voices[0].id)
```

```
def speak(text):
    engine.say(text)
    engine.runAndWait()
```

```
speak("Initializing Assistant")
```

```
def wishMe():
    hour = int(datetime.datetime.now().hour)
```

```
    if hour >= 0 and hour < 12:
        speak("Good Morning" + MASTER + ", how may I help you?")
```

```
    elif hour >= 12 and hour < 18:
        speak("Good Afternoon" + MASTER + ", how may I help you?")
```

```
    else:
        speak("Good Evening" + MASTER + ", how may I help you?")
```

```
def takeCommand():
```



```

r = sr.Recognizer()
with sr.Microphone() as source:
    print("Listening...")
    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("Uh-Oh, say that again please...")
    speak("Uh-Oh, say that again please...")
    query = None
return query

def sendEmail(to, content):
    server = smtplib.SMTP('smtp.gmail.com', 587)
    server.ehlo()
    server.starttls()
    server.login('your email id', 'your passwd')
    server.sendmail('reciever email id', to, content)
    server.close()

def main():
    wishMe()
    speak("I am your virtual assistant and will be happy to help!")
    query = takeCommand()

    #Logic for executing tasks
    if 'wikipedia' in query.lower():
        speak('Searching wikipedia...')
        query = query.replace("wikipedia", "")
        results = wikipedia.summary(query, sentences = 2)
        print(results)
        speak(results)

    elif 'open youtube' in query.lower():
        url = "youtube.com"

```

```
chrome_path = 'C:\Program Files (x86)\Google\Chrome\Application\chrome.exe  
%s'
```

```
webbrowser.get(chrome_path).open(url)
```

```
elif 'open google' in query.lower():
```

```
url = "google.com"
```

```
chrome_path = 'C:\Program Files (x86)\Google\Chrome\Application\chrome.exe  
%s'
```

```
webbrowser.get(chrome_path).open(url)
```

```
elif 'open Facebook' in query.lower():
```

```
url = "facebook.com"
```

```
chrome_path = 'C:\Program Files (x86)\Google\Chrome\Application\chrome.exe  
%s'
```

```
webbrowser.get(chrome_path).open(url)
```

```
elif 'open live' in query.lower():
```

```
url = "lpulive.lpu.in"
```

```
chrome_path = 'C:\Program Files (x86)\Google\Chrome\Application\chrome.exe  
%s'
```

```
webbrowser.get(chrome_path).open(url)
```

```
elif 'open university management system' in query.lower():
```

```
url = "ums.lpu.in"
```

```
chrome_path = 'C:\Program Files (x86)\Google\Chrome\Application\chrome.exe  
%s'
```

```
webbrowser.get(chrome_path).open(url)
```

```
elif 'play music' in query.lower():
```

```
songs_dir = "D:\\Music"
```

```
songs = os.listdir(songs_dir)
```

```
print(songs)
```

```
os.startfile(os.path.join(songs_dir, songs[0]))
```

```
elif 'time' in query.lower():
```

```
strTime = datetime.datetime.now().strftime("%H:%M:%S")
```

```
speak("Sir, the time is" + strTime)
```

```
elif 'open code' in query.lower():
```

```
codePath = "C:\\Users\\DELL\\Desktop\\project.py"
os.startfile(codePath)
```

```
elif 'open notepad' in query.lower():
```

```
notePath = "C:\\Users\\DELL\\Desktop\\project.txt"
os.startfile(notePath)
```

```
elif 'tell me a joke' in query.lower():
```

```
    speak("A user interface is like a joke; if you have to explain it, then it's not that good!")
```

```
elif 'open github' in query.lower():
```

```
    url = "github.com"
```

```
    chrome_path = 'C:\\Program Files (x86)\\Google\\Chrome\\Application\\chrome.exe %s'
```

```
    webbrowser.get(chrome_path).open(url)
```

```
elif 'send mail' in query.lower():
```

```
    try:
```

```
        speak("What should I say?")
```

```
        content = takeCommand()
```

```
        to = "reciever's mail id"
```

```
        sendEmail(to, content)
```

```
        speak("Email has been sent")
```

```
    except Exception as e:
```

```
        print(e)
```

```
        speak("Sorry" + MASTER + "I was unable to send the email")
```

```
main()
```





