**SONOPANT DANDEKAR SHIKSHAN MANDALI’S**

****A

Project Report On

**VIRTUAL ASSISTANT**

SUBMITTED BY

**Miss. PRIYA HIRAMAN RAJAK**

**66037**

UNDER THE GUIDANCE OF

**Prof. TEJAL PATIL.**

THIRD YEAR BACHELOR OF SCIENCE IN

COMPUTER SCIENCE

SEMESTER-V

**MUMBAI UNIVERSITY**

**2021-2022**

**SONOPANT DANDEKAR SHIKSHAN MANDALI’S**



## SONOPANT DANDEKAR ARTS, V.S. APTE COMMERCE AND M.H. MEHTA SCIENCE COLLEGE,

**PALGHAR-401404.**

**Department of Computer Science.**

**CERTIFICATE**

This is to certify that **PRIYA HIRAMAN RAJAK** is a student studying in TY.BSC.CS SEM V. She has completed project work entitled **VIRTUAL ASSISTANT** under the guidance of Faculty Member **Prof. TEJAL PATIL.** satisfactorily and has submitted to the University of Mumbai in partial fulfillment of the requirement during the academic year 2021- 2022. The matter presented in the project report has not been submitted earlier.

## College Stamp

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

## H.O.D Project Guide External



# PLAGIARISM SCAN REPORT

Words 290 Date August 06, 2021

Characters 1748 Excluded URL

|  |  |  |  |
| --- | --- | --- | --- |
| 0%  Plagiarism | 100%  Unique | 0  Plagiarized Sentences | 14  Unique  Sentences |

# Content Checked For Plagiarism

In today’s span almost all tasks are digitalized. We have Smartphone

in hands and it is nothing less than having world at your one finger. These days we aren’t even using fingers. We just give command by speaking only of the task and it is done. There exist systems where we can say Text Mom ,“I have plan to dinner with friends so don’t make dinner for me.” And the text is sent. That is how Virtual

Assistant work.

It also supports extraordinary task such as switch on the TV or Fan Or Ac ,or booking train or booking an order from restaurants ,etc.

Virtual Assistants are software programs that help you facilate your day to day tasks, such as showing weather report, playing music, creating reminders, making shopping lists, alarms etc. They can take commands via text or by voice.

For my project the wake word is KEY. We have seen so many virtual assistants, like Apple’s Siri ,Amazon’s Alexa and Microsoft’s Cortana. For this project, wake word was chosen KEY. KEY is effortless to use. Call the wake word ‘KEY’ followed by the command. And within seconds, it gets completed the task which we have given to him. Voice searches have dominated over text search. Web searches conducted via mobile devices have only just overtaken those carried out using a computer and the analysts are already predicting that 50% of searches will be via voice by 2020.Virtual assistants are turning out to be smarter than ever. We can also allow our intelligent assistant to make email work for us. This project was started on the basis of that there is sufficient amount of openly available data and information on the web that can be utilized to build a virtual assistant .

# INDEX

|  |  |  |
| --- | --- | --- |
| **SR.NO** | **CONTENTS** | **PAGE.NO** |
| 1 | Acknowledgment | 1 |
| 2 | Requirement | 2 |
| 3 | Introduction | 3 |
| 4 | Installation of python idle 3.9 | 4-8 |
| 5 | Objectives | 9 |
| 6 | System Design | 10 |
| 7 | Coding | 11-20 |
| 8 | Result | 21-25 |
| 9 | Advantages | 26 |
| 10 | Conclusion | 27 |
| 11 | Future Scope | 28 |
| 12 | Reference | 29 |

**ACKNOWLEDGEMENT**

### The gratification and joy that accompanies the successful completion of any task would be incomplete without the humble and deep-felt expression of gratitude to everyone who has made it possible.

I would like to extend my heartfelt gratitude to **RASHMI VARADE** who gave me an opportunity to work under her guidance in this department.

Further would like to place on record my grateful thanks to friends and family who spent their valuable time in guiding me in this project, whose cooperation and encouragement, has immensely helped in the completion of the project.

I also would like to convey my sincere thanks to **Prof TEJAL PATIL** my internal project guide, who has been a constant source of motivation and inspiration and has helped in bring out the best in me. I am also thankful to all staff members of computer science department who as a team have contributed in the successful completion of this project.

Place: **Tarapur**

Date:**16 October 2021**

**Priya Hiraman Rajak**

**REQUIREMENT**

* Hardware Requirement:
* Laptop(4GB RAM)
* i3/i5 Processer
* 500 GB/1TB hard disk
* Internet Connection (Wi-Fi).
* Software Requirement:
* Python IDLE 3.9 (64 Bits)
* SQLite

**INTRODUCTION**

In today’s span almost all tasks are digitalized. We have Smartphone in hands and it is nothing less than having world at your one finger. These days we aren’t even using fingers. We just give command by speaking only of the task and it is done. There exist systems where we can say Text Mom ,“I have plan to dinner with friends so don’t make dinner for me.” And the text is sent. That is how Virtual Assistant work. It also supports extraordinary task such as switch on the TV or Fan Or Ac ,or booking train or booking an order from restaurants ,etc.

Virtual Assistants are software programs that help you facilate your day to day tasks, such as showing weather report, playing music, creating reminders, making shopping lists, alarms etc. They can take commands via text or by voice. For my project the wake word is KEY. We have seen so many virtual assistants, like Apple’s Siri ,Amazon’s Alexa and Microsoft’s Cortana. For this project, wake word was chosen KEY. KEY is effortless to use. Call the wake word ‘KEY’ followed by the command. And within seconds, it gets completed the task which we have given to him. Voice searches have dominated over text search. Web searches conducted via mobile devices have only just overtaken those carried out using a computer and the analysts are already predicting that 50% of searches will be via voice by 2020.Virtual assistants are turning out to be smarter than ever. We can also allow our intelligent assistant to make email work for us. This project was started on the basis of that there is sufficient amount of openly available data and information on the web that can be utilized to build a virtual assistant.

# INSTALLATION OF THE PYTHON IDLE 3.9

### Installation of python involves following steps:

1. Download binaries
2. Run the executable installer
3. Add path of python to environmental variable

**Step 1: Download python installer binaries**

1. Open official python website in web browser. Navigate the downloads tab for windows.
2. Choose the latest version of python 3.9.
3. Click on link to download windows x8-64 executable installer. For 64-bit system.

**Step 2: Run the Executable Installer**

1. Once the installer is downloaded, run the python installer.
2. Check the install launcher for all users check box. Further check the Add python 3.7 to path check box to include the interpreter in the execution path.
3. Select customize installation.

Choose the optional features by checking the following check boxes:

1. Documentation
2. Pip
3. Tcl/tk and IDLE (to install tkinter and IDLE)
4. Python test suite (to install standard library suite of python)
5. Install the global launcher for „.py‟ files. This makes it easier to start python
6. Install for all users.
7. This takes you to Advanced options available while installing python. Here, select the install for all users and Add python to environment variables check boxes.
8. Optionally, we can select the Associate files with python, create shortcuts for installed applications and other advanced options.

After selecting the advanced options ,clickon install to start installation.

1. once the installation is over , we will see a python setup successful window.

**Step 3: Add python to environmental variables**

The last step in installation process is to add python path to the system environment variables. This step is done to access python through the command line. In case we added python to environment variables while setting the advance options during the installation procedure, you can avoid this step. Else, this step is done manually as follows.

In start menu, search for “advance system setting”. Select “view advance system settings”. In the “system properties” window, click on the “advanced” tab and then click on the “Environment variables” button.

Locate the python installation directory on system. If we follow the above steps, python will be installed in below locations:

>C:\Program Files (x86)\python 37-32: for 32-bit installation

>C:\Program Files\python 37-32: for 64-bit installation

The folder name may be different from “python37-32” if we installed a different version. Look for a folder whose name starts with python.

**Step 4: Verify the python Installation**

You have successfully installed python 3.9 on windows 10. You can verify if the python installation is successful either through the command line or through the IDLE app that gets installed along with the installation.

Search for the command prompt and type “python”. You can see that python 3.9 is successfully installed.

An alternative way to search for “python” is the start menu and clicking on IDLE (python 3.9 64-bit). You can start coding in python using the integrated development environment (IDLE).

Now, we are ready to start developing python applications in your

windows 10 system.

* **Pyttsx**

Pyttsx stands for Python Text to Speech. It is a cross-platform Python wrapper for text-to-speech synthesis.

It is a Python package supporting common text-to-speech engines on MacOS X, Windows, and Linux. It works for both Python2.x and 3.x versions. Its main advantage is that it works offline.

* **Speech Recognition**

This is a library for performing speech recognition, with support for several engines and APIs, online and offline.

It supports APIs like Google Cloud Speech API, IBM Speech to Text, Microsoft Bing Voice Recognition etc.

* **SQLite**

SQLite is a capable library, providing an in-process relational database for efficient storage of small-to-medium-sized data sets. It supports most of the common features of SQL with few exceptions. Best of all, most Python users do not need to install anything to get started working with SQLite, as the standard library in most distributions ships with the sqlite3module.SQLite runs embedded in memory alongside your application, allowing you to easily extend SQLite with your own Python code. SQLite provides quite a few hooks, a reasonable subset of which are implemented by the standard library database driver

**OBJECTIVES**

* A computer is a device that helps everyone to achieve things fast.
* It does things faster than humans and it did not get bored whenever it performs a repetitive task.
* We can automate our repetitive tasks using a computer so tasks can perform fast and we just need to convey to a computer either by giving a voice command or by typing the command.
* You can interact with your laptop’s microphone/console. The response generated by the assistant will display on the console or as a speech via the speaker.
* Weather forecasting, Launch Games, Launch Windows Applications, Open Websites, tells you about almost everything you ask, tells you date and time, greetings, news, etc.

**System Design**

 Voice Input

Speech recognition

**Architecture**

API call

Content extraction

System call

Text to speech module



Python Backend

# CODING

import speech\_recognition as sr # recognise speech

import playsound # to play an audio files

from gtts import gTTS # google text to speech

import random

from time import ctime # get time details

import webbrowser # open browser

import ssl

import certifi

import time

import os # to remove created audio files

from PIL import Image

import subprocess

import pyautogui #screenshot

import pyttsx3

import bs4 as bs

import urllib.request

import sqlite3

class person:

name = ''

def setName(self, name):

self.name = name

class asis:

name = ''

def setName(self, name):

self.name = name

def there\_exists(terms):

for term in terms:

if term in voice\_data:

return True

def engine\_speak(text):

text = str(text)

engine.say(text)

engine.runAndWait()

r = sr.Recognizer() # initialise a recogniser

# listen for audio and convert it to text:

def record\_audio(ask=""):

with sr.Microphone() as source: # microphone as source

if ask:

engine\_speak(ask)

audio = r.listen(source, 5, 5) # listen for the audio via source

print("Done Listening")

voice\_data = ''

try:

voice\_data = r.recognize\_google(audio) # convert audio to text

except sr.UnknownValueError: # error: recognizer does not understand

engine\_speak('I did not get that')

except sr.RequestError:

engine\_speak('Sorry, the service is down') # error: recognizer is not connected

print(">>", voice\_data.lower()) # print what user said

return voice\_data.lower()

# get string and make a audio file to be played

def engine\_speak(audio\_string):

audio\_string = str(audio\_string)

tts = gTTS(text=audio\_string, lang='en') # text to speech(voice)

r = random.randint(1,20000000)

audio\_file = 'audio' + str(r) + '.mp3'

tts.save(audio\_file) # save as mp3

playsound.playsound(audio\_file) # play the audio file

print(asis\_obj.name + ":", audio\_string) # print what app said

os.remove(audio\_file) # remove audio file

def respond(voice\_data):

**# 1: greeting**

if there\_exists(['hey','hi','hello']):

greetings = ["hey, how can I help you" + person\_obj.name, "hey, what's up?" + person\_obj.name, "I'm listening" + person\_obj.name, "how can I help you?" + person\_obj.name, "hello" + person\_obj.name]

greet = greetings[random.randint(0,len(greetings)-1)]

engine\_speak(greet)

**# 2: name**

if there\_exists(["what is your name","what's your name","tell me your name"]):

if person\_obj.name:

engine\_speak("whats with my name ")

else:

engine\_speak("my name is key. what's your name?")

if there\_exists(["my name is"]):

person\_name = voice\_data.split("is")[-1].strip()

engine\_speak("okay, i will remember that " + person\_name)

person\_obj.setName(person\_name) # remember name in person object

if there\_exists(["your name should be"]):

asis\_name = voice\_data.split("be")[-1].strip()

engine\_speak("okay, i will remember that my name is " + asis\_name)

asis\_obj.setName(asis\_name) # remember name in asis object

**# 3: greeting**

if there\_exists(["how are you","how are you doing"]):

engine\_speak("I'm very well, thanks for asking " + person\_obj.name)

**# 4: time**

if there\_exists(["what's the time","tell me the time","what time is it"]):

time = ctime().split(" ")[3].split(":")[0:2]

if time[0] == "00":

hours = '12'

else:

hours = time[0]

minutes = time[1]

time = f'{hours} {minutes}'

engine\_speak(time)

**# 5: search google**

if there\_exists(["search for"]) and 'youtube' not in voice\_data:

search\_term = voice\_data.split("for")[-1]

url = "https://google.com/search?q=" + search\_term

webbrowser.get().open(url)

engine\_speak("Here is what I found for" + search\_term + "on google")

**# 6: search youtube**

if there\_exists(["youtube"]):

search\_term = voice\_data.split("for")[-1]

url = "https://www.youtube.com/results?search\_query=" + search\_term

webbrowser.get().open(url)

engine\_speak("Here is what I found for " + search\_term + "on youtube")

**# 7: search for music**

if there\_exists(["play music"]):

search\_term= voice\_data.split("for")[-1]

url="https://open.spotify.com/search/"+search\_term

webbrowser.get().open(url)

engine\_speak("You are listening to"+ search\_term +"enjoy sir")

**# 8: search for amazon.com**

if there\_exists(["amazon.com"]):

search\_term = voice\_data.split("for")[-1]

url="http://www.amazon.in"+search\_term

webbrowser.get().open(url)

engine\_speak("here is what i found for"+search\_term + "on amazon.com")

**# 9: make a note**

if there\_exists(["make a note"]):

search\_term=voice\_data.split("for")[-1]

url="https://keep.google.com/u/0/"

webbrowser.get().open(url)

engine\_speak("Here you can make notes")

**#10: weather**

if there\_exists(["weather","tell me the weather report","whats the condition outside"]):

search\_term = voice\_data.split("for")[-1]

url="https://www.google.com/search?sxsrf=ACYBGNSQwMLDByBwdVFIUCbQqya-ET7AAA%3A1578847393212&ei=oUwbXtbXDN-C4-EP-5u82AE&q=weather&oq=weather&gs\_l=psy-ab.3..35i39i285i70i256j0i67l4j0i131i67j0i131j0i67l2j0.1630.4591..5475...1.2..2.322.1659.9j5j0j1......0....1..gws-wiz.....10..0i71j35i39j35i362i39.\_5eSPD47bv8&ved=0ahUKEwiWrJvwwP7mAhVfwTgGHfsNDxsQ4dUDCAs&uact=5"

webbrowser.get().open(url)

engine\_speak("Here is what I found for on google")

**#11:open gmail**

if there\_exists(["open my mail","gmail","check my email"]):

search\_term = voice\_data.split("for")[-1]

url="https://mail.google.com/mail/u/0/#inbox"

webbrowser.get().open(url)

engine\_speak("here you can check your gmail")

if there\_exists(["exit", "quit", "goodbye"]):

engine\_speak("we could continue more priya, but.,,...,,,,,..,,,,, byee")

exit()

time.sleep(1)

person\_obj = person()

asis\_obj = asis()

asis\_obj.name = 'key'

engine = pyttsx3.init()

while(1):

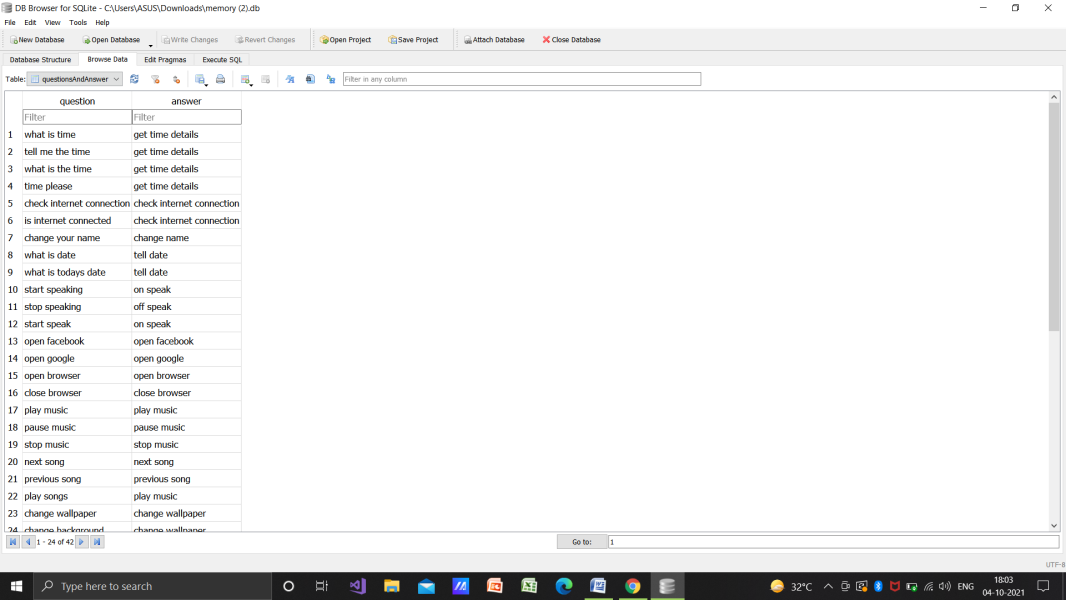
voice\_data = record\_audio("Recording") # get the voice input

print("Done")

print("Q:", voice\_data)

respond(voice\_data) # respond

* **SQLITE DATABASE**



# RESULT

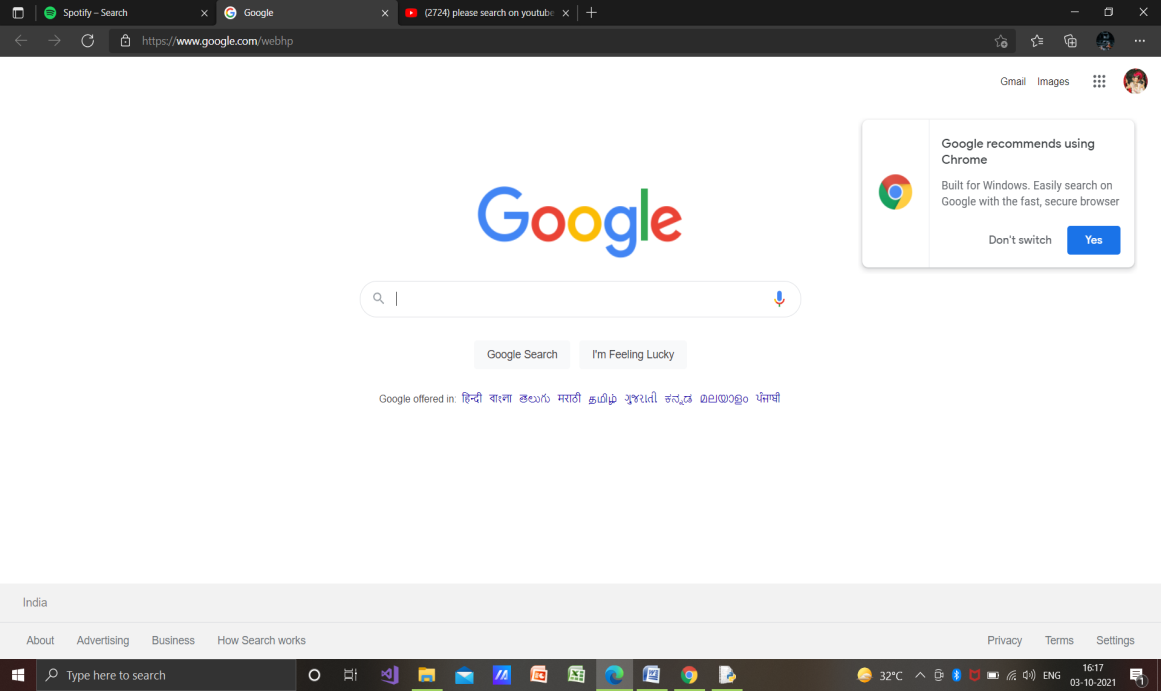
# Screenshot (731).png

# Screenshot (737).pngScreenshot (736).png

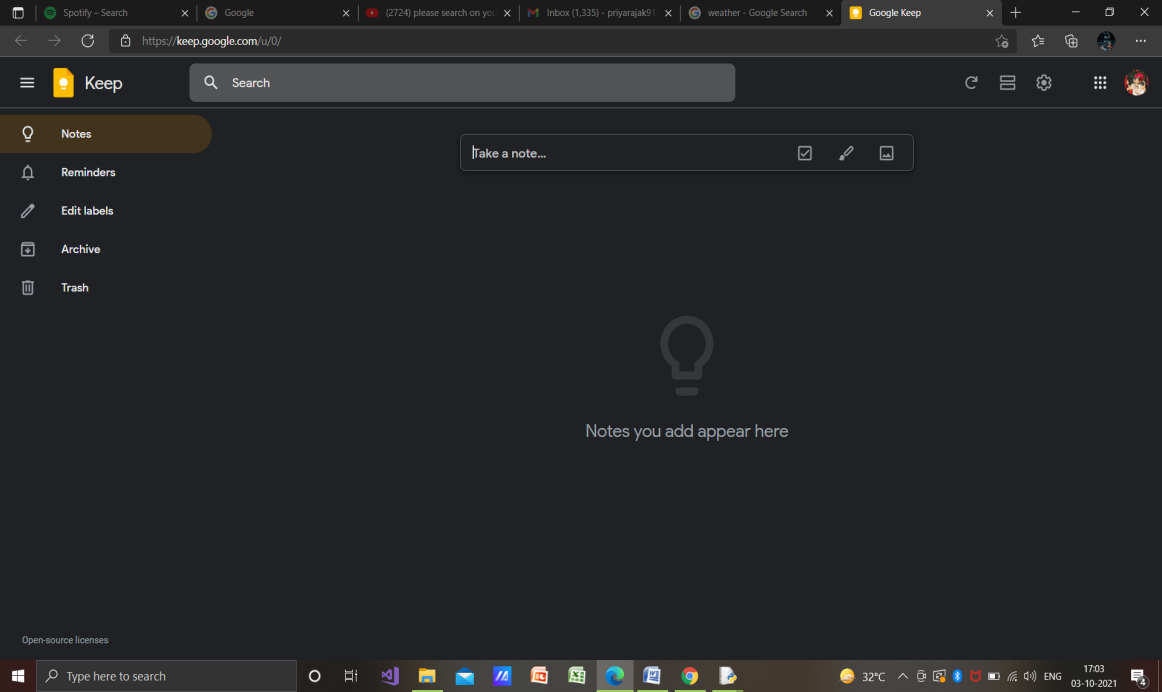
# Screenshot (738).png

# Screenshot (752).png

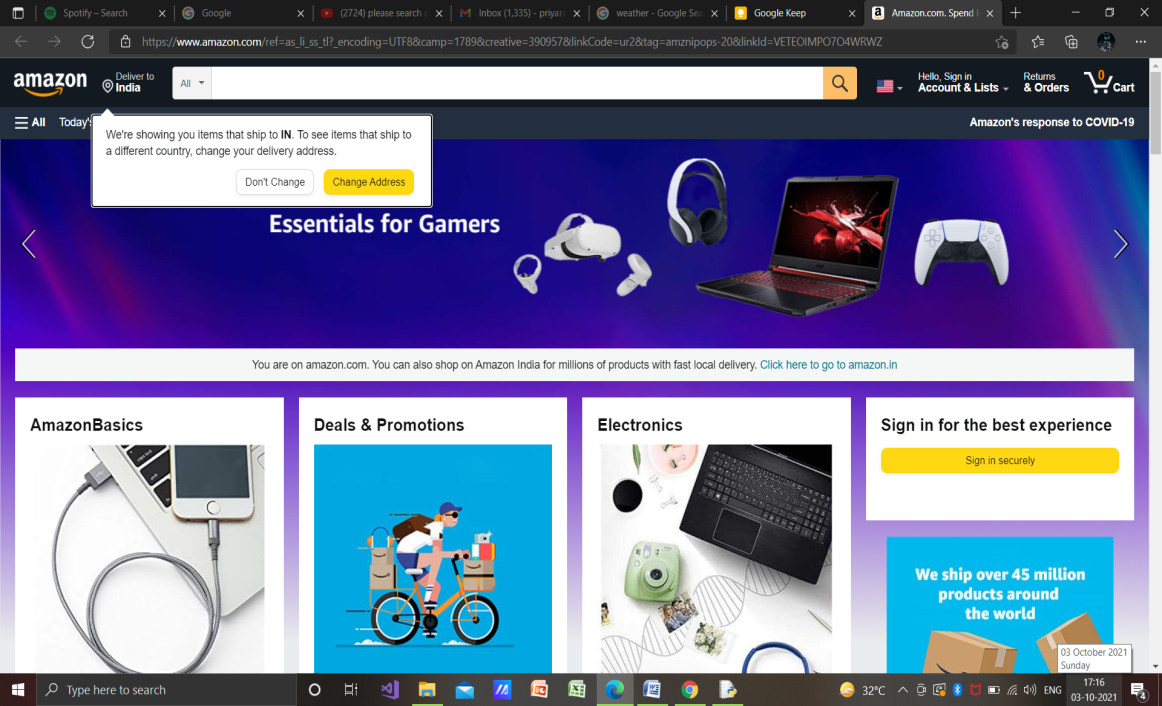
** Key search for…..**

****

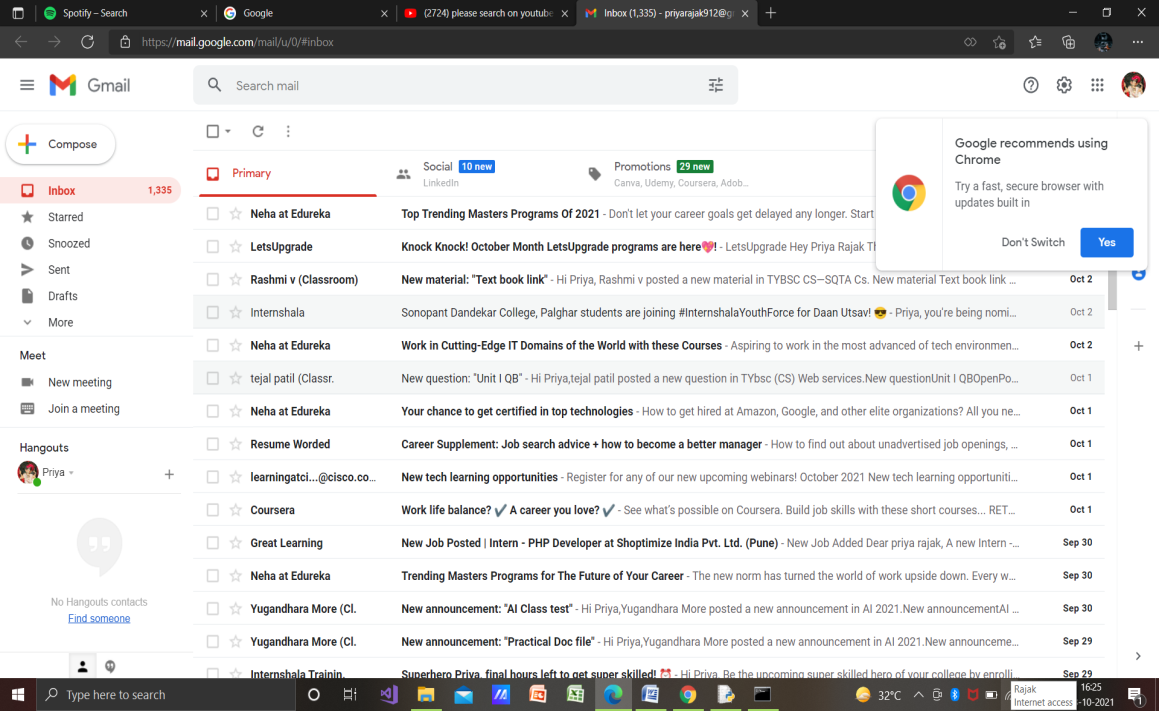
** Make a note**

****

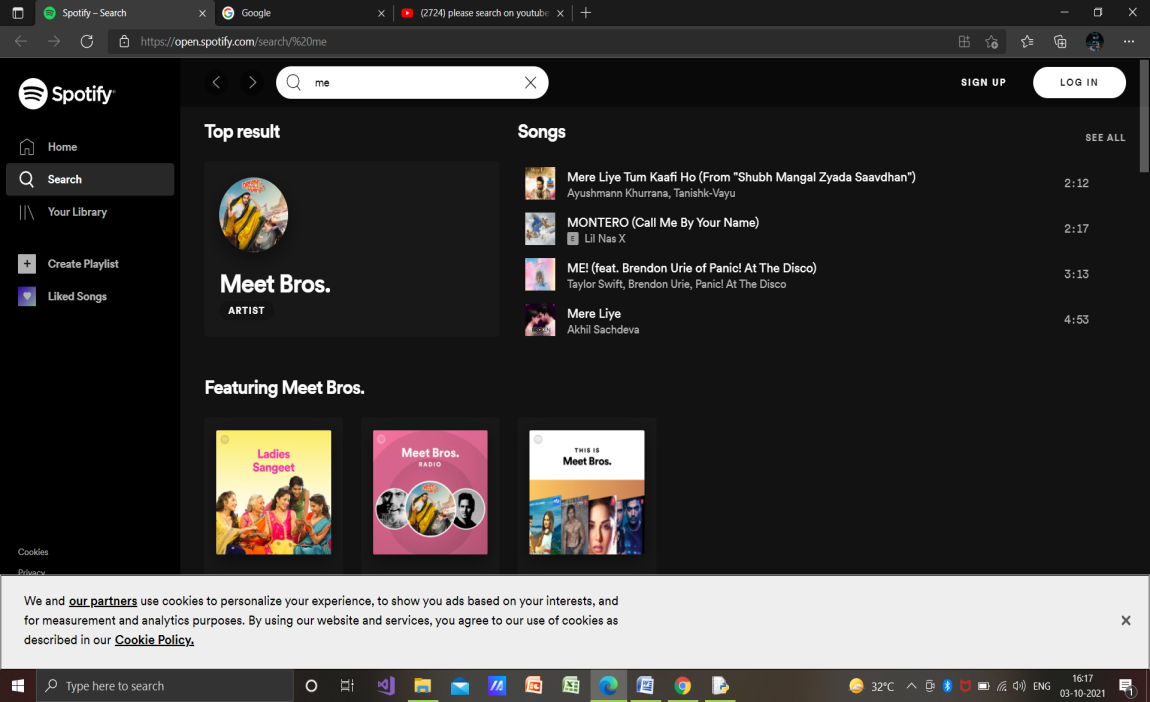
** amzon.com**

****

** check my mail**

****

** play song for me**

****

** weather report**

****

**ADVANTAGES**

* A Virtual Assistant makes your life easier.
* A Virtual Assistant saves you time and money.
* A Virtual Assistant gives you more free time for your personal life.
* A Virtual Assistant can be an expert in the field.
* A Virtual Assistant allows you to provide more coverage.

**CONCLUSION**

* The personal voice assistant system presented in this paper is very fundamental system with few features however the additional and advance feature may be introduced as future work of this project , In this paper the design and implementation of a Intelligent Personal Voice Assistance is described.
* The project is built using available open source software modules with visual studio code community backing which can accommodate any updates in future.
* The modular approach used in this project makes it more flexible and easy to integrate additional modules and features without disturbing the current system functionaries.
* It not only works on human commands but also it is designed for give responses to the user on the basis of query being asked or the words spoken by the user such as opening tasks and operations.

**FUTURE SCOPE**

* Voice assistants will continue to offer more individualized experiences as they get better at differentiating between voices.
* However, it’s not just developers that need to address the complexity of developing for voice as brands also need to understand the capabilities of each device and integration and if it makes sense for their specific brand.
* They will also need to focus on maintaining a user experience that is consistent within the coming years as complexity becomes more of a concern.
* This is because the visual interface with voice assistants is missing.
* Users simply cannot see or touch a voice interface
* This Intelligent Voice Assistant has an enormous and limitless scope in the future. Like Siri, Google Now and Cortana most popular personal voice assistants.
* The project will easily able to integrate with devices near future for a Connected Home using Internet of Things, voice command system and computer vision

**REFERENCE**

* [https://www.youtube.com/watch?v=5Ke4fJjRaIo&list=PLjC8JXsSUrrg6Plc3khOW6MI\_O7DoxnnF&index=3](%09https://www.youtube.com/watch?v=5Ke4fJjRaIo&list=PLjC8JXsSUrrg6Plc3khOW6MI_O7DoxnnF&index=3)
* <https://www.geeksforgeeks.org/voice-assistant-using-python/>