**DESKTOP ASSISTANTS**

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

**BACHELOR OF ENGINEERING**

**IN**

**COMPUTER SCIENCE & ENGINEERING**



**Submitted to:**

**Mentor name**

**Er.Tanu Dhiman**

**Submitted By:**

**NAME**

**Divyansh singh**

**Vivek kumar singh**

**UID**

**18BCS1560-CSE-5B**

**18BCS1557-CSE-5B**

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**Chandigarh University, Gharuan**

**August 2019**

**PROJECT SYNOPSIS:**

**Topic----Desktop assistants**

**Introduction :-** Who doesn't want to have the luxury to own an assistant who always listens for your call, anticipates your every need, and takes action when necessary? That luxury is now available thanks to artificial intelligence-based voice Desktop assistants. Voice assistants come in somewhat small packages and can perform a variety of actions after hearing your command. They can turn on lights, answer questions, play music, place online orders and do all kinds of AI-based stuff.Voice assistants are not to be confused with virtual assistants, which are people who work remotely and can, therefore, handle all kinds of tasks. Rather, voice assistants are technology based. As voice assistants become more robust, their utility in both the personal and business realms will grow as well.

**What is a Voice Assistant ?**

A **voice assistant** or **intelligent personal assistant** is a software agent that can perform tasks or services for an individual based on verbal commands i.e. by interpreting 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, open or close any application etc with verbal commands.Let me give you the example of [Braina (Brain Artificial)](https://en.wikipedia.org/wiki/Braina) which is an intelligent personal assistant, human language interface, automation and **voice recognition software** for Windows PC. Braina is a multi-functional AI software that allows you to interact with your computer using **voice commands** in most of the languages of the world. Braina also allows you to accurately convert speech to text in over 100 different languages of the world.

**History of Voice Assistants :-** In recent times, Voice assistants got the major platform after Apple integrated the most astonishing Virtual Assistant — Siri which is officially a part of Apple Inc. But the timeline of greatest evolution began with the year 1962 event at the Seattle World Fair where IBM displayed a unique apparatus called Shoebox. It was the actual size of a shoebox and could perform scientific functions and can perceive 16 words and also speak them in the human recognizable voice with 0 to 9 numerical digits.

During the period of the 1970s, researchers at Carnegie Mellon University in Pittsburgh, Pennsylvania — with the considerable help of the U.S Department of Defence and its Defence Advanced Research Projects Agency (DARPA) — made Harpy. It could understand almost 1,000 words, which is approximately the vocabulary of a three-year-old child. Big organizations like Apple and IBM sooner in the 90s started to make things that utilized voice acknowledgment. In 1993, Macintosh began to building speech recognition with its Macintosh PCs with PlainTalk. In April 1997, Dragon NaturallySpeaking was the first constant dictation product which could comprehend around 100 words and transform it into readable content.

**Voice assistant the future :-** Having said that, how cool it would be to build a simple voice-based desktop/laptop assistant that has the capability to:-

1. Open the subreddit in the browser.  
2. Open any website in the browser.  
3. Send an email to your contacts.  
4. Launch any system application.  
5. Tells you the current weather and temperature of almost any city  
6. Tells you the current time.  
7. Greetings  
8. Play you a song on VLC media player(of course you need to have [VLC media player](https://www.videolan.org/vlc/index.html) installed in your laptop/desktop)  
9. Change desktop wallpaper.  
10. Tells you latest news feeds.  
11. Tells you about almost anything you ask.

So here in this article, we are going to build a voice-based application which is capable of doing all the above-mentioned tasks.

**Feasibility Study :-**

Main objective of this thesis is to show feasibility of building a personal assistant software (a smart agent) using semantic data sources available on the web, user generated content, data from the sensors of user’s desktop devices and providing knowledge from knowledge databases as well as from inference technologies of web 3.0. To design a smart agent that has contextual information about the user and helps in managing and planning tasks, using semantic web technologies and open data available on the Internet. Contextual information about the user can be location, current time, calendar appointments, relation between tasks, decomposition of tasks, past history of tasks, user interests, likes etc. Agent can use data gathered about the user as well as environment data to better understand what each of the tasks mean and decompose the tasks based on sequence of steps stored in its knowledge base and then plan individual tasks. Planning part of the agent will strive to optimize resources and try to improve productivity of the user. It can be used as a time management application as well as 4 a task management application. By combining, related tasks together that can be completed at the same time and around the same location, agent will optimize the user’s resources to complete these tasks. A feedback loop from the user will help the agent to make decisions when there are multiple paths and agent does not have sufficient information to make those decisions. Assumptions, limitations and constraints in the solution will be highlighted and any additional infrastructure necessary as a complement to the system will be identified.

Methodology/ Planning of work  :-

# Building our desktop voice assistant using python :-

Having said that, how cool it would be to build a simple voice-based desktop/laptop assistant that has the capability to:-

1. Open the subreddit in the browser.  
2. Open any website in the browser.  
3. Send an email to your contacts.  
4. Launch any system application.  
5. Tells you the current weather and temperature of almost any city  
6. Tells you the current time.  
7. Greetings  
8. Play you a song on VLC media player(of course you need to have [VLC media player](https://www.videolan.org/vlc/index.html) installed in your laptop/desktop)  
9. Change desktop wallpaper.  
10. Tells you latest news feeds.  
11. Tells you about almost anything you ask.So here in this article, we are going to build a voice-based application which is capable of doing all the above-mentioned tasks

Innovations in Project **:-**

Start by importing all the required libraries :

import speech\_recognition as sr  
import os  
import sys  
import re  
import webbrowser  
import smtplib  
import requests  
import subprocess  
from pyowm import OWM  
import youtube\_dl  
import vlc  
import urllib  
import urllib2  
import json  
from bs4 import BeautifulSoup as soup  
from urllib2 import urlopen  
import wikipedia  
import random  
from time import strftime

For our voice-assistant to perform all the above-discussed features, we have to code the logic of each of them in one method.

So our first step is to create the method which will interpret user voice response.

def myCommand():  
 r = sr.Recognizer()  
 with sr.Microphone() as source:  
 print('Say something...')  
 r.pause\_threshold = 1  
 r.adjust\_for\_ambient\_noise(source, duration=1)  
 audio = r.listen(source)  
 try:  
 command = r.recognize\_google(audio).lower()  
 print('You said: ' + command + '\n')  
 #loop back to continue to listen for commands if unrecognizable speech is received  
 except sr.UnknownValueError:  
 print('....')  
 command = myCommand();  
 return command

Next, create a method that will convert text to speech.

def sofiaResponse(audio):  
 print(audio)  
 for line in audio.splitlines():  
 os.system("say " + audio)

Now create a loop to continue executing multiple commands. Inside the method assistant() passing user command(myCommand()) as parameters.

while True:  
 assistant(myCommand())

Our next step is to create multiple if statements corresponding to each of the features. So let us see how to create these small modules inside if statement for each command.

**Software and Hardware Requirements :-**

System requirements: Python 2.7, Spyder IDE, MacOS Mojave(version 10.14)

Install all these python libraries :

pip install [SpeechRecognition](https://pypi.org/project/SpeechRecognition/)  
pip install [beautifulsoup4](https://pypi.org/project/beautifulsoup4/)  
pip install [vlc](https://pypi.org/project/beautifulsoup4/)  
pip install [youtube-dl](https://pypi.org/project/youtube_dl/)  
pip install [pyowm](https://pypi.org/project/pyowm/)  
pip install [wikipedia](https://pypi.org/project/wikipedia/)