

BCA SEMESTER – 6

PROJECT TYPE : Desktop Application



G ASSISTANT - A PERSONAL VOICE ASSISTENT

DEVELOPED BY : **GHANSHYAM VAJA**

SPU ID : **2020022333**

Submitted From : Savjani College-Veraval To : Bhakta Kavi Nasinh Mehta University



Submission Year : **2023**

CERTIFICATE of ORIGINALITY

This is to certify that this Project Report entitled "**G Assistant – A Personal Voice Assistant**" submitted to **Bhakta Kavi Narsinh Mehta University** for **Bachelor in computer application - VI(BCA - 6)**, is an original work carried out by **Mr. GHANSHYAM VAJA**.

SPU ID : 2020022333 under the guidance of **PATHAK BHAVIK sir** matter embodie.

this project is a genuine work done by the student and has not been submitted whether to this University or any other university/Institute for the fulfillment of any course of study.

**DR. JIGAR RAVAL Sir
(Principal)**

College Seal

**PATHAK BHAVIK sir
(Guide)**

ACKNOWLEDGEMENT

The satisfaction that accompanies that the successful completion of any task would be incomplete without the mention of people whose ceaseless cooperation made it possible, whose constant guidance and encouragement crown all efforts with success.

I am grateful to my **PARENTS**, **project guide : PATHAK BHAVIK Sir** and **Principal : Dr. JIGAR RAVAL sir** for the guidance, inspiration and constructive suggestions that helped me in the preparation of this project.

I am also thankful to **my colleagues** who have helped me in successful completion of the project.

GHANSHYAM VAJA
(Student)

Project Specification

Project Title	G ASSISTENT - A Personal Voice Assistant
Project Category	Desktop Application
GUI	Tkinter
Written in	Python 3.10.8
Developed for	Windows OS 7 and above
Operating System and Processor(In local machine)	OS : Windows 11 Processor : Intel® Core i5 11Gen
Documentation Preparing Tool	Microsoft Word 2021
Developed by	GHANSHYAM VAJA
Project Guide	PATHAK BHAVIK Sir
Submitted From	K.M. & K.K. Savjani College - Veraval
Submitted to	Bhakta Kavi Narsinh Mehta University - Junagadh

INDEX

1. About Project	5
2. Software Development Life Cycle (SDLC)	9
3. Feasibility Study	12
4. Python Modules	14
5. Database Tables	20
6. ER Diagram	21
7. Data Flow Diagram	21
8. Source Code(main.py)	22
9. Spec file (main.spec)	58
10. Executable File (main.exe)	60
11. Supported Commands	61
12. Application Screenshots	63
13. Limitations	72
14. Future Enhancements	73
15. Bibliography	74
16. Reference	75

1. About Project

What is voice Assistant?

A **voice assistant** is a **software agent** that can interpret human speech and respond via synthesized voices. It communicates with the users in natural language.

A voice assistant, also called **an intelligent personal assistant** or a connected speaker, are new types of products marketed by Apple, Amazon and Google and are based on natural language speech recognition. **They allow a search** to be carried out using a **voice command** entered by the user, as well as **information retrieval by voice synthesis**.

Voice assistants use technologies like **voice recognition**, **speech synthesis**, and **NLP(Natural Language Processing)** to provide services to the users. Voice recognition is the heart of a voice application and is a rapidly evolving technology that provides an alternative to keyboard typing. Voice recognition is an important component for the user as a gateway to use his or her voice as an input component.

Used Technologies in this Project:

- **Python 3.10.8->** Used to Write Logic for G Assistant Desktop Application
- **Tkinter ->**Used to Create GUI for G Assistant Desktop Application
- **Database :** MySQL 8.0 (**Database Provider :** db4free.net)

The name of My Project is **G Assistant – A Personal Voiice Assistant**. This project is made by me **for project submission of BCA-6**. This is made in **python**. In this, **Tkinter** is used for GUI. **MySQL** is used for Store Data in Database.

PYTHON 3.10.8 :



Python is a popular programming language. It was created by **Guido van Rossum**, and released in **1991**.

It is used for:

- web development (server-side),
- software development,
- mathematics,
- system scripting.
- AI(Artificial Intelligence)
- ML(Machine Learning)
- Data Science

What can Python do?

- Python can be used on a server to create web applications.
- Python can be used alongside software to create workflows.
- Python can connect to database systems. It can also read and modify files.
- Python can be used to handle big data and perform complex mathematics.
- Python can be used for rapid prototyping, or for production-ready software development.

Why Python?

- Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).
- Python has a simple syntax similar to the English language.
- Python has syntax that allows developers to write programs with fewer lines than some other programming languages.
- Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.
- Python can be treated in a procedural way, an object-oriented way or a functional way.

Good to know

- The most recent major version of Python is Python 3, which we shall be using in this tutorial. However, Python 2, although not being updated with anything other than security updates, is still quite popular.
- In this tutorial Python will be written in a text editor. It is possible to write Python in an Integrated Development Environment, such as Thonny, Pycharm, Netbeans or Eclipse which are particularly useful when managing larger collections of Python files.

Python Syntax compared to other programming languages

- Python was designed for readability, and has some similarities to the English language with influence from mathematics.
- Python uses new lines to complete a command, as opposed to other programming languages which often use semicolons or parentheses.
- Python relies on indentation, using whitespace, to define scope; such as the scope of loops, functions and classes. Other programming languages often use curly-brackets for this purpose.

Tkinter :

Python offers multiple options for developing GUI (Graphical User Interface). Out of all the GUI methods, tkinter is the most commonly used method. It is a standard Python interface to the Tk GUI toolkit shipped with Python. Python with tkinter is the fastest and easiest way to create the GUI applications. Creating a GUI using tkinter is an easy task.

To create a tkinter app:

1. Importing the module – tkinter
2. Create the main window (container)
3. Add any number of widgets to the main window
4. Apply the event Trigger on the widgets.

MySQL 8.0 (Database Provider : db4free.net):



- MySQL is a **relational database management system**
- MySQL is open-source
- MySQL is **free**
- MySQL is ideal for both small and large applications
- MySQL is very fast, reliable, scalable, and easy to use
- MySQL is cross-platform
- MySQL is compliant with the ANSI SQL standard
- MySQL was **first released in 1995**
- MySQL is developed, distributed, and supported by Oracle Corporation
- MySQL is named after **co-founder Monty Widenius's daughter: My**

Functionalities :

From this project, computer user(windows user) can do

- Web browsing,
- Open whatsapp,
- Open Instagram,
- Open Google,
- question and answers,
- Youtube Video watching and downloading,
- Wikipedia Search,
- Game Playing (Guess the Number Game, Snake and Ladder Game),
- find Location of any Place,
- Listening Jokes,
- Restart Computer,
- Open Command Prompt,
- Shut Down Computer,
- Take Photo,
- Get Date and Time etc.....

2. Software Development Life Cycle (SDLC)

A software life cycle model (also termed process model) is a pictorial and diagrammatic representation of the software life cycle. A life cycle model represents all the methods required to make a software product transit through its life cycle stages. It also captures the structure in which these methods are to be undertaken.

A software life cycle model describes entry and exit criteria for each phase. A phase can begin only if its stage-entry criteria have been fulfilled. So without a software life cycle model, the entry and exit criteria for a stage cannot be recognized. Without software life cycle models, it becomes tough for software project managers to monitor the progress of the project.

Software Development is the development of software for distinct purposes. For software development, there is a specific programming language like Java, Python, C/C++, etc. The entire process of software development isn't as simple as its definition, it's a complicated process. Accordingly, it requires an efficient approach from the developer in the form of the Software Development Life Cycle (SDLC).



Stages of SDLC :

1. Planning and requirement analysis
2. Defining Requirements
3. Designing the Software
4. Developing the project
5. Testing

6. Deployment
7. Maintenance

1. Planning and requirement analysis :

Requirement Analysis is the most important and necessary stage in SDLC.

The senior members of the team perform it with inputs from all the stakeholders and domain experts or SMEs in the industry.

Planning for the quality assurance requirements and identifications of the risks associated with the projects is also done at this stage.

Business analyst and Project organizer set up a meeting with the client to gather all the data like what the customer wants to build, who will be the end user, what is the objective of the product. Before creating a product, a core understanding or knowledge of the product is very necessary.

2. Defining Requirements :

Once the requirement analysis is done, the next stage is to certainly represent and document the software requirements and get them accepted from the project stakeholders.

This is accomplished through "SRS"- Software Requirement Specification document which contains all the product requirements to be constructed and developed during the project life cycle.

3. Designing the Software :

The next phase is about to bring down all the knowledge of requirements, analysis, and design of the software project. This phase is the product of the last two, like inputs from the customer and requirement gathering.

4. Developing the project :

In this phase of SDLC, the actual development begins, and the programming is built. The implementation of design begins concerning writing code. Developers have to follow the coding guidelines described by their management and programming tools like compilers, interpreters, debuggers, etc. are used to develop and implement the code.

5. Testing :

After the code is generated, it is tested against the requirements to make sure that the products are solving the needs addressed and gathered during the requirements stage.

During this stage, unit testing, integration testing, system testing, acceptance testing are done.

6. Deployment :

Once the software is certified, and no bugs or errors are stated, then it is deployed.

Then based on the assessment, the software may be released as it is or with suggested enhancement in the object segment.

After the software is deployed, then its maintenance begins.

7. Maintenance : Once when the client starts using the developed systems, then the real issues come up and requirements to be solved from time to time.

This procedure where the care is taken for the developed product is known as maintenance.

3. Feasibility Study

- **What is Feasibility Study?**

As the name implies, a feasibility analysis is used to determine the viability of an idea, such as ensuring a project is legally and technically feasible as well as economically justifiable. It tells us whether a project is worth the investment—in some cases, a project may not be doable. There can be many reasons for this, including requiring too many resources, which not only prevents those resources from performing other tasks but also may cost more than an organization would earn back by taking on a project that isn't profitable.

- **Understanding of Feasibility Study :**

Project management is the process of planning, organizing, and managing resources to bring about the successful completion of specific project goals and objectives. A feasibility study is a preliminary exploration of a proposed project or undertaking to determine its merits and viability. A feasibility study aims to provide an independent assessment that examines all aspects of a proposed project, including technical, economic, financial, legal, and environmental considerations. This information then helps decision-makers determine whether or not to proceed with the project.

The feasibility study results can also be used to create a realistic project plan and budget.

Without a feasibility study, it cannot be easy to know whether or not a proposed project is worth pursuing.

- **Types of Feasibility Study**

- **Technical Feasibility :** This assessment focuses on the technical resources available to the organization. It helps organizations determine whether the technical resources meet capacity and whether the technical team is capable of converting the ideas into working systems. Technical feasibility also involves the evaluation of the hardware, software, and other technical requirements of the proposed system. As an exaggerated example, an organization wouldn't want to try to put Star Trek's transporters in their building—currently, this project is not technically feasible.
- **Economic Feasibility :** This assessment typically involves a cost/ benefits analysis of the project, helping organizations determine the viability, cost, and benefits associated with a project before financial resources are allocated. It also serves as an independent project assessment and enhances project credibility—helping decision-makers determine the positive economic benefits to the organization that the proposed project will provide.
- **Legal Feasibility :** This assessment investigates whether any aspect of the proposed project conflicts with legal requirements like zoning laws, data protection acts or social media laws. Let's say an organization wants to construct a new office building in a specific location. A feasibility study might reveal the organization's ideal location

isn't zoned for that type of business. That organization has just saved considerable time and effort by learning that their project was not feasible right from the beginning.

- **Operational Feasibility :** This assessment involves undertaking a study to analyze and determine whether—and how well—the organization's needs can be met by completing the project. Operational feasibility studies also examine how a project plan satisfies the requirements identified in the requirements analysis phase of system development.
- **Prepare an Opening Day Balance Sheet :** The opening day balance sheet is a snapshot of the company's financial position at the beginning of the business venture. The purpose of the opening day balance sheet is to give an idea of the amount of money that the company has to work with and track its expenses and income as they occur. This information is vital to making sound business decisions. The opening day balance sheet will include the following:
 - Cash on hand
 - Accounts receivable
 - Inventory
 - Prepaid expenses
 - Fixed assets
 - Accounts payable
 - Notes payable
 - Long-term liabilities
 - Share
- **Review and Analyze All Data :** The feasibility study should include reviewing and analyzing all data relevant to the proposed project. The data collected should be verified against source documentation, and any discrepancies should be noted. The purpose of the feasibility study is to provide a basis for making a decision, and the data should be sufficient to support that decision.
The analysis should consider both the positive and negative aspects of the proposed project. The financial analysis should be thorough, and all assumptions should be documented. The risk assessment should identify any potential risks and mitigation strategies. The team assigned to the project should review the feasibility study and recommend the organization's leadership.
Organizational leadership should decide whether to proceed with the project based on the feasibility study's findings. If the project is approved, the organization should develop a project plan that includes a detailed budget and timeline.

4. Python Used Modules

A **Python module** is a file containing Python definitions and statements. A module can define functions, classes, and variables. A module can also include runnable code. Grouping related code into a module makes the code easier to understand and use. It also makes the code logically organized.

- | | | |
|------------------------|-----------------------|------------------------------|
| 1. os | 2. socket | 3. speech_recognition |
| 4. pytsxs3 | 5. pyaudio | 6. tkinter |
| 7. subprocess | 8. ctypes | 9. math |
| 10. re | 11. random | 12. datatime |
| 13. wikipedia | 14. webbrowser | 15. winshell |
| 16. pyjokes | 17. ecapture | 18. win32com.client |
| 19. pywhatkit | 20. PIL | 21. Pytube |
| 22. pyinstaller | 23. Sys | 24. Time |

1. os

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.

2. socket

This module provides access to the BSD *socket* interface. It is available on all modern Unix systems, Windows, MacOS, and probably additional platforms. `socket` comes under Python's standard utility modules.

3. speech_recognition

Speech Recognition is an important feature in several applications used such as home automation, artificial intelligence, etc. This is useful as it can be used on microcontrollers such as Raspberry Pis with the help of an external microphone.

Installation command : `pip install SpeechRecognition`

4. pyttsx3

pyttsx3 is a text-to-speech conversion library in Python. Unlike alternative libraries, it works offline and is compatible with both Python 2 and 3. An application invokes the pyttsx3.init() factory function to get a reference to a pyttsx3.Engine instance. It is a very easy to use tool which converts the entered text into speech. The pyttsx3 module supports two voices first is female and the second is male which is provided by “**sapi5**” for windows. It supports three **TTS engines** :

- **sapi5** – SAPI5 on Windows
- **nsss** – NSSpeechSynthesizer on Mac OS X
- **espeak** – eSpeak on every other platform

Installation command : pip install pyttsx3

5. pyaudio

PyAudio provides Python bindings for PortAudio v19, the cross-platform audio I/O library. With PyAudio, you can easily use Python to play and record audio on a variety of platforms, such as GNU/Linux, Microsoft Windows, and Apple macOS.

Installation command : pip install PyAudio

6. tkinter

Python offers multiple options for developing GUI (Graphical User Interface). Out of all the GUI methods, tkinter is the most commonly used method. It is a standard Python interface to the Tk GUI toolkit shipped with Python. Python with tkinter is the fastest and easiest way to create the GUI applications. Creating a GUI using tkinter is an easy task.

Installation Command : pip install tkinter

7. subprocess

The subprocess module present in Python(both 2.x and 3.x) is used to run new applications or programs through Python code by creating new processes. It also helps to obtain the input/output/error pipes as well as the exit codes of various commands. subprocess comes under Python’s standard utility modules.

8. ctypes

ctypes is a Python package to create and manipulate C data types in Python, and to call functions in dynamic link libraries/shared dlls. It allows wrapping these libraries in pure Python. ctypes comes under Python’s standard utility modules.

9. math

Python provides the math module to deal with such calculations. Math module provides functions to deal with both basic operations such as addition(+), subtraction(-), multiplication(*), division(/) and advance operations like trigonometric, logarithmic, exponential functions. math comes under Python's standard utility modules.

10. re

A Regular Expressions (RegEx) is a special sequence of characters that uses a search pattern to find a string or set of strings. It can detect the presence or absence of a text by matching it with a particular pattern, and also can split a pattern into one or more sub-patterns. Python provides a re module that supports the use of regex in Python. re comes under Python's standard utility modules.

11. random

Python Random module is an in-built module of Python which is used to generate random numbers. These are pseudo-random numbers means these are not truly random. This module can be used to perform random actions such as generating random numbers, print random a value for a list or string, etc. random comes under Python's standard utility modules.

12. datatime

Python Datetime module supplies classes to work with date and time. These classes provide a number of functions to deal with dates, times and time intervals. Date and datetime are an object in Python, so when you manipulate them, you are actually manipulating objects and not string or timestamps. datetime comes under Python's standard utility modules.

13. wikipedia

Wikipedia is a Python library that makes it easy to access and parse data from Wikipedia. Search Wikipedia, get article summaries, get data like links and images from a page, and more. Wikipedia wraps the MediaWiki API so you can focus on using Wikipedia data, not getting it.

Installation Command : pip install Wikipedia

14. webbrowser

In Python, webbrowser module is a convenient web browser controller. It provides a high-level interface that allows displaying Web-based documents to users.

webbrowser can also be used as a CLI tool. It accepts a URL as the argument with the following optional parameters: -n opens the URL in a new browser window, if possible, and -t opens the URL in a new browser tab.

Installation Command : pip install Wikipedia

15. winshell

The winshell module is a light wrapper around the Windows shell functionality.

It includes convenience functions for accessing special folders, for using the shell's file copy, rename & delete functionality, and a certain amount of support for structured storage.

Installation Command : pip install winshell

16. pyjokes

Python supports creation of random jokes using one of its libraries. Pyjokes is a python library that is used to create one-line jokes for programmers. Informally, it can also be referred as a fun python library which is pretty simple to use.

Installation Command : pip install pyjokes

17. ecapture

It is used to capture images from your Camera.

Installation Command : pip install ecapture

18. win32com.client

Support for COM servers written in Python. The modules in this package provide most of the underlying framework for magically turning Python classes into COM servers, exposing the correct public methods, registering your server in the registry, etc.

19. pywhatkit

Python offers numerous inbuilt libraries to ease our work. Among them pywhatkit is a Python library for sending WhatsApp messages at a certain time, it has several other features too.

Following are some features of pywhatkit module:

- Send WhatsApp messages.
- Play a YouTube video.
- Perform a Google Search.
- Get information on a particular topic.

Installation Command : pip install pywhatkit

20. PIL

Python Imaging Library (expansion of PIL) is the de facto image processing package for Python language. It incorporates lightweight image processing tools that aids in editing, creating and saving images. Support for Python Imaging Library got discontinued in 2011, but a project named pillow forked the original PIL project and added Python3.x support to it. Pillow was announced as a replacement for PIL for future usage. Pillow supports a large number of image file formats including BMP, PNG, JPEG, and TIFF. The library encourages adding support for newer formats in the library by creating new file decoders.

Installation Command : `pip install Pillow`

21. Pytube

YouTube is very popular video sharing website. Downloading a video from YouTube is a tough job. Downloading the Downloader and get the video using that or go to any other website which fetches the video and saves on your computer. Using Python, this task is very easy. Few lines of code will download the video from YouTube for you. For this, there a python library named as ‘pytube’. pytube is a lightweight, dependency-free Python library which is used for downloading videos from the web. *pytube* is a genuine, lightweight, dependency-free Python library (and command-line utility) for downloading YouTube videos.

Installation Command : `pip install pytube`

22. pyinstaller

PyInstaller bundles a Python application and all its dependencies into a single package. The user can run the packaged app without installing a Python interpreter or any modules.

Installation Command : `pip install pyinstaller`

23. sys

The sys module in Python provides various functions and variables that are used to manipulate different parts of the Python runtime environment. It allows operating on the interpreter as it provides access to the variables and functions that interact strongly with the interpreter.

24. time

As the name suggests Python time module allows to work with time in Python. It allows functionality like getting the current time, pausing the Program from executing, etc. So before starting with this module we need to import it.

5. Database Tables

Here 2 **tables** will be created for the user. There will be a **table called Users** in which the details of the user that he has entered at signup time will be stored and **a new table will be created from the mail id of the user**. In which the commands that will be given to the Assistant after the user is signed in will be the user's HostName, user's IPAddress and Current DateTime Store.

Table 1 : Users

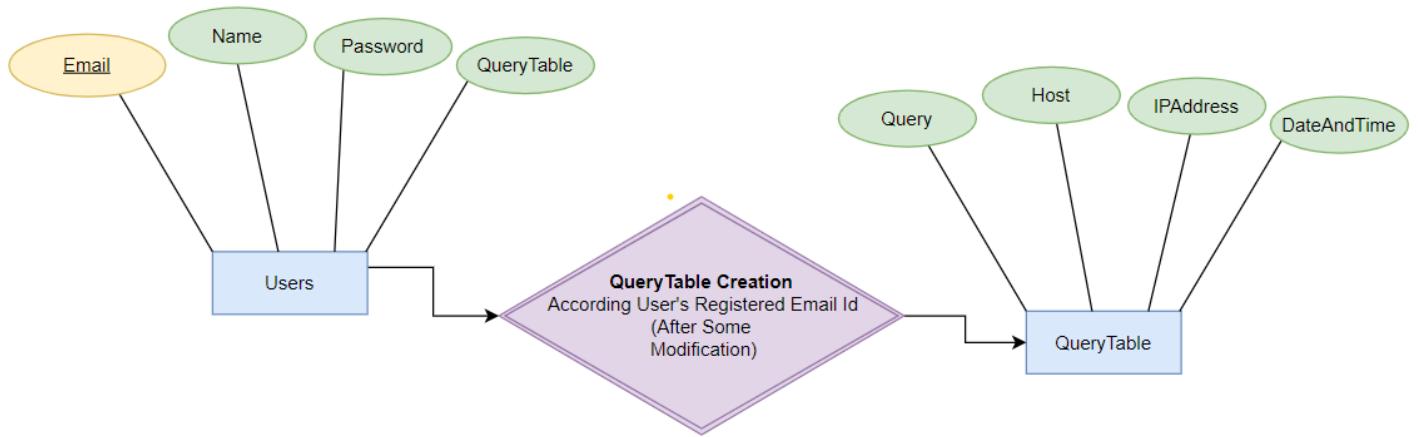
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	Email 	varchar(59)	utf8_unicode_ci		No	None		
2	Name	varchar(59)	utf8_unicode_ci		No	None		
3	Password	varchar(59)	utf8_unicode_ci		No	None		
4	Query Table	varchar(59)	utf8_unicode_ci		No	None		

Indexes 									
Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment	
PRIMARY	BTREE	Yes	No	Email	0	A	No		

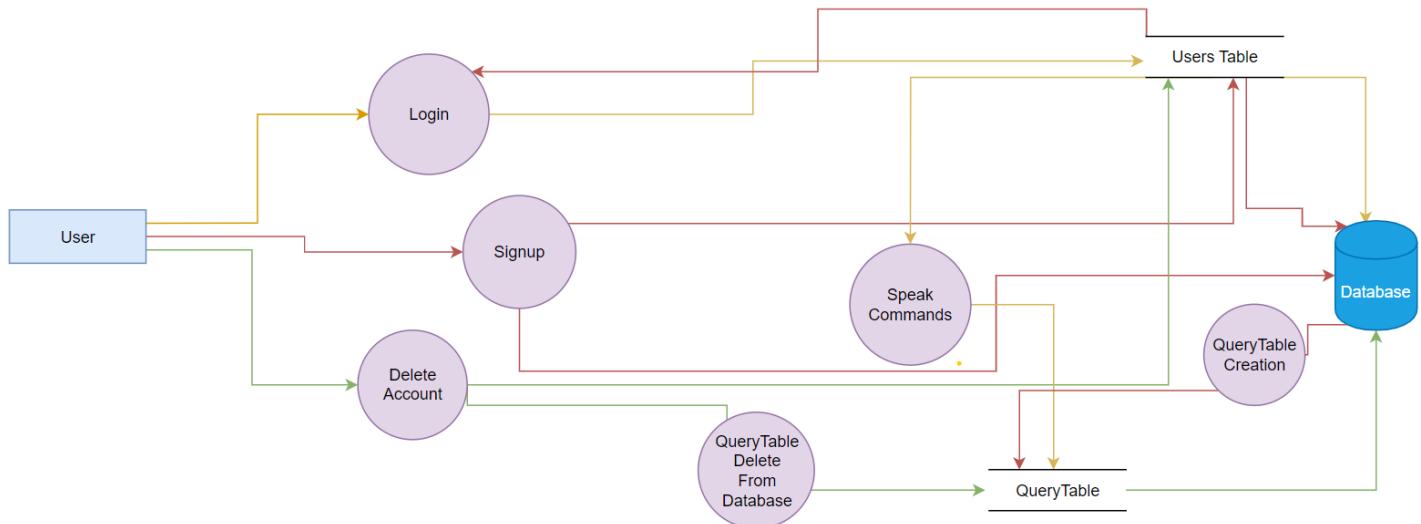
Table 2 : Individual Table - According to User's Email ID

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	Query	varchar(150)	utf8_unicode_ci		No	None		
2	Host	varchar(65)	utf8_unicode_ci		No	None		
3	IPAddress	varchar(59)	utf8_unicode_ci		No	None		
4	DateAndTime	varchar(59)	utf8_unicode_ci		No	None		

6. Entity Relationship Diagram



7. DataFlow Diagram (De Marco & Yorden Method)



8. Source Code(main.py)

```
import os
import sys
import mysql.connector
import socket
import speech_recognition as sr
import pyttsx3
import pyaudio
import tkinter
from tkinter import *
import subprocess
import ctypes
import math
import re
import random
import datetime
import time
import wikipedia
import webbrowser
import winshell
import pyjokes
import ecapture
import win32com.client as wincl
import pywhatkit as pwk
from PIL import Image, ImageTk
from pytube import YouTube

#####
#DATABASE
#####
CONNECTION#####
myconn = ''
mycursor = ''
result = ''
email = ''
Email = ''
User = ''
TName = ''
checkRecord = ''
emailValidate = r'\b[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Z|a-z]{2,}\b'
PasswordValidate = "^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[@$!%*#?&])[A-Za-z\d@$!#%*?&]{8,42}$"

#####
#Database
#####
Connection#####
def dbConnection():
    # myconn = mysql.connector.connect(
    # host="db4free.net",
    # user="g402349",
    # password="G@402349",
    # database="g402349"
    # )

    try:
        myconn = mysql.connector.connect(
            host="sql5.freesqldatabase.com",
            user="sql5531033",
```

```

        password="3RFNy7cent",
        database="sql15531033"
    )
    mycursor = myconn.cursor()
except:
    print("\nInternal Server Error, Please try to open G Assistant After some
time")
    speak("Internal Server Error, Please try to open G Assistant After some
time")
    return False

#####Login Method
def Login():
    connStatus = dbConnection()
    if(connStatus == False):
        sys.exit()

    speak("Enter your Registered email in command prompt")
    email = input("Enter your registered email : ")
    while re.match(emailValidate, email):
        break
    else:
        speak("enter valid email in command prompt")
        email = input("Enter valid Email : ")
        while re.search(emailValidate, email) == None:
            speak("enter valid email in command prompt")
            email = input("Enter valid Email : ")
    speak("Enter your Password in command prompt")
    password = input("Enter your Password : ")
    email = email.lower()
    sql = f"SELECT *FROM Users WHERE Email = '{email}' and Password = '{password}'"

    mycursor.execute(sql)
    checkRecord = mycursor.fetchone()
    while checkRecord is None:
        speak("your entered details is not matched in our database")
        speak("enter your registered email in command prompt")
        email = input("Enter your registered Email : ")
        speak("Enter your Password in command prompt")
        password = input("Enter your Password : ")
        email = email.lower()
        sql = f"SELECT *FROM Users WHERE Email = '{email}' and Password = '{password}'"
        mycursor.execute(sql)
        checkRecord = mycursor.fetchone()
    if checkRecord is not None:
        global User
        global Email
        Email = checkRecord[0]
        Email = Email.replace(',', ' ')
        User = checkRecord[1]
        User = User.replace(',', ' ')
        speak(f"Hi, {User} Your Authentication is Done now you can use G
Assistant....")

#####Signup Method
def Signup():

```

```

connStatus = dbConnection()
if connStatus == False:
    sys.exit()

speak("Enter your Name in command prompt")
Name = input("Enter your Name : ")
speak("Enter your email in command prompt")
email = input("Enter your email : ")
while re.match(emailValidate, email):
    break
    pass
else:
    speak("enter valid email in command prompt")
    email = input("Enter valid Email : ")
    while re.match(emailValidate, email):
        break
        pass
    else:
        speak("enter valid email in command prompt")
        email = input("Enter valid Email : ")
speak("Enter Password in command prompt")
Password = input("Enter Password : ")
res = re.search(PasswordValidate, Password)
email = email.lower()
sql = f"SELECT *FROM Users WHERE Email = '{email}'"
mycursor.execute(sql)
checkRecord = mycursor.fetchone()
while checkRecord is None:
    break
while re.search(emailValidate, email) == None:
    speak("enter valid email in command prompt")
    email = input("Enter valid Email : ")
while res == None:
    print("\n Warning : \nMinimum password length is needs required more than 8 character")
    print("password can be combination of uppercase latters, lowercase latters, numbers and special characters")
    speak("Warning")
    speak("Minimum password length is nneds too be required more than 8 character")
    speak("password can be combination of uppercase latters, lowercase latters, numbers and special characters")
    speak("Enter password in valid format ")
    Password = input("Enter password in valid format : ")
    res = re.search(PasswordValidate, Password)
while checkRecord is not None:
    speak("This email is registered in our database try to enter different email")
    email = input("Enter different Email : ")
    while re.search(emailValidate, email) == None:
        speak("enter valid email in command prompt")
        email = input("Enter valid Email : ")
    speak("Enter Password")
    Password = input("Enter Password : ")
    res = re.search(PasswordValidate, Password)
    while res == None:

```

```

        print("\n Warning : \nMinimum password length is needs required more
than 8 character")
        print("password can be combination of uppercase latters, lowercase
latters, numbers and special characters")
        speak("Warning")
        speak("Minimum password length is nneds too be required more than 8
character")
        speak("password can be combination of uppercase latters, lowercase
latters, numbers and special characters")
        speak("Enter password in valid format ")
        Password = input("Enter password in valid format : ")
        res = re.search(PasswordValidate, Password)
        email = email.lower()
        sql = f"SELECT *FROM Users WHERE Email = '{email}'"
        mycursor.execute(sql)
        checkRecord = mycursor.fetchone()
        email = email.lower()
        sql = f"SELECT *FROM Users WHERE Email = '{email}'"
        mycursor.execute(sql)
        checkRecord = mycursor.fetchone()
        if mycursor.rowcount:
            #####Create User Table
            TName = email.lower()
            TName = TName.replace("@", "_")
            TName = TName.replace(".", "_")
            if checkRecord is None:
                email = email.lower()
                sql = "Insert into Users>Email, Name, Password, QueryTable)VALUES(%s,
%s, %s, %s)"
                val = (email, Name, Password, TName)
                mycursor.execute(sql, val)
                myconn.commit()
            if res is not None:
                mycursor.execute(f"CREATE TABLE {TName} (Query VARCHAR(150), Host
VARCHAR(59), IPAddress VARCHAR(65), DateAndTime VARCHAR(59))")
                email.lower()
                sql = f"SELECT *FROM Users WHERE Email = '{email}' and Password =
'{Password}'"
                mycursor.execute(sql)
                checkRecord = mycursor.fetchone()
                User = checkRecord[1]
                User = User.replace(',', '')
                if True:
                    speak(f"Hi, {User} Your registration is Successfully completed now you
can use G Assistant After login....")
                    Login()
                else:
                    speak("Something went Wrong try again to Signup")
                    Signup()

def DeleteAccount():
    connStatus = dbConnection()
    if connStatus == False:
        sys.exit()

    speak("Enter your Registered email in command prompt")
    email = input("Enter your registered email : ")

```

```

while re.match(emailValidate, email):
    break
else:
    speak("enter valid email in command prompt")
    email = input("Enter valid Email : ")
speak("Enter your Password in command prompt")
password = input("Enter your Password : ")
email = email.lower()
sql = f"SELECT *FROM Users WHERE Email = '{email}' and Password = '{password}'"
mycursor.execute(sql)
checkRecord = mycursor.fetchone()
User = checkRecord[1]
User = User.replace(',', '')
while checkRecord is None:
    speak("your entered details is not matched in our database")
    speak("enter your registered email in command prompt")
    email = input("Enter your registered Email : ")
    speak("Enter your Password in command prompt")
    password = input("Enter your Password : ")
    email = email.lower()
    sql = f"SELECT *FROM Users WHERE Email = '{email}' and Password = '{password}'"
    mycursor.execute(sql)
    checkRecord = mycursor.fetchone()
    Name = checkRecord[0]
    Name = Name.replace(',', '')
TName = email.replace("@", "_")
TName = TName.replace(".", "_")
email = email.lower()
sql2 = f"DELETE FROM Users WHERE Email = '{email}'"
sql3 = f"DROP TABLE {TName}"
mycursor.execute(sql2)
if mycursor.rowcount:
    speak(f"Hi, {User} your account is deleted successfully.....")
myconn.commit()
mycursor.execute(sql3)

# def CT():
#     res = mycursor.execute(f"CREATE TABLE Users (Email VARCHAR(59) primary key,
Name VARCHAR(59), Password VARCHAR(59), QueryTable VARCHAR(59))")
#     if res:
#         print("Table is Created Successfully.....")
#####
#####Guess The Number Console
Game#####
def GuessNumber() :
    print("Welcome in Number Guessing Game \n")
    speak("Welcome in Number Guessing Game")
    print("Application Type : Console Game")
    speak("Application Type : Console Game")
    print("Version : 2.0.1 LTS")
    speak("Version : 2.0.1 LTS")
    print("Developer : GHANSHYAM VAJA", "\n")
    speak("Developer : GHANSHYAM vaja")

```

```

rand = 0
randomNumber = 0
userNumber = 0
chooseLevel = 0
easyCount = 5
mediumCount = 7
hardCount = 10
ultrahardCount = 15
leftCount = 0

print("      Level          Range          Turns")
print("1 - Easy      (1 to 100)      5\n2 - Medium     (1 to 1000)
7\n3 - Hard       (1 to 10000)    10\n4 - Ultra Hard (1 to 100000)   15")
speak("1 for Easy      Range = 1 to 100      turns = 5\n")
speak("2 for Medium     Range = 1 to 1000     turns=7\n")
speak("3 for Hard       Range=1 to 10000    turns=10\n")
speak("4 for Ultra Hard Range=1 to 100000   turns=15")
print("\nChoose Level(1 | 2 | 3 | 4) : ", end="")
speak("Choose Level between 1 to 4")
chooseLevel = int(input())
while (chooseLevel <= 0 or chooseLevel > 4) :
    print("Choose Level(1 | 2 | 3 | 4) : ", end="")
    speak("Choose Level")
    chooseLevel = int(input())
if (chooseLevel == 1) :
    randomNumber = random.randint(0, 101)
elif(chooseLevel == 2) :
    randomNumber = random.randint(0, 1001)
elif(chooseLevel == 3) :
    randomNumber = random.randint(0, 10001)
else :
    randomNumber = random.randint(0,100001)
while True :
    if (easyCount == 0 or mediumCount == 0 or hardCount == 0 or
ultrahardCount == 0) :
        print("\n-----")
        print("GAME OVER", end="")
        speak("GAME OVER")
        break
    print("\nGuess A Number Between 1 To ", end="")
    speak("\nGuess A Number Between 1 To")
    if (chooseLevel == 1) :
        easyCount -= 1
        leftCount = easyCount
        print("100", end="")
        speak("100")
    elif(chooseLevel == 2) :
        mediumCount -= 1
        leftCount = mediumCount
        print("1000", end="")
        speak("1000")
    elif(chooseLevel == 3) :
        hardCount -= 1
        leftCount = hardCount
        print("10000", end="")
        speak("10000")
    else :

```

```

        ultrahardCount -= 1
        leftCount = ultrahardCount
        print("100000", end="")
        speak("100000")
    print(" : ", end="")
    userNumber = int(input())
    if (userNumber == randomNumber) :
        print("\n-----")
        print("\nCongratulations, Your Guess Is Correct, YOU WON.....")
        speak("Congratulations, Your Guess Is Correct, YOU WON.....")
    elif(userNumber > randomNumber) :
        print("Your Guess Is Large      Available Turn : " + str(leftCount))
        speak("Your Guess Is Large")
        speak("Available Turn : " + str(leftCount))
    elif(userNumber < randomNumber) :
        print("Your Guess Is Small      Available Turn : " + str(leftCount))
        speak("Your Guess Is Small")
        speak("Available Turn : " + str(leftCount))
    if((userNumber > 0) == False) :
        break
    print(" \n-----")
    print("Your Answer Is : ")
    print(randomNumber)
    speak("Your Answer Is ")
    speak(randomNumber)
    print("-----")
#####
#####Snake and Ladder Console
Game#####
class Play_With_Computer :
    WONPOINT = 100
    rand = random.randint(1, 7)
    rollDice = ''
    playerPosition = 0
    computerPosition = 0
    playerDice = 0
    computerDice = 0
    Name = None
    Snake_Mouth_Position = [0] * (8)
    Snake_Tail_Position = [0] * (8)
    Ladder_Up_Position = [0] * (8)
    Ladder_Down_Position = [0] * (8)
    flag = 0
    Level = ''
    ctr = 0
    def PlayerAndComputerPosition(self) :
        if (Play_With_Computer.playerPosition >
Play_With_Computer.WONPOINT) :
            Play_With_Computer.playerPosition -=
Play_With_Computer.playerDice
            if (Play_With_Computer.computerPosition >
Play_With_Computer.WONPOINT) :
                Play_With_Computer.computerPosition -=
Play_With_Computer.computerDice
                # playerPosition With Snake Tail Position

```



```

                elif(Play_With_Computer.computerPosition ==
Play_With_Computer.Ladder_Down_Position[i]) :
                    Play_With_Computer.computerPosition =
Play_With_Computer.Ladder_Up_Position[i]
                        i += 1
                        # player position display
                        print("\n----->Player's Position<-----")
-----")
                        speak("Player's Position")
                        i = 0
                        while (i < 2) :
                            if (i == 0) :
                                print(f" \n YOU ->
{Play_With_Computer.Name} = {Play_With_Computer.playerPosition}")
                                speak(f" \n YOU ->
{Play_With_Computer.Name} = {Play_With_Computer.playerPosition}")
                            else :
                                print(f" \n COMPUTER =
{Play_With_Computer.computerPosition}")
                                speak(f" \n COMPUTER =
{Play_With_Computer.computerPosition}")
                        i += 1
                def IsWon(self) :
                    if (Play_With_Computer.playerPosition == Play_With_Computer.WONPOINT) :
                        print("\n-----WINNER-----\n")
                        print(f" CONGRATS {Play_With_Computer.Name}")
                        print("\n-----WINNER-----\n")
                        speak("\n-----WINNER-----\n")
                        speak(f" CONGRATS {Play_With_Computer.Name}")
                        speak("\n-----WINNER-----\n")
-----)
                        self.flag = 1
                    if (Play_With_Computer.computerPosition == Play_With_Computer.WONPOINT) :
                        print("\n-----WINNER-----\n")
                        print(" COMPUTER \n")
                        print(" YOU LOSE !!!!!")
----- \n")
                        speak("WINNER")
                        speak("COMPUTER")
                        speak("YOU LOSE")
                        self.flag = 1
                # setGame()
                def setGame(self) :
                    print("----->SET GAME<-----\n", end = "")
                    speak("SET GAME")
                    print("          Levels : 1 - EASY      2 - MEDIUM      3 - HARD \n \n", end = "")
                    speak("Levels")
                    speak("1 - EASY")
                    speak("2 - MEDIUM")
                    speak("3 - HARD")

```

```

print("Choose a Level (1 | 2 | 3) : ", end="")
speak("Choose a Level")
speak("1 or 2 or 3")
self.Level = input()[0]
while (self.Level != '1' and self.Level != '2' and self.Level != '3') :
    print("Choose a valid Level (1 | 2 | 3) : ", end="")
    speak("Choose a valid Level")
    self.Level = input()[0]
print("\nEnter Your Name : ", end="")
speak("Enter Your Name")
Play_With_Computer.Name = input()
def SnakesAndLaddersPosition(self) :
    if (self.Level == '1') :
        # Snake Mouth Position Setting
        Play_With_Computer.Snake_Mouth_Position[0] = random.randint(95, (99 -
95) + 95)
        Play_With_Computer.Snake_Mouth_Position[1] = random.randint(60, (65 -
60) + 60)
        Play_With_Computer.Snake_Mouth_Position[2] = random.randint(30, (35 -
30) + 30)
        # Snake Tail Position Setting
        Play_With_Computer.Snake_Tail_Position[0] = random.randint(24, (26 -
24) + 24)
        Play_With_Computer.Snake_Tail_Position[1] = random.randint(51, (56 -
51) + 51)
        Play_With_Computer.Snake_Tail_Position[2] = random.randint(15, (19 -
15) + 15)
        elif(self.Level == '2') :
            # Snake Mouth Position Setting
            Play_With_Computer.Snake_Mouth_Position[0] = random.randint(98, (99 -
98) + 98)
            Play_With_Computer.Snake_Mouth_Position[1] = random.randint(87, (87 -
81) + 81)
            Play_With_Computer.Snake_Mouth_Position[2] = random.randint(55, (59 -
55) + 55)
            Play_With_Computer.Snake_Mouth_Position[3] = random.randint(32, (38 -
32) + 32)
            Play_With_Computer.Snake_Mouth_Position[4] = random.randint(15, (19 -
15) + 15)
            # Snake Tail Position Setting
            Play_With_Computer.Snake_Tail_Position[0] = random.randint(51, (56 -
51) + 51)
            Play_With_Computer.Snake_Tail_Position[1] = random.randint(72, (78 -
72) + 72)
            Play_With_Computer.Snake_Tail_Position[2] = random.randint(41, (42 -
41) + 41)
            Play_With_Computer.Snake_Tail_Position[3] = random.randint(26, (29 -
26) + 26)
            Play_With_Computer.Snake_Tail_Position[4] = random.randint(6, (9 -
6) + 6)
        else :
            # Snake Mouth Position Setting
            Play_With_Computer.Snake_Mouth_Position[0] = random.randint(98, (99 -
98) + 98)
            Play_With_Computer.Snake_Mouth_Position[1] = random.randint(92, (96 -
92) + 92)

```

```

        Play_With_Computer.Snake_Mouth_Position[2] = random.randint(71, (74 -
71) + 71)
        Play_With_Computer.Snake_Mouth_Position[3] = random.randint(62, (65 -
62) + 62)
        Play_With_Computer.Snake_Mouth_Position[4] = random.randint(41, (42 -
41) + 41)
        Play_With_Computer.Snake_Mouth_Position[5] = random.randint(65, (69 -
65) + 65)
        Play_With_Computer.Snake_Mouth_Position[6] = random.randint(24, (26 -
24) + 24)
        Play_With_Computer.Snake_Mouth_Position[7] = random.randint(15, (19 -
15) + 15)
        # Snake Tail Position Setting
        Play_With_Computer.Snake_Tail_Position[0] = random.randint(51, (56 -
51) + 51)
        Play_With_Computer.Snake_Tail_Position[1] = random.randint(41, (45 -
41) + 41)
        Play_With_Computer.Snake_Tail_Position[2] = random.randint(32, (36 -
32) + 32)
        Play_With_Computer.Snake_Tail_Position[3] = random.randint(22, (26 -
22) + 22)
        Play_With_Computer.Snake_Tail_Position[4] = random.randint(26, (29 -
26) + 26)
        Play_With_Computer.Snake_Tail_Position[5] = random.randint(15, (19 -
15) + 15)
        Play_With_Computer.Snake_Tail_Position[6] = random.randint(15, (19 -
15) + 15)
        Play_With_Computer.Snake_Tail_Position[7] = random.randint(6, (9 - 6) +
6)
    if (self.Level == '1') :
        # Ladder Down Position Setting
        Play_With_Computer.Ladder_Down_Position[0] = random.randint(6, (9 - 6) +
6)
        Play_With_Computer.Ladder_Down_Position[1] = random.randint(65, (69 -
65) + 65)
        Play_With_Computer.Ladder_Down_Position[2] = random.randint(87, (87 -
81) + 81)
        # Laddet Up Position Setting
        Play_With_Computer.Ladder_Up_Position[0] = random.randint(81, (87 - 81) +
81)
        Play_With_Computer.Ladder_Up_Position[1] = random.randint(92, (96 - 92) +
92)
        Play_With_Computer.Ladder_Up_Position[2] = random.randint(96, (99 - 96) +
96)
    elif(self.Level == '2') :
        # Ladder Down Position Setting
        Play_With_Computer.Ladder_Down_Position[0] = random.randint(15, (19 -
15) + 15)
        Play_With_Computer.Ladder_Down_Position[1] = random.randint(37, (38 -
37) + 37)
        Play_With_Computer.Ladder_Down_Position[2] = random.randint(51, (59 -
51) + 51)
        Play_With_Computer.Ladder_Down_Position[3] = random.randint(65, (69 -
65) + 65)
        Play_With_Computer.Ladder_Down_Position[4] = random.randint(81, (86 -
81) + 81)
        # Ladder Up Position Setting

```

```

+     Play_With_Computer.Ladder_Up_Position[0] = random.randint(81, (86 - 81)
+ 81)     Play_With_Computer.Ladder_Up_Position[1] = random.randint(41, (42 - 41)
+ 41)     Play_With_Computer.Ladder_Up_Position[2] = random.randint(81, (86 - 81)
+ 81)     Play_With_Computer.Ladder_Up_Position[3] = random.randint(92, (96 - 92)
+ 92)     Play_With_Computer.Ladder_Up_Position[4] = random.randint(96, (99 - 96)
+ 96)
+     else :
+         # Ladder Down Position Setting
+         Play_With_Computer.Ladder_Down_Position[0] = random.randint(6, (9 - 6)
+ 6)         Play_With_Computer.Ladder_Down_Position[1] = random.randint(24, (26 -
24) + 24)         Play_With_Computer.Ladder_Down_Position[2] = random.randint(36, (38 -
36) + 36)         Play_With_Computer.Ladder_Down_Position[3] = random.randint(41, (45 -
41) + 41)         Play_With_Computer.Ladder_Down_Position[4] = random.randint(54, (59 -
54) + 54)         Play_With_Computer.Ladder_Down_Position[5] = random.randint(68, (69 -
68) + 68)         Play_With_Computer.Ladder_Down_Position[6] = random.randint(72, (78 -
72) + 72)         Play_With_Computer.Ladder_Down_Position[7] = random.randint(81, (86 -
81) + 81)
+         # Ladder Up Position Setting
+         Play_With_Computer.Ladder_Up_Position[0] = random.randint(81, (87 - 81)
+ 81)         Play_With_Computer.Ladder_Up_Position[1] = random.randint(92, (96 - 92)
+ 92)         Play_With_Computer.Ladder_Up_Position[2] = random.randint(74, (78 - 74)
+ 74)         Play_With_Computer.Ladder_Up_Position[3] = random.randint(65, (69 - 65)
+ 65)         Play_With_Computer.Ladder_Up_Position[4] = random.randint(81, (87 - 81)
+ 81)         Play_With_Computer.Ladder_Up_Position[5] = random.randint(74, (78 - 74)
+ 74)         Play_With_Computer.Ladder_Up_Position[6] = random.randint(95, (96 - 95)
+ 95)         Play_With_Computer.Ladder_Up_Position[7] = random.randint(92, (99 - 92)
+ 92)
+     def SnakesAndLaddersPositionPrint(self) :
+         print("\n----->SNAKES POSITION | LADDERS POSITION <----")
+         speak("SNAKES POSITION and LADDERS POSITION")
+         if (self.Level == '1') :
+             i = 0
+             while (i < 3) :
+                 if (i == 0) :
+                     print(f"
{Play_With_Computer.Snake_Mouth_Position[i]}      to
{Play_With_Computer.Snake_Tail_Position[i]}      |

```

```

0{Play_With_Computer.Ladder_Down_Position[i]}      to
{Play_With_Computer.Ladder_Up_Position[i]}")        |
else :
    print(f"
{Play_With_Computer.Snake_Mouth_Position[i]}      to
{Play_With_Computer.Snake_Tail_Position[i]}        |
{Play_With_Computer.Ladder_Down_Position[i]}      to
{Play_With_Computer.Ladder_Up_Position[i]}")        |
    i += 1
elif(self.Level == '2') :
    i = 0
    while (i < 5) :
        if (i == 0) :
            print(f"
{Play_With_Computer.Snake_Mouth_Position[i]}      to
{Play_With_Computer.Snake_Tail_Position[i]}        |
{Play_With_Computer.Ladder_Down_Position[i]}      to
{Play_With_Computer.Ladder_Up_Position[i]}")        |
                elif(i == 4) :
                    print(f"
{Play_With_Computer.Snake_Mouth_Position[i]}      to
0{Play_With_Computer.Snake_Tail_Position[i]}        |
{Play_With_Computer.Ladder_Down_Position[i]}      to
{Play_With_Computer.Ladder_Up_Position[i]}")        |
            else :
                print(f"
{Play_With_Computer.Snake_Mouth_Position[i]}      to
{Play_With_Computer.Snake_Tail_Position[i]}        |
{Play_With_Computer.Ladder_Down_Position[i]}      to
{Play_With_Computer.Ladder_Up_Position[i]}")        |
                i += 1
        else :
            i = 0
            while (i < 8) :
                if (i == 0) :
                    print(f"
{Play_With_Computer.Snake_Mouth_Position[i]}      to
{Play_With_Computer.Snake_Tail_Position[i]}        |
0{Play_With_Computer.Ladder_Down_Position[i]}      to
{Play_With_Computer.Ladder_Up_Position[i]}")        |
                        elif(i == 7) :
                            print(f"
{Play_With_Computer.Snake_Mouth_Position[i]}      to
0{Play_With_Computer.Snake_Tail_Position[i]}        |
{Play_With_Computer.Ladder_Down_Position[i]}      to
{Play_With_Computer.Ladder_Up_Position[i]}")        |
                    else :
                        print(f"
{Play_With_Computer.Snake_Mouth_Position[i]}      to
{Play_With_Computer.Snake_Tail_Position[i]}        |
{Play_With_Computer.Ladder_Down_Position[i]}      to
{Play_With_Computer.Ladder_Up_Position[i]}")        |
                        i += 1
                print("-----")
\n", end="")
    def RollDice(self) :

```

```

        print(" \n----->Roll The Dice<-----\n", end = "")
        speak("Roll The Dice")
        print(f" \n          {Play_With_Computer.Name}'s Turn (Enter character \"R\" or \"r\" ) : ", end = "")
        speak("Enter character Capital or Small R")
        Play_With_Computer.rollDice = input()
        while (Play_With_Computer.rollDice != 'R' and Play_With_Computer.rollDice != 'r') :
            print("           Enter Valid Character R or r : ", end = "")
            speak("Enter Valid Character Capital or Small R")
            Play_With_Computer.rollDice = input()
        Play_With_Computer.playerDice = random.randint(1, (6 - 1) + 1)
        Play_With_Computer.playerPosition += Play_With_Computer.playerDice
        print(f"           Your Dice Score : {Play_With_Computer.playerDice}")
        I = 0
        while (I < 8) :
            if (self.Level == '1' and I < 3) :
                if (Play_With_Computer.playerPosition ==
Play_With_Computer.Snake_Mouth_Position[I]) :
                    print(f" \n           ----->Oops, {Play_With_Computer.Name} Swallowed By Snake<-----")
                    speak(f"Oops, {Play_With_Computer.Name} Swallowed By Snake")
                elif(Play_With_Computer.playerPosition ==
Play_With_Computer.Ladder_Down_Position[I]) :
                    print(f" \n           ----->Hurry, {Play_With_Computer.Name} Climb Up The Ladder<-----")
                    speak(f">Hurry, {Play_With_Computer.Name} Climb Up The Ladder")
                elif(self.Level == '2' and I < 5) :
                    if (Play_With_Computer.playerPosition ==
Play_With_Computer.Snake_Mouth_Position[I]) :
                        print(f" \n           ----->Oops, {Play_With_Computer.Name} Swallowed By Snake<-----")
                        speak(f"nOops, {Play_With_Computer.Name} Swallowed By Snake")
                    elif(Play_With_Computer.playerPosition ==
Play_With_Computer.Ladder_Down_Position[I]) :
                        print(f" \n           ----->Hurry, {Play_With_Computer.Name} Climb Up The Ladder<-----")
                        speak(f">Hurry, {Play_With_Computer.Name} Climb Up The Ladder")
                else :
                    if (Play_With_Computer.playerPosition ==
Play_With_Computer.Snake_Mouth_Position[I]) :
                        print(f" \n           ----->Oops, {Play_With_Computer.Name} Swallowed By Snake<-----")
                        speak(f"Oops, {Play_With_Computer.Name} Swallowed By Snake")
                    elif(Play_With_Computer.playerPosition ==
Play_With_Computer.Ladder_Down_Position[I]) :
                        print(f" \n           ----->Hurry, {Play_With_Computer.Name} Climb Up The Ladder<-----")
                        speak(f">Hurry, {Play_With_Computer.Name} Climb Up The Ladder")
                    I += 1
            print("\n          Computer's Turn (Wait) : ", end = "")
            speak("Computer's Turn (Wait)")
            Play_With_Computer.computerDice = random.randint(1, (6 - 1) + 1)
            Play_With_Computer.computerPosition += Play_With_Computer.computerDice
            try :

```

```

# Thread
k = 0
while (k < 5) :
    Thread.sleep(501)
    k += 1
except Exception as expn :
    if (random.randint(6, (9 - 6) + 6)% 2 == 0) :
        print(f"R \n Computer\''s Dice Score : {Play_With_Computer.computerDice}")
    else :
        print(f"r \n Computer\''s Dice Score : {Play_With_Computer.computerDice}")
I = 0
while (I < 8) :
    if (self.Level == '1' and I < 3) :
        if (Play_With_Computer.computerPosition ==
Play_With_Computer.Snake_Mouth_Position[I]) :
            print(" \n ----->Computer Swallowed By Snake<-----")
    speak("Computer Swallowed By Snake")
    elif(Play_With_Computer.computerPosition ==
Play_With_Computer.Ladder_Down_Position[I]) :
        print(" \n ----->Computer Climb Up The Ladder<-----")
    speak("Computer Climb Up The Ladder")
    elif(self.Level == '2' and I < 5) :
        if (Play_With_Computer.computerPosition ==
Play_With_Computer.Snake_Mouth_Position[I]) :
            print(" \n ----->Computer Swallowed By Snake<-----")
    speak("Computer Swallowed By Snake")
    elif(Play_With_Computer.computerPosition ==
Play_With_Computer.Ladder_Down_Position[I]) :
        print(" \n ----->Computer Climb Up The Ladder<-----")
    speak("Computer Climb Up The Ladder")
    else :
        if (Play_With_Computer.computerPosition ==
Play_With_Computer.Snake_Mouth_Position[I]) :
            print(" \n ----->Computer Swallowed By Snake<-----")
    speak("Computer Swallowed By Snake")
    elif(Play_With_Computer.computerPosition ==
Play_With_Computer.Ladder_Down_Position[I]) :
        print(" \n ----->Computer Climb Up The Ladder<-----")
    speak("Computer Climb Up The Ladder")
    I += 1
if (self.flag != 1) :
    self.PlayerAndComputerPosition()
    self.IsWon()
#Play_with_friends class
class Play_With_friends :
    WONPOINT = 100
    rand = random.randint(1, 7)
    n = 100
    rollDice = ' '

```

```

playerPosition = [0] * (n)
diceScore = [0] * (n)
Names = [None] * (n)
Snake_Mouth_Position = [0] * (8)
Snake_Tail_Position = [0] * (8)
Ladder_Up_Position = [0] * (8)
Ladder_Down_Position = [0] * (8)
flag = 0
Level = ''
Count = 0
Snake_flag = 0
Ladder_flag = 0
# setGame()
def setGame(self) :
    print("----->SET GAME<-----")
    speak("SET GAME")
    print("          Levels :   1 - EASY      2 - MEDIUM      3 - HARD \n \n", end = "")
    speak("Levels")
    speak("1 - EASY")
    speak("2 - MEDIUM")
    speak("3 - HARD")
    print("Choose a Level (1 | 2 | 3 ) : ", end = "")
    speak("Choose a Level")
    speak("1 or 2 or 3 ")
    self.Level = input()
    while (self.Level != '1' and self.Level != '2' and self.Level != '3') :
        print("Choose a valid Level (1 | 2 | 3 ) : ", end = "")
        speak("Choose a valid Level")
        self.Level = input()
# SetPlayer()
def SetPlayer(self) :
    print("\nEnter no. of Players : ", end = "")
    speak("Enter number of Players")
    Play_With_friends.n = int(input())
    while (self.n < 0) :
        print("Enter Valid no. of Players : ")
        speak("Enter Valid no. of Players")
        Play_With_friends.n = input()
    print("\nEnter players Names : ")
    speak("Enter players Names")
    i = 0
    while (i < Play_With_friends.n) :
        print(f"player {i + 1} : ", end = "")
        speak(f"Enter player {i + 1} Name")
        Play_With_friends.Names[i] = input()
        i += 1
def SnakesAndLaddersPosition(self) :
    if (self.Level == '1') :
        # Snake Mouth Position Setting
        Play_With_friends.Snake_Mouth_Position[0] = random.randint(95, (99 -
95) + 95)
        Play_With_friends.Snake_Mouth_Position[1] = random.randint(60, (65 -
60) + 60)
        Play_With_friends.Snake_Mouth_Position[2] = random.randint(30, (35 -
30) + 30)

```

```

        # Snake Tail Position Setting
        Play_With_friends.Snake_Tail_Position[0] = random.randint(24, (26 - 24)
+ 24)
        Play_With_friends.Snake_Tail_Position[1] = random.randint(51, (56 - 51)
+ 51)
        Play_With_friends.Snake_Tail_Position[2] = random.randint(15, (19 - 15)
+ 15)
    elif(self.Level == '2') :
        # Snake Mouth Position Setting
        Play_With_friends.Snake_Mouth_Position[0] = random.randint(98, (99 -
98) + 98)
        Play_With_friends.Snake_Mouth_Position[1] = random.randint(87, (87 -
81) + 81)
        Play_With_friends.Snake_Mouth_Position[2] = random.randint(55, (59 -
55) + 55)
        Play_With_friends.Snake_Mouth_Position[3] = random.randint(32, (38 -
32) + 32)
        Play_With_friends.Snake_Mouth_Position[4] = random.randint(15, (19 -
15) + 15)
        # Snake Tail Position Setting
        Play_With_friends.Snake_Tail_Position[0] = random.randint(51, (56 - 51)
+ 51)
        Play_With_friends.Snake_Tail_Position[1] = random.randint(72, (78 - 72)
+ 72)
        Play_With_friends.Snake_Tail_Position[2] = random.randint(41, (42 - 41)
+ 41)
        Play_With_friends.Snake_Tail_Position[3] = random.randint(26, (29 - 26)
+ 26)
        Play_With_friends.Snake_Tail_Position[4] = random.randint(6, (9 - 6) +
6)
    else :
        # Snake Mouth Position Setting
        Play_With_friends.Snake_Mouth_Position[0] = random.randint(98, (99 -
98) + 98)
        Play_With_friends.Snake_Mouth_Position[1] = random.randint(92, (96 -
92) + 92)
        Play_With_friends.Snake_Mouth_Position[2] = random.randint(71, (74 -
71) + 71)
        Play_With_friends.Snake_Mouth_Position[3] = random.randint(62, (65 -
62) + 62)
        Play_With_friends.Snake_Mouth_Position[4] = random.randint(41, (42 -
41) + 41)
        Play_With_friends.Snake_Mouth_Position[5] = random.randint(65, (69 -
65) + 65)
        Play_With_friends.Snake_Mouth_Position[6] = random.randint(24, (26 -
24) + 24)
        Play_With_friends.Snake_Mouth_Position[7] = random.randint(15, (19 -
15) + 15)
        # Snake Tail Position Setting
        Play_With_friends.Snake_Tail_Position[0] = random.randint(51, (56 - 51)
+ 51)
        Play_With_friends.Snake_Tail_Position[1] = random.randint(41, (45 - 41)
+ 41)
        Play_With_friends.Snake_Tail_Position[2] = random.randint(32, (36 - 32)
+ 32)
        Play_With_friends.Snake_Tail_Position[3] = random.randint(22, (26 - 22)
+ 22)

```

```

        Play_With_friends.Snake_Tail_Position[4] = random.randint(26, (29 - 26)
+ 26)     Play_With_friends.Snake_Tail_Position[5] = random.randint(15, (19 - 15)
+ 15)     Play_With_friends.Snake_Tail_Position[6] = random.randint(15, (19 - 15)
+ 15)     Play_With_friends.Snake_Tail_Position[7] = random.randint(6, (9 - 6) +
6)     if (self.Level == '1') :
        # Ladder Down Position Setting
        Play_With_friends.Ladder_Down_Position[0] = random.randint(6, (9 - 6) +
6)     Play_With_friends.Ladder_Down_Position[1] = random.randint(65, (69 -
65) + 65)     Play_With_friends.Ladder_Down_Position[2] = random.randint(87, (87 -
81) + 81)     # Laddet Up Position Setting
        Play_With_friends.Ladder_Up_Position[0] = random.randint(81, (87 - 81)
+ 81)     Play_With_friends.Ladder_Up_Position[1] = random.randint(92, (96 - 92)
+ 92)     Play_With_friends.Ladder_Up_Position[2] = random.randint(96, (99 - 96)
+ 96)     elif(self.Level == '2') :
        # Ladder Down Position Setting
        Play_With_friends.Ladder_Down_Position[0] = random.randint(15, (19 -
15) + 15)     Play_With_friends.Ladder_Down_Position[1] = random.randint(37, (38 -
37) + 37)     Play_With_friends.Ladder_Down_Position[2] = random.randint(51, (59 -
51) + 51)     Play_With_friends.Ladder_Down_Position[3] = random.randint(65, (69 -
65) + 65)     Play_With_friends.Ladder_Down_Position[4] = random.randint(81, (86 -
81) + 81)     # Ladder Up Position Setting
        Play_With_friends.Ladder_Up_Position[0] = random.randint(81, (86 - 81)
+ 81)     Play_With_friends.Ladder_Up_Position[1] = random.randint(41, (42 - 41)
+ 41)     Play_With_friends.Ladder_Up_Position[2] = random.randint(81, (86 - 81)
+ 81)     Play_With_friends.Ladder_Up_Position[3] = random.randint(92, (96 - 92)
+ 92)     Play_With_friends.Ladder_Up_Position[4] = random.randint(96, (99 - 96)
+ 96)     else :
        # Ladder Down Position Setting
        Play_With_friends.Ladder_Down_Position[0] = random.randint(6, (9 - 6) +
6)     Play_With_friends.Ladder_Down_Position[1] = random.randint(24, (26 -
24) + 24)     Play_With_friends.Ladder_Down_Position[2] = random.randint(36, (38 -
36) + 36)     Play_With_friends.Ladder_Down_Position[3] = random.randint(41, (45 -
41) + 41)

```

```

        Play_With_friends.Ladder_Down_Position[4] = random.randint(54, (59 -
54) + 54)
        Play_With_friends.Ladder_Down_Position[5] = random.randint(68, (69 -
68) + 68)
        Play_With_friends.Ladder_Down_Position[6] = random.randint(72, (78 -
72) + 72)
        Play_With_friends.Ladder_Down_Position[7] = random.randint(81, (86 -
81) + 81)
    # Ladder Up Position Setting
    Play_With_friends.Ladder_Up_Position[0] = random.randint(81, (87 - 81)
+ 81)
    Play_With_friends.Ladder_Up_Position[1] = random.randint(92, (96 - 92)
+ 92)
    Play_With_friends.Ladder_Up_Position[2] = random.randint(74, (78 - 74)
+ 74)
    Play_With_friends.Ladder_Up_Position[3] = random.randint(65, (69 - 65)
+ 65)
    Play_With_friends.Ladder_Up_Position[4] = random.randint(81, (87 - 81)
+ 81)
    Play_With_friends.Ladder_Up_Position[5] = random.randint(74, (78 - 74)
+ 74)
    Play_With_friends.Ladder_Up_Position[6] = random.randint(95, (96 - 95)
+ 95)
    Play_With_friends.Ladder_Up_Position[7] = random.randint(92, (99 - 92)
+ 92)
def SnakesAndLaddersPositionPrint(self) :
    print(" \n----->SNAKES POSITION | LADDERS POSITION <-----")
---- \n", end = "")
    speak("SNAKES POSITION and LADDERS POSITION")
    if (self.Level == '1') :
        i = 0
        while (i < 3) :
            if (i == 0) :
                print(f"
{Play_With_friends.Snake_Mouth_Position[i]}      to
{Play_With_friends.Snake_Tail_Position[i]}      |
{Play_With_friends.Ladder_Down_Position[i]}      to
{Play_With_friends.Ladder_Up_Position[i]}")
            else :
                print(f"
{Play_With_friends.Snake_Mouth_Position[i]}      to
{Play_With_friends.Snake_Tail_Position[i]}      |
{Play_With_friends.Ladder_Down_Position[i]}      to
{Play_With_friends.Ladder_Up_Position[i]}")
            i += 1
    elif(self.Level == '2') :
        i = 0
        while (i < 5) :
            if (i == 0) :
                print(f"
{Play_With_friends.Snake_Mouth_Position[i]}      to
{Play_With_friends.Snake_Tail_Position[i]}      |
{Play_With_friends.Ladder_Down_Position[i]}      to
{Play_With_friends.Ladder_Up_Position[i]}")
            elif(i == 4) :
                print(f"
{Play_With_friends.Snake_Mouth_Position[i]}      to

```

```

0{Play_With_friends.Snake_Tail_Position[i]}           | 
{Play_With_friends.Ladder_Down_Position[i]}          to
{Play_With_friends.Ladder_Up_Position[i]}")           |
else :
    print(f"
{Play_With_friends.Snake_Mouth_Position[i]}          to
{Play_With_friends.Snake_Tail_Position[i]}           |
{Play_With_friends.Ladder_Down_Position[i]}          to
{Play_With_friends.Ladder_Up_Position[i]}")           |
    i += 1
else :
    i = 0
    while (i < 8) :
        if (i == 0) :
            print(f"
{Play_With_friends.Snake_Mouth_Position[i]}          to
{Play_With_friends.Snake_Tail_Position[i]}           |
0{Play_With_friends.Ladder_Down_Position[i]}          to
{Play_With_friends.Ladder_Up_Position[i]}")           |
        elif(i == 7) :
            print(f"
{Play_With_friends.Snake_Mouth_Position[i]}          to
0{Play_With_friends.Snake_Tail_Position[i]}           |
{Play_With_friends.Ladder_Down_Position[i]}          to
{Play_With_friends.Ladder_Up_Position[i]}")           |
        else :
            print(f"
{Play_With_friends.Snake_Mouth_Position[i]}          to
{Play_With_friends.Snake_Tail_Position[i]}           |
{Play_With_friends.Ladder_Down_Position[i]}          to
{Play_With_friends.Ladder_Up_Position[i]}")           |
    i += 1
    print("-----\n", end = "")
def RollDice(self) :
    print("\n----->Roll The Dice<-----\n", end = "")
    speak("Roll The Dice")
    i = 0
    while (i < Play_With_friends.n) :
        print(f"\n          {Play_With_friends.Names[i]}'s Turn (Enter character "R" or "r" ) : ", end = "")
        speak(f"{Play_With_friends.Names[i]}'s Turn")
        speak("Enter Capital or Small R")
        Play_With_friends.rollDice = input()
        while (Play_With_friends.rollDice != 'R' and Play_With_friends.rollDice != 'r') :
            print("\n          Enter Valid Character R or r : ")
            speak("Enter Valid Capital or Small R")
            Play_With_friends.rollDice = input()
        Play_With_friends.diceScore[i] = random.randint(1, (6 - 1) + 1)
        Play_With_friends.playerPosition[i] += Play_With_friends.diceScore[i]
        print(f"          dice Score = {Play_With_friends.diceScore[i]}")
        speak(f"dice Score = {Play_With_friends.diceScore[i]}")
        I = 0
        while (I < Play_With_friends.n) :

```

```

        k = 0
        while (k < 8) :
            if (self.Level == '1' and I < 3) :
                if (Play_With_friends.playerPosition[I] ==
Play_With_friends.Snake_Mouth_Position[k] and (Play_With_friends.playerPosition[I] ==
!= 0 and Play_With_friends.Snake_Mouth_Position[k] != 0)) :
                    print(" \n      ----->Oops, Swallowed By Snake<--"
-----")
                    speak("Oops, Swallowed By Snake")
                elif(Play_With_friends.playerPosition[I] ==
Play_With_friends.Ladder_Down_Position[k] and (Play_With_friends.playerPosition[I] ==
!= 0 and Play_With_friends.Ladder_Down_Position[k] != 0)) :
                    print(" \n      ----->Hurry, Climb Up The
Ladder<----")
                    speak("Hurry, Climb Up The Ladder")
                elif(self.Level == '2' and I < 5) :
                    if (Play_With_friends.playerPosition[I] ==
Play_With_friends.Snake_Mouth_Position[k] and (Play_With_friends.playerPosition[I] ==
!= 0 and Play_With_friends.Snake_Mouth_Position[k] != 0)) :
                        print(" \n      ----->Oops, Swallowed By Snake<--"
-----")
                        speak("Oops, Swallowed By Snake")
                    elif(Play_With_friends.playerPosition[I] ==
Play_With_friends.Ladder_Down_Position[k] and (Play_With_friends.playerPosition[I] ==
!= 0 and Play_With_friends.Ladder_Down_Position[k] != 0)) :
                        print(" \n      ----->Hurry, Climb Up The
Ladder<----")
                        speak("Hurry, Climb Up The Ladder")
                else :
                    if (Play_With_friends.playerPosition[I] ==
Play_With_friends.Snake_Mouth_Position[k] and (Play_With_friends.playerPosition[I] ==
!= 0 and Play_With_friends.Snake_Mouth_Position[k] != 0)) :
                        print(" \n      ----->Oops, Swallowed By Snake<--"
-----")
                        speak("Oops, Swallowed By Snake")
                    elif(Play_With_friends.playerPosition[I] ==
Play_With_friends.Ladder_Down_Position[k] and (Play_With_friends.playerPosition[I] ==
!= 0 and Play_With_friends.Ladder_Down_Position[k] != 0)) :
                        print(" \n      ----->Hurry, Climb Up The
Ladder<----")
                        speak("Hurry, Climb Up The Ladder")
            speak("Hurry, Climb Up The Ladder")
            k += 1
            I += 1
            if (self.flag != 1) :
                self.PlayerPosition()
                self.IsWon()
            i += 1
        def PlayerPosition(self) :
            i = 0
            while (i < Play_With_friends.n) :
                if (Play_With_friends.playerPosition[i] > Play_With_friends.WONPOINT) :
                    Play_With_friends.playerPosition[i] ==
Play_With_friends.diceScore[i]
                    i += 1
                # playerPosition With Snake_Tail_Position
                i = 0
                while (i < Play_With_friends.n) :

```

```

        if (self.Level == '1') :
            if (Play_With_friends.playerPosition[i] ==
Play_With_friends.Snake_Mouth_Position[0]) :
                Play_With_friends.playerPosition[i] =
Play_With_friends.Snake_Tail_Position[0]
            elif(Play_With_friends.playerPosition[i] ==
Play_With_friends.Snake_Mouth_Position[1]) :
                Play_With_friends.playerPosition[i] =
Play_With_friends.Snake_Tail_Position[1]
            elif(Play_With_friends.playerPosition[i] ==
Play_With_friends.Snake_Mouth_Position[2]) :
                Play_With_friends.playerPosition[i] =
Play_With_friends.Snake_Tail_Position[2]
            elif(self.Level == '2') :
                if (Play_With_friends.playerPosition[i] ==
Play_With_friends.Snake_Mouth_Position[0]) :
                    Play_With_friends.playerPosition[i] =
Play_With_friends.Snake_Tail_Position[0]
                elif(Play_With_friends.playerPosition[i] ==
Play_With_friends.Snake_Mouth_Position[1]) :
                    Play_With_friends.playerPosition[i] =
Play_With_friends.Snake_Tail_Position[1]
                elif(Play_With_friends.playerPosition[i] ==
Play_With_friends.Snake_Mouth_Position[2]) :
                    Play_With_friends.playerPosition[i] =
Play_With_friends.Snake_Tail_Position[2]
                elif(Play_With_friends.playerPosition[i] ==
Play_With_friends.Snake_Mouth_Position[3]) :
                    Play_With_friends.playerPosition[i] =
Play_With_friends.Snake_Tail_Position[3]
                elif(Play_With_friends.playerPosition[i] ==
Play_With_friends.Snake_Mouth_Position[4]) :
                    Play_With_friends.playerPosition[i] =
Play_With_friends.Snake_Tail_Position[4]
            else :
                pass
        else :
            if (Play_With_friends.playerPosition[i] ==
Play_With_friends.Snake_Mouth_Position[0]) :
                Play_With_friends.playerPosition[i] =
Play_With_friends.Snake_Tail_Position[0]
            elif(Play_With_friends.playerPosition[i] ==
Play_With_friends.Snake_Mouth_Position[1]) :
                Play_With_friends.playerPosition[i] =
Play_With_friends.Snake_Tail_Position[1]
            elif(Play_With_friends.playerPosition[i] ==
Play_With_friends.Snake_Mouth_Position[2]) :
                Play_With_friends.playerPosition[i] =
Play_With_friends.Snake_Tail_Position[2]
            elif(Play_With_friends.playerPosition[i] ==
Play_With_friends.Snake_Mouth_Position[3]) :
                Play_With_friends.playerPosition[i] =
Play_With_friends.Snake_Tail_Position[3]
            elif(Play_With_friends.playerPosition[i] ==
Play_With_friends.Snake_Mouth_Position[4]) :
                Play_With_friends.playerPosition[i] =
Play_With_friends.Snake_Tail_Position[4]

```

```

        elif(Play_With_friends.playerPosition[i] ==
Play_With_friends.Snake_Mouth_Position[5]) :
            Play_With_friends.playerPosition[i] =
Play_With_friends.Snake_Tail_Position[5]
        elif(Play_With_friends.playerPosition[i] ==
Play_With_friends.Snake_Mouth_Position[6]) :
            Play_With_friends.playerPosition[i] =
Play_With_friends.Snake_Tail_Position[6]
        elif(Play_With_friends.playerPosition[i] ==
Play_With_friends.Snake_Mouth_Position[7]) :
            Play_With_friends.playerPosition[i] =
Play_With_friends.Snake_Tail_Position[7]
    i += 1
    # playerPosition With Ladder
    i = 0
    while (i < Play_With_friends.n) :
        if (self.Level == '1') :
            if (Play_With_friends.playerPosition[i] ==
Play_With_friends.Ladder_Down_Position[0]) :
                Play_With_friends.playerPosition[i] =
Play_With_friends.Ladder_Up_Position[0]
            elif(Play_With_friends.playerPosition[i] ==
Play_With_friends.Ladder_Down_Position[1]) :
                Play_With_friends.playerPosition[i] =
Play_With_friends.Ladder_Up_Position[1]
            elif(Play_With_friends.playerPosition[i] ==
Play_With_friends.Ladder_Down_Position[2]) :
                Play_With_friends.playerPosition[i] =
Play_With_friends.Ladder_Up_Position[2]
            elif(self.Level == '2') :
                if (Play_With_friends.playerPosition[i] ==
Play_With_friends.Ladder_Down_Position[0]) :
                    Play_With_friends.playerPosition[i] =
Play_With_friends.Ladder_Up_Position[0]
                elif(Play_With_friends.playerPosition[i] ==
Play_With_friends.Ladder_Down_Position[1]) :
                    Play_With_friends.playerPosition[i] =
Play_With_friends.Ladder_Up_Position[1]
                elif(Play_With_friends.playerPosition[i] ==
Play_With_friends.Ladder_Down_Position[2]) :
                    Play_With_friends.playerPosition[i] =
Play_With_friends.Ladder_Up_Position[2]
                elif(Play_With_friends.playerPosition[i] ==
Play_With_friends.Ladder_Down_Position[3]) :
                    Play_With_friends.playerPosition[i] =
Play_With_friends.Ladder_Up_Position[3]
                elif(Play_With_friends.playerPosition[i] ==
Play_With_friends.Ladder_Down_Position[4]) :
                    Play_With_friends.playerPosition[i] =
Play_With_friends.Ladder_Up_Position[4]
            else :
                if (Play_With_friends.playerPosition[i] ==
Play_With_friends.Ladder_Down_Position[0]) :
                    Play_With_friends.playerPosition[i] =
Play_With_friends.Ladder_Up_Position[0]
                elif(Play_With_friends.playerPosition[i] ==
Play_With_friends.Ladder_Down_Position[1]) :

```

```

        Play_With_friends.playerPosition[i] =
Play_With_friends.Ladder_Up_Position[1]
            elif(Play_With_friends.playerPosition[i] ==
Play_With_friends.Ladder_Down_Position[2]) :
                Play_With_friends.playerPosition[i] =
Play_With_friends.Ladder_Up_Position[2]
                    elif(Play_With_friends.playerPosition[i] ==
Play_With_friends.Ladder_Down_Position[3]) :
                        Play_With_friends.playerPosition[i] =
Play_With_friends.Ladder_Up_Position[3]
                            elif(Play_With_friends.playerPosition[i] ==
Play_With_friends.Ladder_Down_Position[4]) :
                                Play_With_friends.playerPosition[i] =
Play_With_friends.Ladder_Up_Position[4]
                                    elif(Play_With_friends.playerPosition[i] ==
Play_With_friends.Ladder_Down_Position[5]) :
                                        Play_With_friends.playerPosition[i] =
Play_With_friends.Ladder_Up_Position[5]
                                            elif(Play_With_friends.playerPosition[i] ==
Play_With_friends.Ladder_Down_Position[6]) :
                                                Play_With_friends.playerPosition[i] =
Play_With_friends.Ladder_Up_Position[6]
                                                    elif(Play_With_friends.playerPosition[i] ==
Play_With_friends.Ladder_Down_Position[7]) :
                                                        Play_With_friends.playerPosition[i] =
Play_With_friends.Ladder_Up_Position[7]
                                                            i += 1
# player position display
print("\n-----:::::Player\'s Position::::-----\n")
    speak("Player\'s Position")
    i = 0
    while (i < Play_With_friends.n) :
        self.Count = 1
        if (Play_With_friends.diceScore[i] > 0) :
            print(f"CURRENT : {Play_With_friends.Names[i]}")
= {Play_With_friends.playerPosition[i]}
            speak(f"CURRENT : {Play_With_friends.Names[i]}")
= {Play_With_friends.playerPosition[i]}
            else :
                print(f"CURRENT : {Play_With_friends.Names[i]}")
= {Play_With_friends.playerPosition[i]}
                speak(f"CURRENT : {Play_With_friends.Names[i]}")
= {Play_With_friends.playerPosition[i]}
                if (i == Play_With_friends.n - 1) :
                    print("-----", end ="")
                    self.Count += 1
                    Play_With_friends.diceScore[i] = 0
                    i += 1
def IsWon(self) :
    i = 0
    while (i < Play_With_friends.n) :
        if (Play_With_friends.playerPosition[i] == Play_With_friends.WONPOINT)
:
            print("\n-----WINNER-----\n")

```

```

                print(f"                                     CONGRATS
{Play_With_friends.Names[i]}")
                speak(f"CONGRATS {Play_With_friends.Names[i]} you won.")
                print("\n-----WINNER-----\n")
                self.flag = 1
                break
            if (self.flag == 1) :
                break
            i += 1
class SnakeNLadderGame :
    def main() :
        print("                                     Welcome To SNACK AND LADDER GAME \n-----\n")
        speak("Welcome To SNACK AND LADDER GAME")
        print("Application Type : Console Game")
        speak("Application Type : Console Game")
        print("Version : 1.0.1 LTS")
        speak("Version : 1.0.1 LTS")
        print("Developer : GHANSHYAM VAJA")
        speak("Developer : GHANSHYAM vaja")
        print("-----\n")
        playWith = ' '
        print("\n      1 - play With Computer      2 - play With friends")
        print(" \nEnter Your choice (1 | 2) : ", end ="")
        speak("Enter Your choice")
        speak("1 for play With Computer")
        speak("2 for play With friends")
        playWith = input()
        while (playWith != '1' and playWith != '2') :
            print(" \nEnter valid choice (1 | 2) : ", end ="")
            speak("Enter valid choice")
            speak("1 or 2")
            playWith = input()
        print(" \n::::::::::::::::::---->WINPOINT : 100<----:::::::::::\n\n")
        speak("WINPOINT 100")
        if (playWith == '1') :
            obj = Play_With_Computer()
            i = 1
            while (obj.flag != 1) :
                if (i == 1) :
                    obj.setGame()
                    obj.SnakesAndLaddersPosition()
                    obj.SnakesAndLaddersPositionPrint()
                    if (i == 1) :
                        print("\n-----:Lets Start The Game:-----\n", end ="")
                        speak("Lets Start The Game")
                    obj.RollDice()
                    # obj.PlayerPosition();
                    # obj.IsWon();
                    i = 2
                else :
                    obj2 = Play_With_friends()
                    k = 1

```



```

try:
    print("Recognizing.....")
    Status = tkinter.StringVar(value="Recognizing.....")
    query = r.recognize_google(audio, language='en-in')
    print(f"you said: {query} \n")
except Exception as e:
    print(e)
    print("Unable to Recognize your Voice.")
    return "none"

return query

#####
#####Commands#####
#####
def run_assistant():
    while True:
        os.system('cls')
        speak("Press 1 and Enter in command prompt to give command")
        Activate = input("Press 1 and Enter key : ")
        while Activate != '1':
            speak("Press 1 and Enter in command prompt to give command")
            Activate = input("Press 1 and Enter key : ")
        query = takeCommand().lower()
        host = socket.gethostname()
        IP = socket.gethostbyname(host)
        DateTime = datetime.datetime.now()
        TName = Email.replace('@', '_')
        TName = TName.replace('.', '_')
        sql = f"Insert into {TName} (Query, Host, IPAddress, DateAndTime)VALUES (%s, %s, %s, %s)"
        val = (query, host, IP, DateTime)
        mycursor.execute(sql, val)
        myconn.commit()

        if "none" in query:
            pass

        elif 'how are you' in query:
            speak("I am fine sir.. And you?")
            query = takeCommand().lower()
            if query == "i am fine" or query == "fine":
                speak("nice to hear this")
                speak("how can i help you sir?")

        elif 'what is your name' in query:
            speak("My name is G 1 point o")

        elif 'change my name' in query:
            global User
            speak("Enter your new name in command prompt")
            OldName = User
            speak("Enter your new name in command prompt")
            NewName = input("Enter your new Name : ")
            while NewName == '':
                NewName = input("Enter your new Name : ")

```

```

mycursor.execute(f"UPDATE Users SET Name='{NewName}' WHERE
Email='{Email}'")
speak(f"Hi, {OldName} your name is changed Successfully.")
speak(f"your new name is {NewName}")
myconn.commit()
User = NewName
break

elif 'change my password' in query or 'change account password' in query:
    speak("Enter your new password in command prompt")
    NewPassword = input("Enter New Password : ")
    res = re.search(PasswordValidate, NewPassword)
    while res == None:
        print("\n Warning : \nMinimum password length is needs required
more than 8 character")
        print("password can be combination of uppercase latters, lowercase
latters, numbers and special characters")
        speak("Warning")
        speak("Minimum password length is nneds too be required more than 8
character")
        speak("password can be combination of uppercase latters, lowercase
latters, numbers and special characters")
        speak("Enter password in valid format ")
        NewPassword = input("Enter password in valid format : ")
        res = re.search(PasswordValidate, NewPassword)
    mycursor.execute(f"UPDATE Users SET Password='{NewPassword}' WHERE
Email='{Email}'")
    speak(f"Hi, {User} your Password is changed Successfully.")
    myconn.commit()
    break

elif 'delete my account' in query:
    speak("Enter your Password in command prompt")
    password = input("Enter your Password : ")
    sql = f"SELECT *FROM Users WHERE Email = '{Email}' and Password =
'{password}'"
    mycursor.execute(sql)
    checkRecord = mycursor.fetchone()
    while checkRecord is None:
        speak("Enter your Password in command prompt")
        password = input("Enter your Password : ")
        sql = f"SELECT *FROM Users WHERE Email = '{Email}' and Password =
'{password}'"
        mycursor.execute(sql)
        checkRecord = mycursor.fetchone()
    sql2 = f"DELETE FROM Users WHERE Email = '{Email}'"
    sql3 = f"DROP TABLE {TName}"
    mycursor.execute(sql2)
    if mycursor.rowcount:
        speak(f"Hi, {User} your account is deleted successfully.....")
    myconn.commit()
    mycursor.execute(sql3)
    os.system('cls')
    speak("Press 1 for Login")
    speak("Press 2 For Signup")
    speak("Press 3 For Delete Account")
    print("\nPress 1 for Login")

```

```

print("Press 2 For Signup")
print("Press 3 For Delete Account")
speak("Enter your choice in command prompt")
choice = input("Enter your Choice (1 or 2 or 3) : ")
while choice != '1' and choice != '2' and choice != '3':
    speak("Enter valid choice")
    choice = input("Enter valid choice : ")
if choice == '1':
    Login()
elif choice == '2':
    Signup()
else:
    DeleteAccount()
break

elif 'what is my name' in query or 'who i am' in query or 'who am i' in
query or 'what\s my name' in query or 'my name' in query:
    speak(f"Sir you name is {User}")
    break

elif 'who made you' in query or 'who created you' in query:
    speak(f'I have been created by ghanshyam vaja')
    break

elif 'who owns you' in query:
    speak(f'Ghanshyam vaja owns me')
    break

elif 'ghanshyam bca sem5 project' in query or 'ghanshyam bca sem 5 project'
in query or 'ghanshyam bca 5 project' in query or 'developer bca sem5 project' in
query or 'developer bca sem 5 project' in query or 'developer bca 5 project' in
query:
    speak("wait for new moments i am openinng Ghanshyam vaja's bca sem 5
project for you")
    webbrowser.open("savjani-college.000webhostapp.com")
    break

elif 'change voice' in query or 'change your voice' in query:
    speak("ok sir, i am changing my voice")
    engine.setProperty('voice', voices[1].id)
    speak("hi sir, this is me your assisstant How can i help you sir?")

elif 'ip address' in query or 'ipaddress' in query or 'what\s my ip' in
query or 'what is my ip address' in query:
    host = socket.gethostname()
    speak(f"your computer name is {host}")
    print(f"your computer ip address is : {socket.gethostbyname(host)}")
    speak(f"and your computer ip address is {socket.gethostbyname(host)}")
    break

elif "open cmd" in query or "command prompt" in query:
    os.system("start cmd")
    speak("I am starting Command Prompt in next few moments..")
    break

elif 'number guessing game' in query or 'play number guessing game' in
query:

```

```

        speak("Sir, now focus on command prompt to play game")
        GuessNumber()
        break

    elif 'snake and ladder game' in query or 'play snake and ladder game' in
query:
        speak("Sir, now focus on command prompt to play game")
        SnakeNLadderGame.main()
        break

    elif 'factorial' in query or 'find factorial' in query:
        value = re.findall(r"[-+]?\\d*\\.\\d+|\\d+", query)
        print(f"factorial of {int(value[0])} is
{math.factorial(int(value[0]))}")
        speak(f"factorial of {int(value[0])} is
{math.factorial(int(value[0]))}")
        break

    elif 'find square root' in query or 'find squareroot' in query or
'squareroot' in query or 'square root' in query:
        value = re.findall(r"[-+]?\\d*\\.\\d+|\\d+", query)
        print(f"square root of {float(value[0])} is
{math.sqrt(float(value[0]))}")
        speak(f"square root of {float(value[0])} is
{math.sqrt(float(value[0]))}")
        break

    elif 'to the power of' in query or 'power of' in query:
        values = re.findall(r"[-+]?\\d*\\.\\d+|\\d+", query)
        print(f"{float(values[0])} to the power of {float(values[1])} is
{math.pow(float(values[0]), float(values[1]))}")
        speak(f"{float(values[0])} to the power of {float(values[1])} is
{math.pow(float(values[0]), float(values[1]))}")
        break

    elif 'value of pi' in query:
        print(f"the value of pi is {math.pi}")
        speak(f"the value of pi is {math.pi}")
        break

    elif 'gcd of' in query or 'gcd' in query:
        values = [int(i) for i in query.split() if i.isdigit()]
        print(f"the gcd of {values[0]} and {values[1]} is {math.gcd(values[1],
values[0])}")
        speak(f"the gcd of {values[0]} and {values[1]} is {math.gcd(values[1],
values[0])}")
        break

    elif 'log of' in query or 'find log' in query:
        values = re.findall(r"[-+]?\\d*\\.\\d+|\\d+", query)
        print(f"the value of log {float(values[0])} with base
{float(values[1])} is {math.log(float(values[0]), float(values[1]))}")
        speak(f"the value of log {float(values[0])} with base
{float(values[1])} is {math.log(float(values[0]), float(values[1]))}")
        break

```

```

        elif 'log2 of' in query or 'log 2 of' in query or 'find log2' in query or
'find log 2' in query:
            query = query.replace('log2', '')
            query = query.replace('log 2', '')
            value = re.findall(r"[-+]?[d*]\.\d+|\d+", query)
            print(f"the value of log2 of {float(value[0])} is
{math.log2(float(value[0]))}")
            speak(f"the value of log2 of {float(value[0])} is
{math.log2(float(value[0]))}")
            break

        elif 'log 10 of' in query or 'log10 of' in query or 'find log 10' in query
or 'find log10' in query:
            query = query.replace('log 10', '')
            query = query.replace('log10', '')
            value = re.findall(r"[-+]?[d*]\.\d+|\d+", query)
            print(f"the value of log10 of {float(value[0])} is
{math.log10(float(value[0]))}")
            speak(f"the value of log10 of {float(value[0])} is
{math.log10(float(value[0]))}")
            break

        elif 'sin' in query or 'sin of' in query or 'find sin' in query or 'find
sin of' in query or 'sine' in query:
            value = re.findall(r"[-+]?[d*]\.\d+|\d+", query)
            print(f"the value of sine of {float(value[0])} is :
{math.sin(float(value[0]))}")
            speak(f"the value of sine of {float(value[0])} is
{math.sin(float(value[0]))}")
            break

        elif 'cos' in query or 'cos of' in query or 'find cos' in query or 'find
cos of' in query or 'cosine' in query:
            value = re.findall(r"[-+]?[d*]\.\d+|\d+", query)
            print(f"the value of cosine of {float(value[0])} is :
{math.cos(float(value[0]))}")
            speak(f"the value of cosine of {float(value[0])} is
{math.cos(float(value[0]))}")
            break

        elif 'tan' in query or 'tan of' in query or 'find tan' in query or 'find
tan of' in query or 'tangent' in query:
            value = re.findall(r"[-+]?[d*]\.\d+|\d+", query)
            print(f"the value of tangent of {float(value[0])} is :
{math.tan(float(value[0]))}")
            speak(f"the value of tangent of {float(value[0])} is
{math.tan(float(value[0]))}")
            break

        elif 'who is' in query:
            pwk.search(query)
            speak(f'Wait for few moments...')
            break

        elif 'what is' in query:
            pwk.search(query)
            speak(f'Wait for few moments...')

```

```

        break

    elif 'where is' in query:
        try:
            query = query.replace('where is', '')
            speak(f'Wait i am finding right location for you...')
            webbrowser.open("https://google.com/maps/place/" + query + '')
        except:
            speak("Place not found")
        break

    elif 'which is' in query or 'which' in query:
        pwk.search(query)
        speak(f'Wait for few moments...')
        break

    elif 'what is' in query:
        pwk.search(query)
        speak(f'Wait for few moments...')
        break

    elif 'the date' in query or 'date' in query or 'what is the date' in query
or 'what is the date today' in query:
        pwk.search(query)
        speak(f'Wait for few moments...')
        break

    elif 'the time' in query or 'time' in query or 'what is the time' in query
or 'what is the time right now' in query or 'current time' in query:
        pwk.search(query)
        speak(f'Wait for few moments...')
        break

    elif 'download youtube video' in query or 'youtube video download' in
query:
        speak("Paste video link in command prompt")
        url = input("Paste Link Here : ")
        yt = YouTube(url)

        print(f"title : {yt.title}")
        speak(f"title : {yt.title}")
        print(f"views : {yt.views}")
        speak(f"views : {yt.views}")

        ys = yt.streams.get_highest_resolution()

        print("Downloading Started.....")
        speak("Downloading Started")
        ys.download("")
        print("Download Completed.")
        speak("Download Commpleted sir")
        print("Path : C:/Users/Default/Downloads")
        break

    elif 'video' in query or 'play video' in query or 'play song on youtube' in
query:
        speak("Wait for few moments i am finding video for you....")

```

```

        query = query.replace("video", ' ')
        pwk.playonyt(query)
        break

    elif 'open google' in query or 'google' in query:
        speak("Wait for few moments i am opening google for you... \n")
        webbrowser.open("google.com")
        break

    elif 'open youtube' in query or 'youtube' in query:
        speak("Wait for few moments i am opening Youtube for you... \n")
        webbrowser.open("youtube.com")
        break

    elif 'open stack overflow' in query or 'stack overflow' in query:
        speak("Wait for few moments i am opening Stack Overflow for you... \n")
        webbrowser.open("stackoverflow.com")
        break

    elif 'apple' in query or 'open apple.com' in query or 'open apple' in
query:
        speak("Wait for few moments i am opening apple.com for you... \n")
        webbrowser.open("apple.com")
        break

    elif 'open whatsapp' in query or 'whatsapp' in query:
        speak("Wait for few moments i am opening whatsapp for you... \n")
        webbrowser.open("web.whatsapp.com")
        break

    elif 'open instagram' in query or 'instagram' in query:
        speak("Wait for few moments i am opening instagram for you... \n")
        webbrowser.open("instagram.com")
        break

    elif 'open twitter' in query or 'twitter' in query:
        speak("Wait for few moments i am opening Twitter for you... \n")
        webbrowser.open("twitter.com")
        break

    elif 'open spotify' in query or 'spotify' in query or 'play music' in query
or 'play song' in query or 'music' in query or 'song' in query:
        speak("Wait for few moments i am opening spotify for you... \n")
        webbrowser.open("spotify.com")
        break

    elif 'search' in query or 'find' in query:
        speak("wait for few moments i am finding best results for you...")
        query = query.replace('search', ' ')
        pwk.search(query)
        break

    elif 'wikipedia' in query or 'according to wikipedia' in query or 'search
in wikipedia' in query or 'tell me about' in query:
        speak("Wait for few moments I am Searching Wikipedia... \n")
        query = query.replace(("wikipedia"), "")
        query = query.replace('according to wikipedia', ' ')

```

```

try:
    result = wikipedia.summary(query, sentences = 29)
    speak("According to Wikipedia.... \n")
    print(result)
    speak(result)
except:
    print(f"Information not found About {query}")
    speak(f"Information not found About {query}")
    break

elif 'take photo' in query or 'take image' in query or 'camera' in query or
'take selfie' in query or 'open camera' in query:
    Name = f"IMG{random.randint(5, 9)}.jpg"
    ecapture.open(0, "G", Name)
    break

elif 'play joke' in query or 'tell me a joke' in query or 'speak a joke' in
query or 'make me happy' in query or 'joke' in query:
    speak(pyjokes.get_joke())
    speak("want to Listen one more??")
    speak("say yes or no")
    confirm = takeCommand().lower()
    if confirm == 'yes':
        speak(pyjokes.get_joke())
    break

elif 'lock' in query or 'lock my computer' in query or 'lock my system' in
query:
    speak("wait for few moments, i am doing lock your computer")
    ctypes.windll.user32.LockWorkStation()
    break

elif 'shutdown' in query or 'shutdown system' in query or 'shutdown my
computer' in query:
    speak("wait for few moment, i am doing process to do your system
shutting down")
    os.system("shutdown /s /t 1")

elif 'restart' in query or 'restart system' in query or 'restart my
computer' in query:
    speak("Wait for few moments, i am doing restart your computer")
    subprocess.call(["shutdown", "/r"])

elif 'terminate yourself' in query or 'deactivte yourself' in query or
'stop your self' in query or 'terminate your self' in query or 'deactivte your
self' in query or 'stop yourself' in query:
    speak("Ok sir, i am terninating myself in few moment")
    speak("3")
    speak("2")
    speak("1")
    myconn.close()
    sys.exit()

else:
    speak(f'Wait for few moments... ')
    pwk.search(query)

```

```

time.sleep(2.9)

#Main Function
if __name__ == '__main__':
    #####UI using tkinter & Driver
    #####Code#####
    root.title("G Assistent")
    root.iconbitmap("C:/Project2/Logo_Icon.ico")
    def a():
        clear = lambda: os.system('cls')
        version = "1 point 0"
        speak(f"Hi, I am G Assistent {version}")
        speak("Press 1 for Login")
        speak("Press 2 For Signup")
        speak("Press 3 For Delete Account")
        print("Press 1 for Login")
        print("Press 2 For Signup")
        print("Press 3 For Delete Account")
        speak("Enter your choice in command prompt")
        choice = input("Enter your Choice (1 or 2 or 3) : ")

        while choice != '1' and choice != '2' and choice != '3':
            speak("Enter valid choice")
            choice = input("Enter valid choice : ")

        if choice == '1':
            Login()
        elif choice == '2':
            Signup()
        else:
            DeleteAccount()
            speak(f"Hi, I am G Assistent {version}")
            speak("Press 1 for Login")
            speak("Press 2 For Signup")
            speak("Press 3 For Delete Account")
            print("\nPress 1 for Login")
            print("Press 2 For Signup")
            print("Press 3 For Delete Account")
            speak("Enter your choice in command prompt")
            choice = input("Enter your Choice (1 or 2 or 3) : ")
            while choice != '1' and choice != '2' and choice != '3':
                speak("Enter valid choice")
                choice = input("Enter valid choice : ")
            if choice == '1':
                Login()
            elif choice == '2':
                Signup()
            else:
                DeleteAccount()

        while True:
            run_assistent()

    image = Image.open("C:/Project2/Logo.jpg")
    photo = ImageTk.PhotoImage(image)
    LogoLabel = Label(image = photo)

```

```
LogoLabel.pack()
Start = tkinter.Button(root, text="Activate", font="sanserif 19", command=a)
#StatusInfo = Label(root, font="sen-serif 29", height="2", width="15",
textvariable=Status)
#StatusInfo.pack(pady=29)
Start.pack()
root.mainloop()
```

9. Spec file (main.spec)

The spec file **tells PyInstaller how to process your script**. It encodes the script names and most of the options you give to the pyinstaller command. The spec file is actually executable Python code. PyInstaller builds the app by executing the contents of the spec file. It contains information like Application Name, Application icon for Python Source code's Executable File(.Exe).

How to Generate spec file of python source code file?

1. Open Command Prompt
2. Install pyinstaller (**pip install pyinstaller**)
3. Run this command -> **pyinstaller path/main.py**

Or

pyinstaller main.py(if you opened command
Prompt in that folder where you have your python file(.py), so you
not need to specify path of python Source code file)

After run above Command you'll get your python Source code file's(.py) **spec file** in your located directory where your python file(.py) is available.

```
# -*- mode: python ; coding: utf-8 -*-

block_cipher = None

a = Analysis(
    ['GAssistant.py'],
    pathex=[],
    binaries=[],
    datas=[],
    hiddenimports=[],
    hookspath=[],
    hooksconfig={},
    runtime_hooks=[],
    excludes=[],
    win_no_prefer_redirects=False,
    win_private_assemblies=False,
    cipher=block_cipher,
    noarchive=False,
)
pyz = PYZ(a.pure, a.zipped_data, cipher=block_cipher)

exe = EXE(
```

```
pyz,  
a.scripts,  
a.binaries,  
a.zipfiles,  
a.datas,  
[],  
name='G Assistent console',  
debug=False,  
bootloader_ignore_signals=False,  
strip=False,  
upx=True,  
upx_exclude=[],  
runtime_tmpdir=None,  
console=True,  
icon='C:/Project2/Logo_Icon.ico',  
disable_windowed_traceback=False,  
argv_emulation=False,  
target_arch=None,  
codesign_identity=None,  
entitlements_file=None,  
)
```

10. Executable File(main.exe)

An **executable file (EXE file)** is a computer file that contains an encoded sequence of instructions that the system can execute directly when the user clicks the file icon. Executable files commonly have an EXE file extension, but there are hundreds of other executable file formats.

We can create exe file from **two ways in python** :

1. through python source code file(main.py)
2. through spec file(main.spec)

1. through python source code file(main.py) :

1. Open Command prompt
2. Install pyinstaller (**pip install pyinstaller**)
3. Run this Command :

pyinstaller -onefile path/main.py (create exe file with console)

Or

pyinstaller -onefile --noconsole path/main.py (create exe without console)

2. through spec file(main.spec) :

1. Open Command prompt
2. Install pyinstaller (**pip install pyinstaller**)
3. Run this Command -> **pyinstaller path/main.spec**(it will creates exe according to spec file configuration)

After that main.exe file will generated that will run on computers without python installation.

11. Supported Commands

Username = 38

Password = 38

For Ex. queryword = Sundar Pichai

How to give a command ? -> Who is

Sundar Pichai

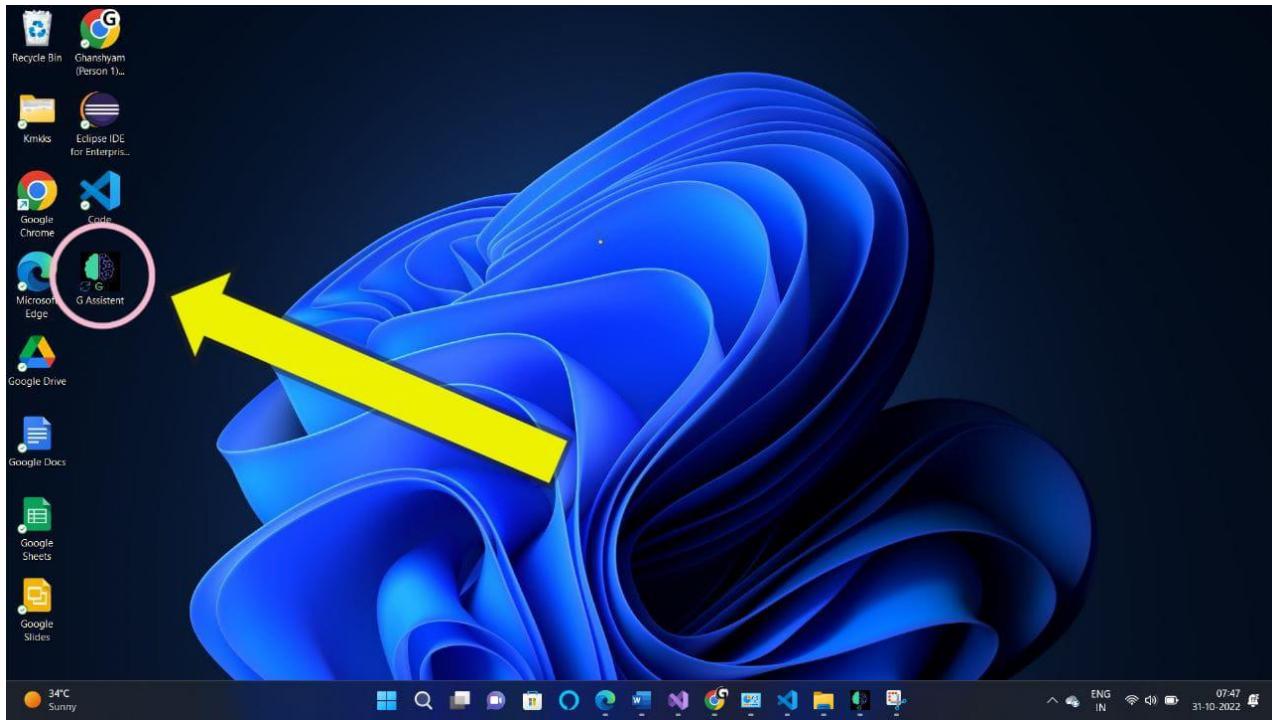
(command 21 : command format) (queryword)

1. 'how are you'
2. 'what is your name'
3. 'change my name'
4. 'change my password' **or** 'change account password'
5. 'delete my account'
6. 'what is my name' **or** 'who i am' or 'who am i' **or** 'what's my name' **or** 'my name'
7. 'who made you' **or** 'who created you'
8. 'who owns you'
9. 'ghanshyam bca sem5 project' **or** 'ghanshyam bca sem 5 project' **or** 'ghanshyam bca 5 project' **or** 'developer bca sem5 project' **or** 'developer bca sem 5 project' **or** 'developer bca 5 project'
10. 'change voice' **or** 'change your voice'
11. 'ip address' **or** 'ipaddress' **or** 'what's my ip' **or** 'what is my ip address'
12. "open cmd" **or** "command prompt"
13. 'number guessing game' **or** 'play number guessing game'
14. 'snake and ladder game' **or** 'play snake and ladder game'
15. ('factorial' or 'find factorial') + **queryword** → value
16. ('find square root' **or** 'find squareroot' **or** 'squareroot' **or** 'square root') + **queryword** → value
17. **x** value +('to the power of' **or** 'power of' in query) + **y** value → **x** and **y** values included in **queryword**
18. 'value of pi'
19. ('gcd of' **or** 'gcd') + **queryword** → values
20. ('log of' **or** 'find log') + **queryword** → value
21. ('log2 of' **or** 'log 2 of' **or** 'find log2' **or** 'find log 2') + **queryword** → value
22. ('log 10 of' **or** 'log10 of' **or** 'find log 10' **or** 'find log10') + **queryword** → value
23. ('sin' **or** 'sin of' **or** 'find sin' **or** 'find sin of' **or** 'sine') + **queryword** → value
24. ('cos' **or** 'cos of' **or** 'find cos' **or** 'find cos of' **or** 'cosine') + **queryword** → value
25. ('tan' **or** 'tan of' **or** 'find tan' **or** 'find tan of' **or** 'tangent') + **queryword** → value
26. 'who is' + **queryword** (Who is Sundar Pichai)
27. 'what is' + **queryword** (What is apple)
28. 'where is' + **queryword** (where is Taj Mehel)
29. 'which is' + **queryword** (which country has taj mehel)
30. 'what is' + **queryword** (what is apple)
31. 'the date' **or** 'what is the date' **or** 'what is the date today'
32. 'the time' **or** 'what is the time' **or** 'what is the time right now' **or** 'current time'
33. 'download youtube video' **or** 'youtube video download'
34. **queryword** + 'video' **or** 'play video' + **queryword** **or** **queryword** + 'play song on youtube'
35. 'open google' **or** 'google'
36. 'open youtube' **or** 'youtube'
37. 'open stack overflow' **or** 'stack overflow'

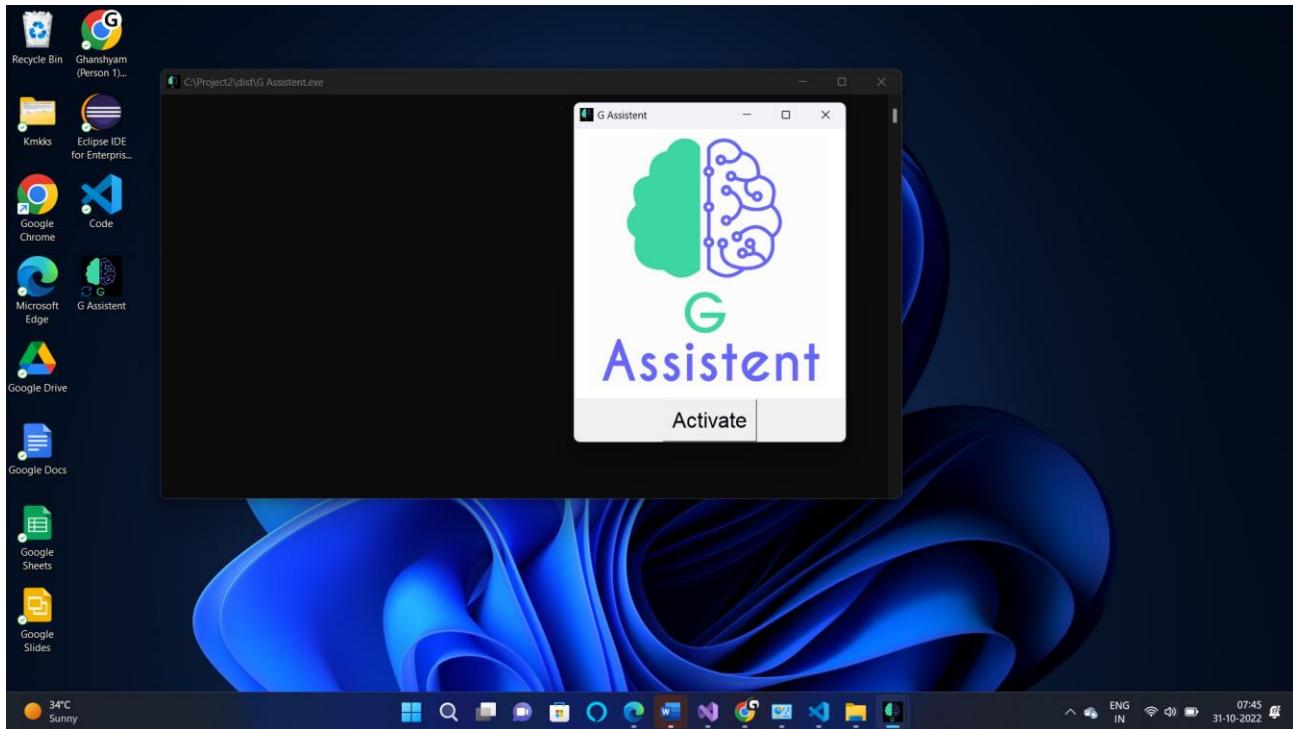
38. 'apple' **or** 'open apple.com' **or** 'open apple'
39. 'open whatsapp' **or** 'whatsapp'
40. 'open instagram' **or** 'instagram'
41. 'open twitter' **or** 'twitter'
42. 'open spotify' **or** 'spotify' **or** 'play music' **or** 'play song' **or** 'music' **or** 'song'
43. 'search' + **queryword** **or** 'find' + **queryword**
44. **queryword** + 'wikipedia' **or** **queryword** + 'according to wikipedia' **or** **queryword** + 'search in wikipedia'
45. 'take photo' **or** 'take image' **or** 'take image' **or** 'camera' **or** 'take selfie' **or** 'open camera'
46. 'play joke' **or** 'tell me a joke' **or** 'speak a joke' **or** 'make me happy' **or** 'joke':
47. 'lock' **or** 'lock my computer' **or** 'lock my system'
48. 'shutdown' **or** 'shutdown system' **or** 'shutdown my computer'
49. 'restart' **or** 'restart system' **or** 'restart my computer'
50. 'terminate yourself' **or** 'deactivte yourself' **or** 'stop yourself' **or** 'terminate your self' **or** 'stop your self' **or** 'terminate your self'
51. Any query will not match with above queries that will search on google

12. Application Screenshots

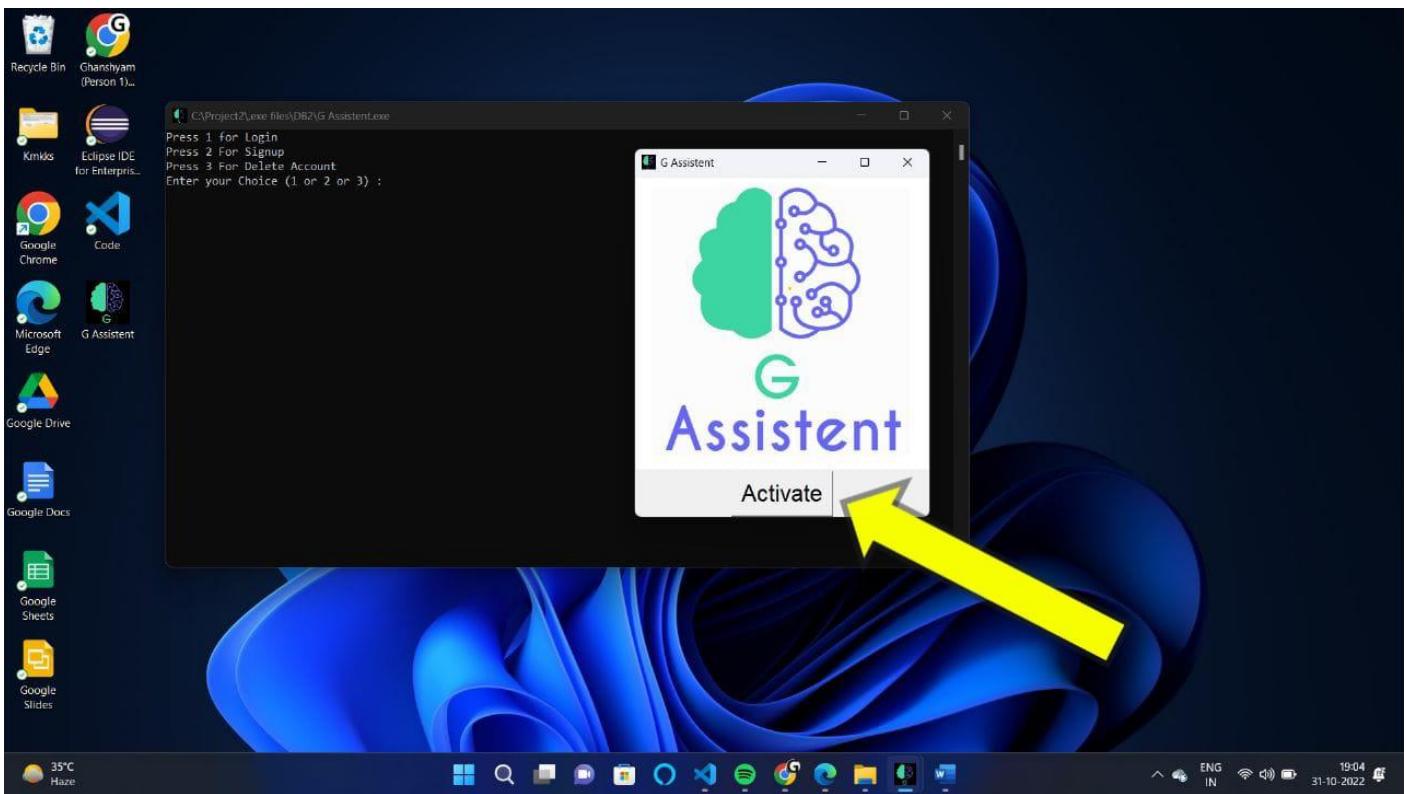
- Application (.exe)



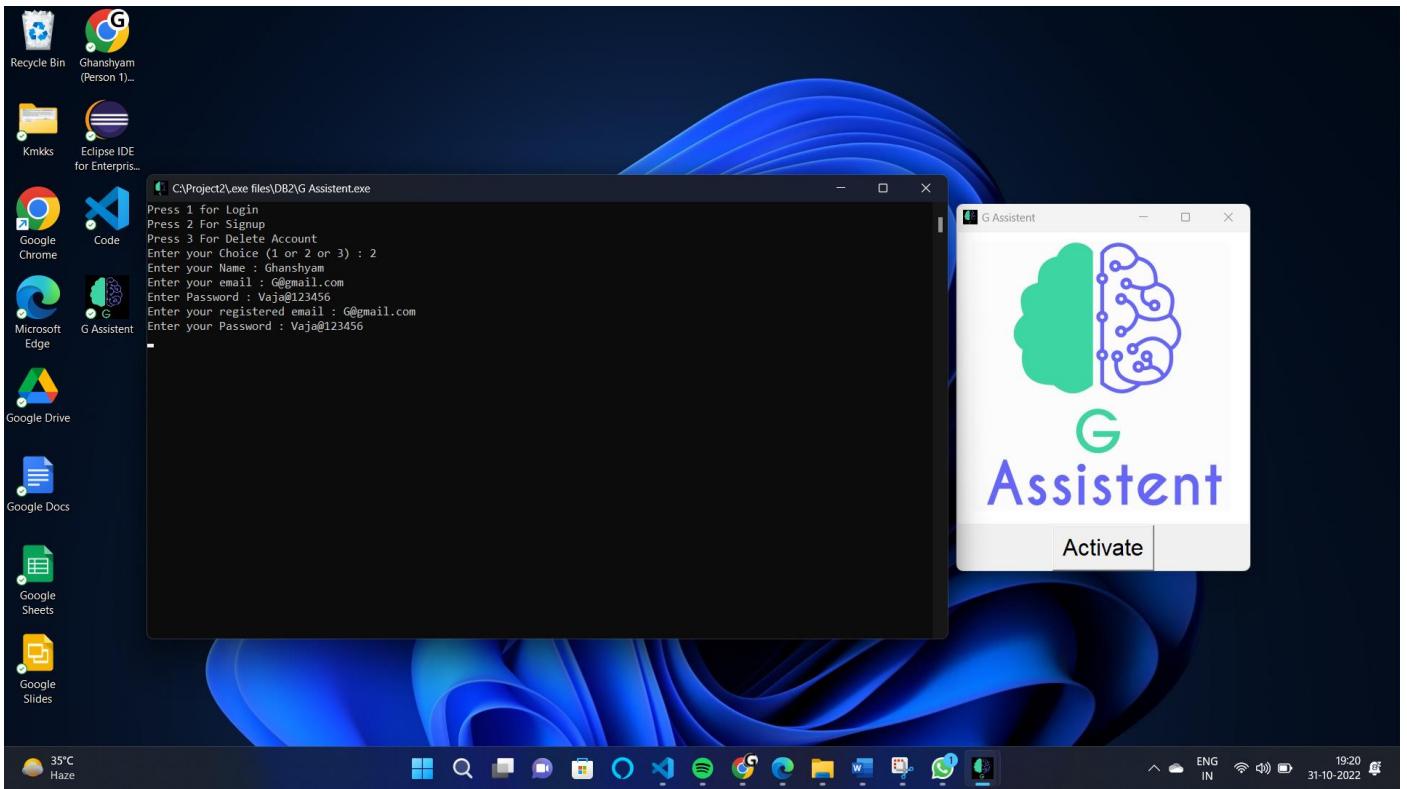
- Application Launch in Computer



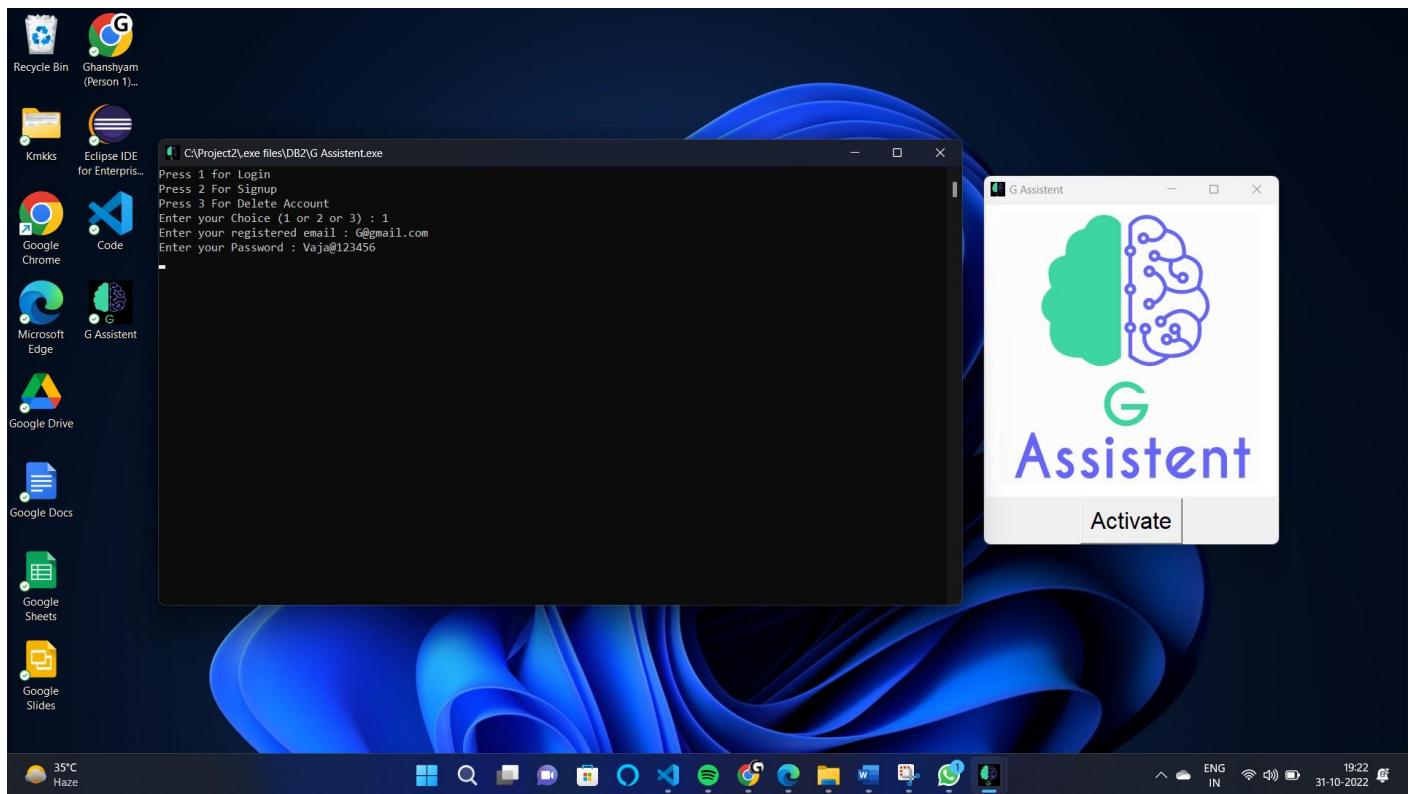
- Activate Application(after clicking on Activate Button)



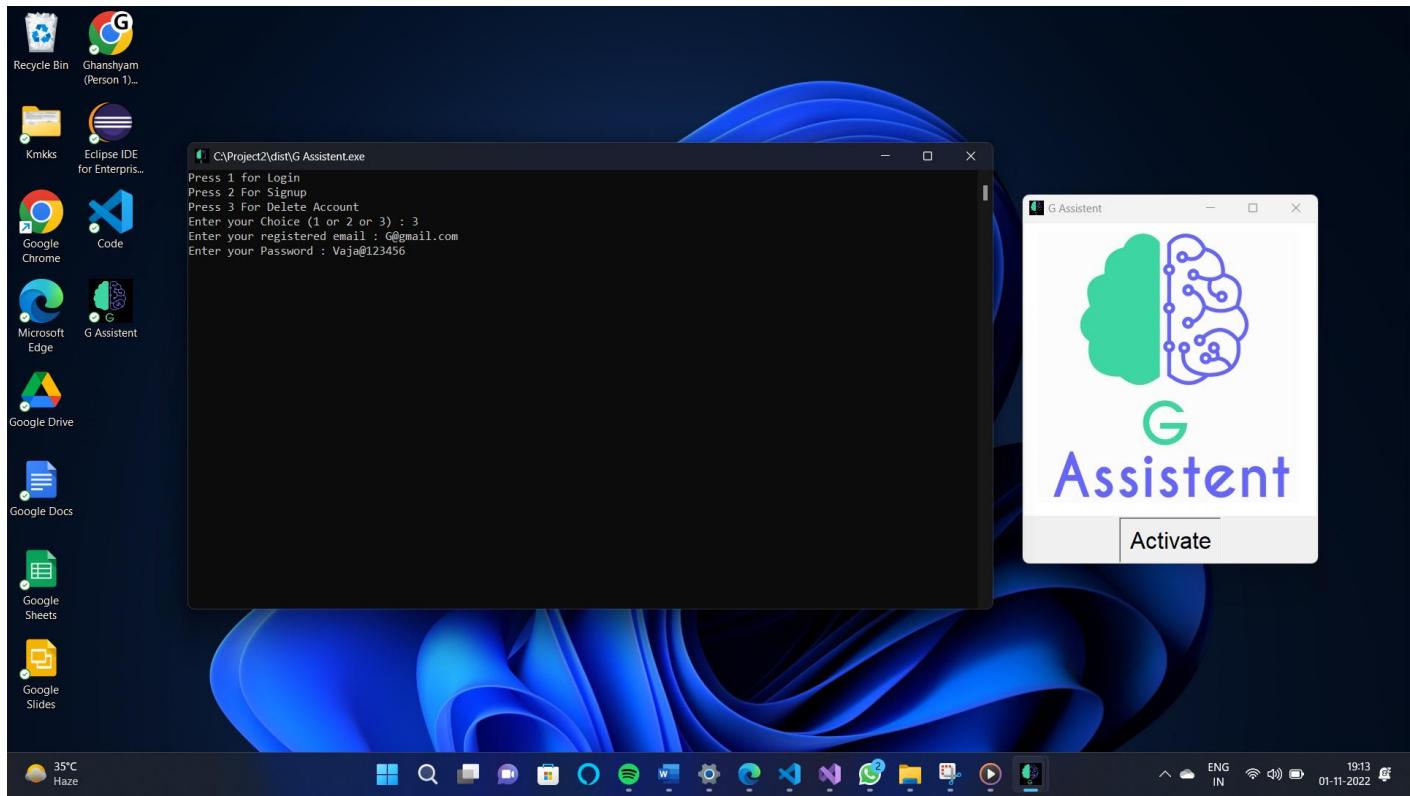
- Signup with Login



• Login

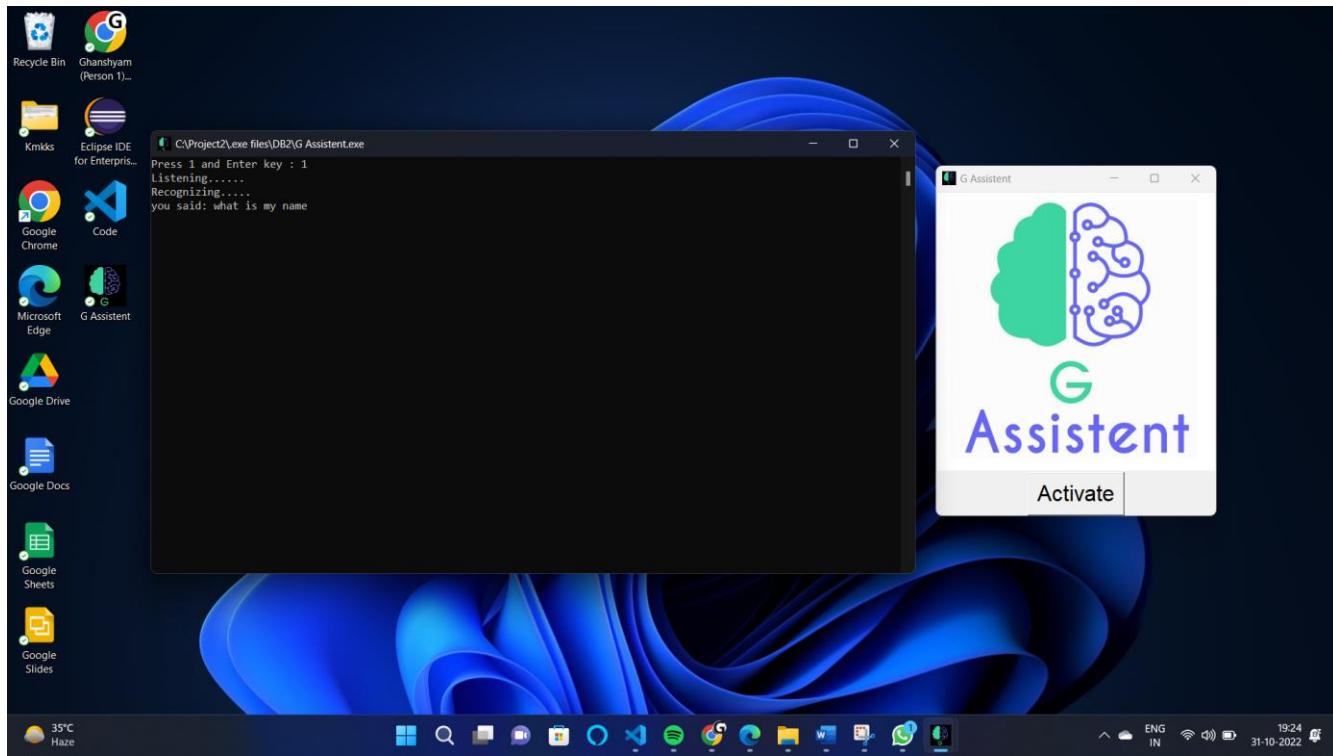


• Delete Account

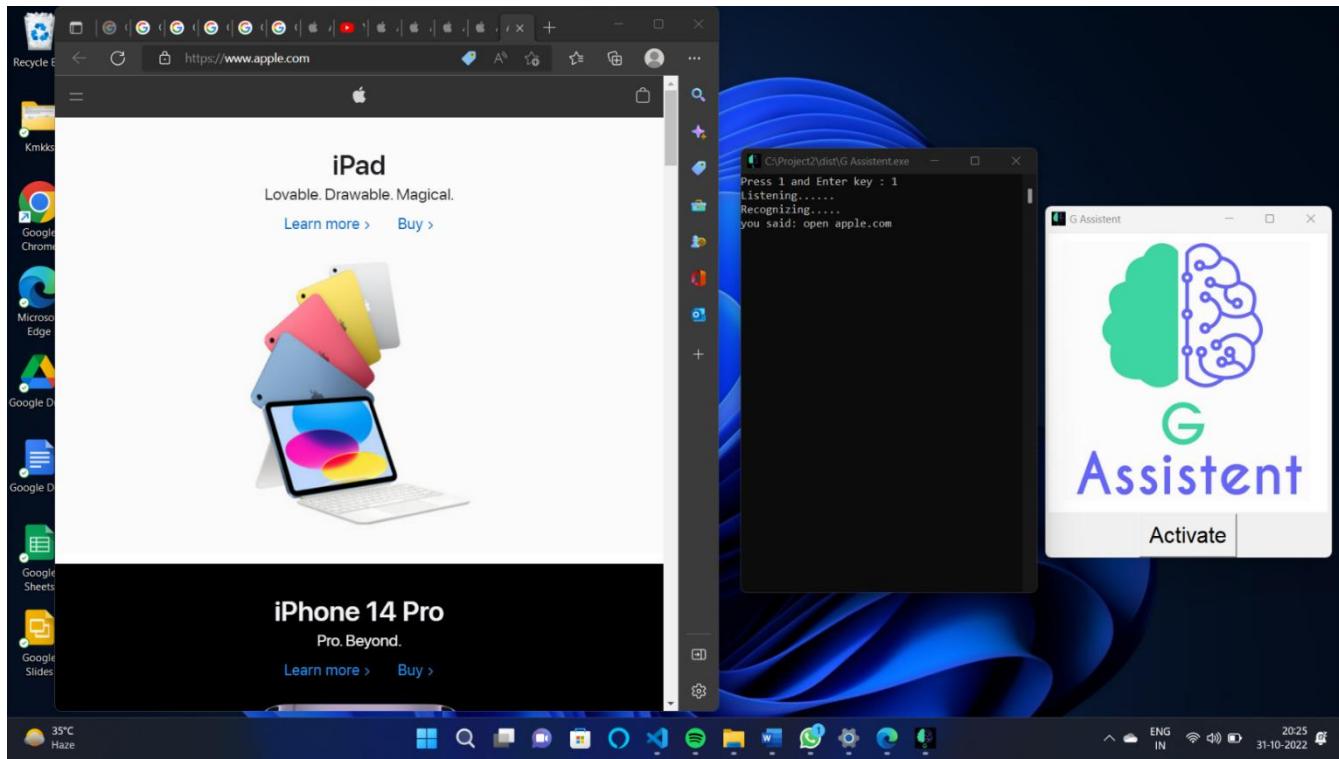


Some commands

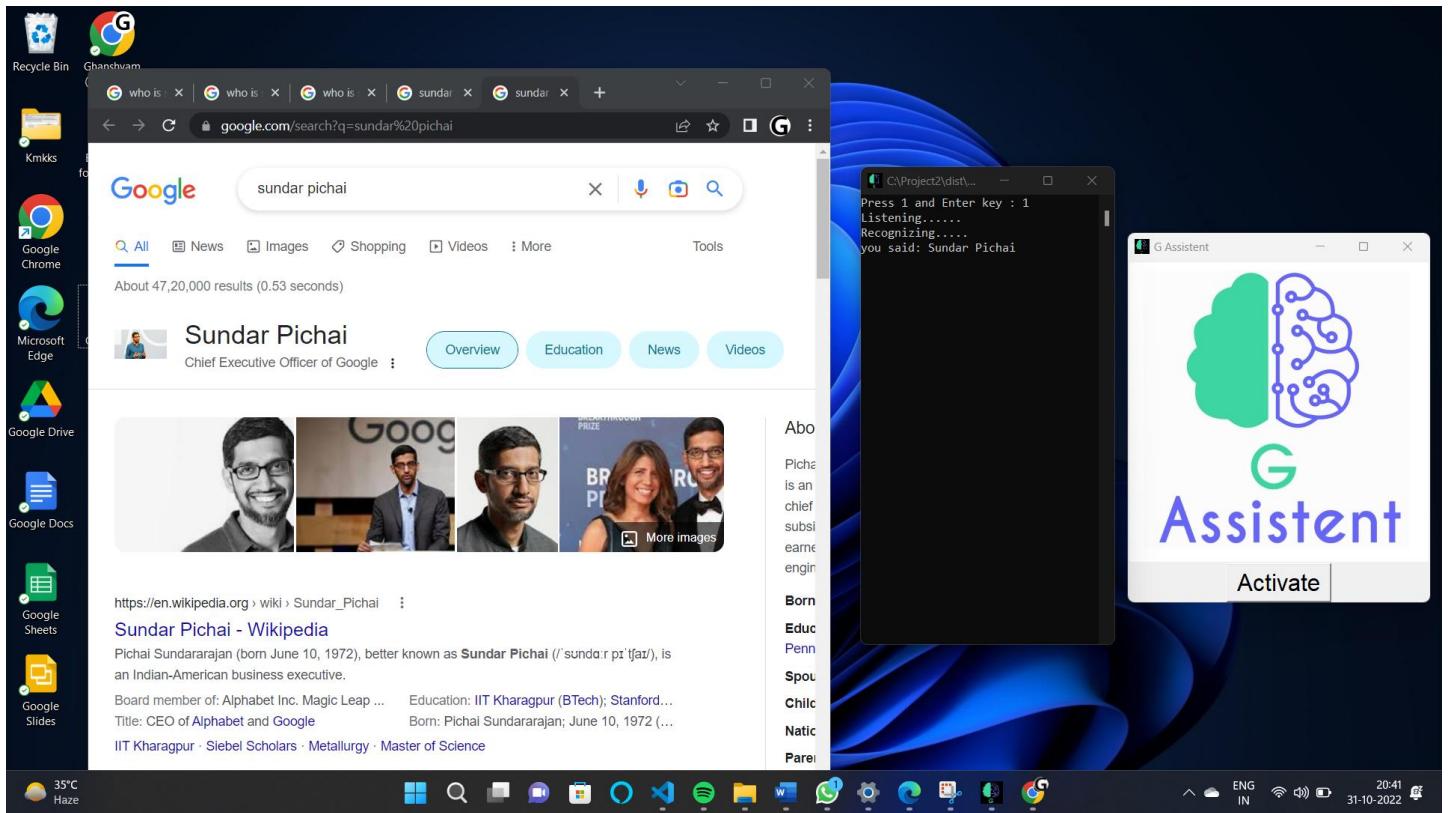
- what is my name



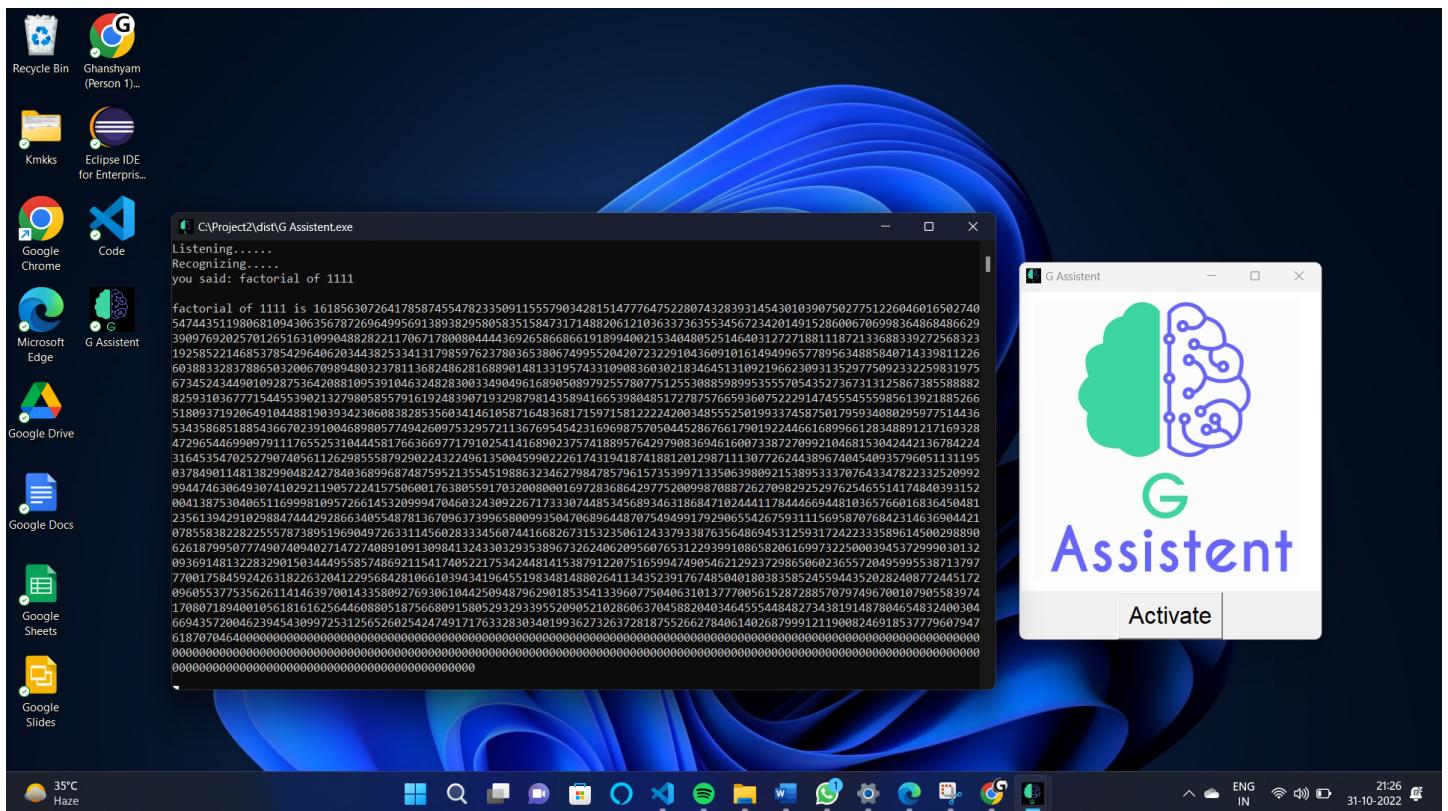
- open apple.com



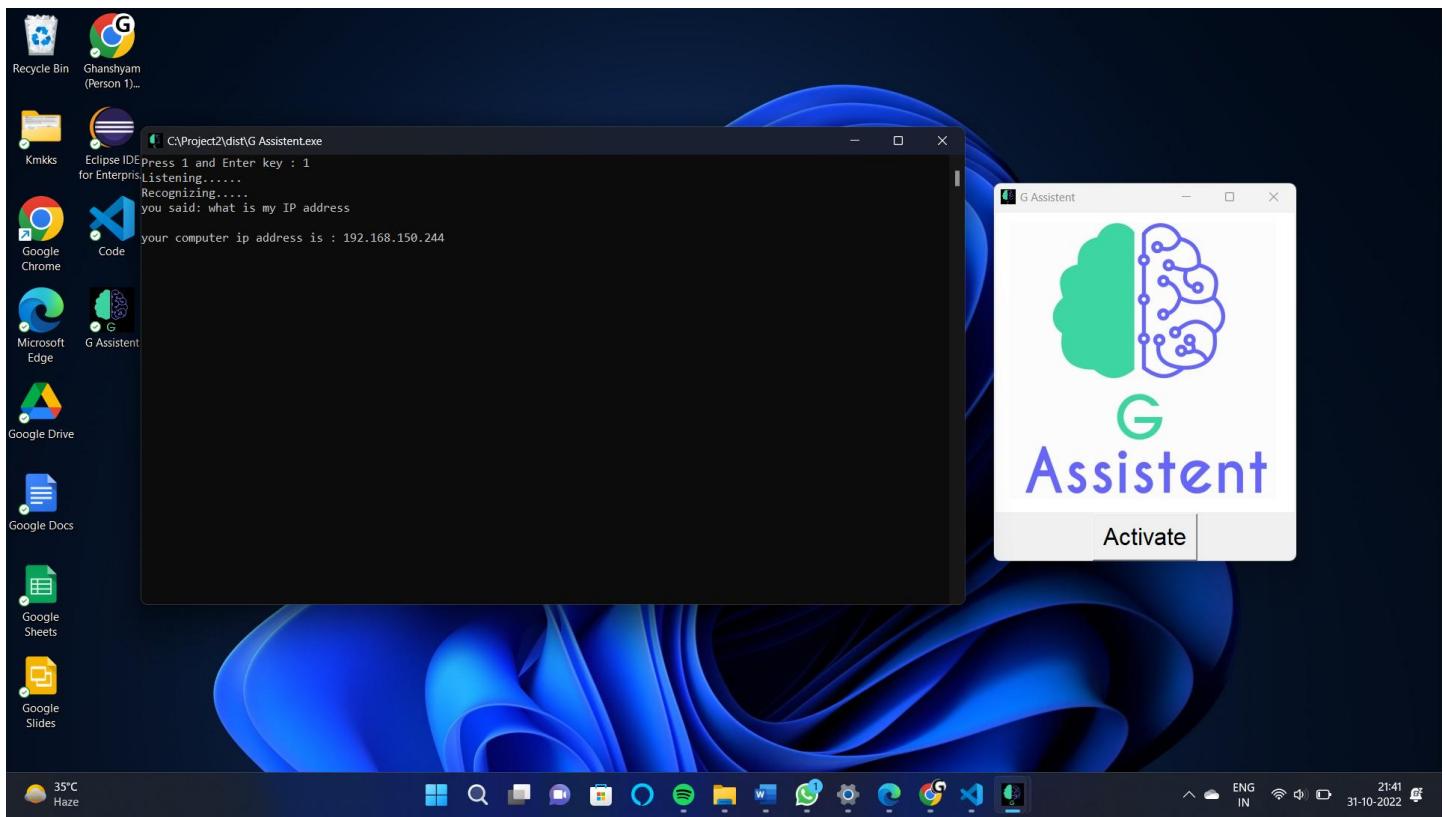
- Sundar Pichai



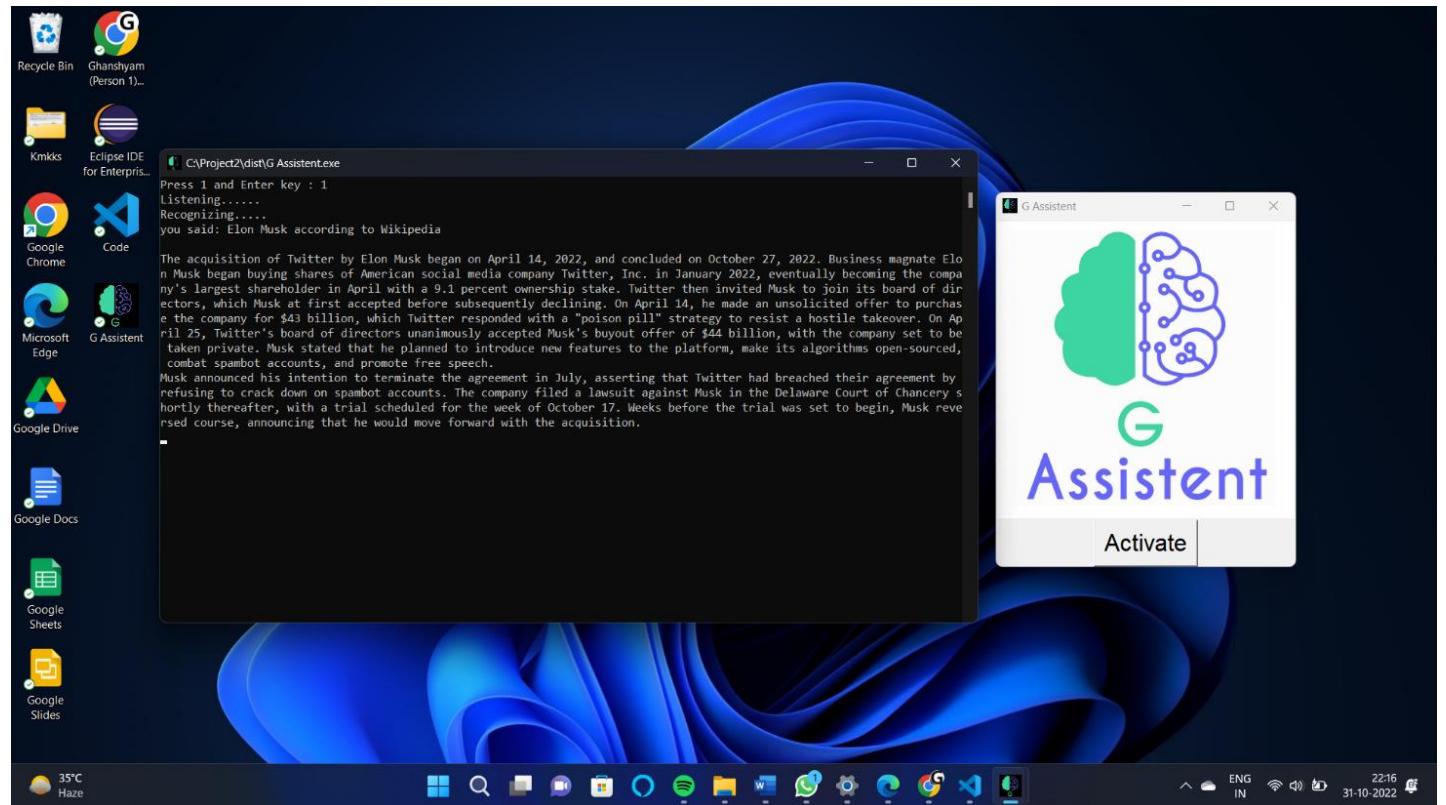
- factorial of 1111



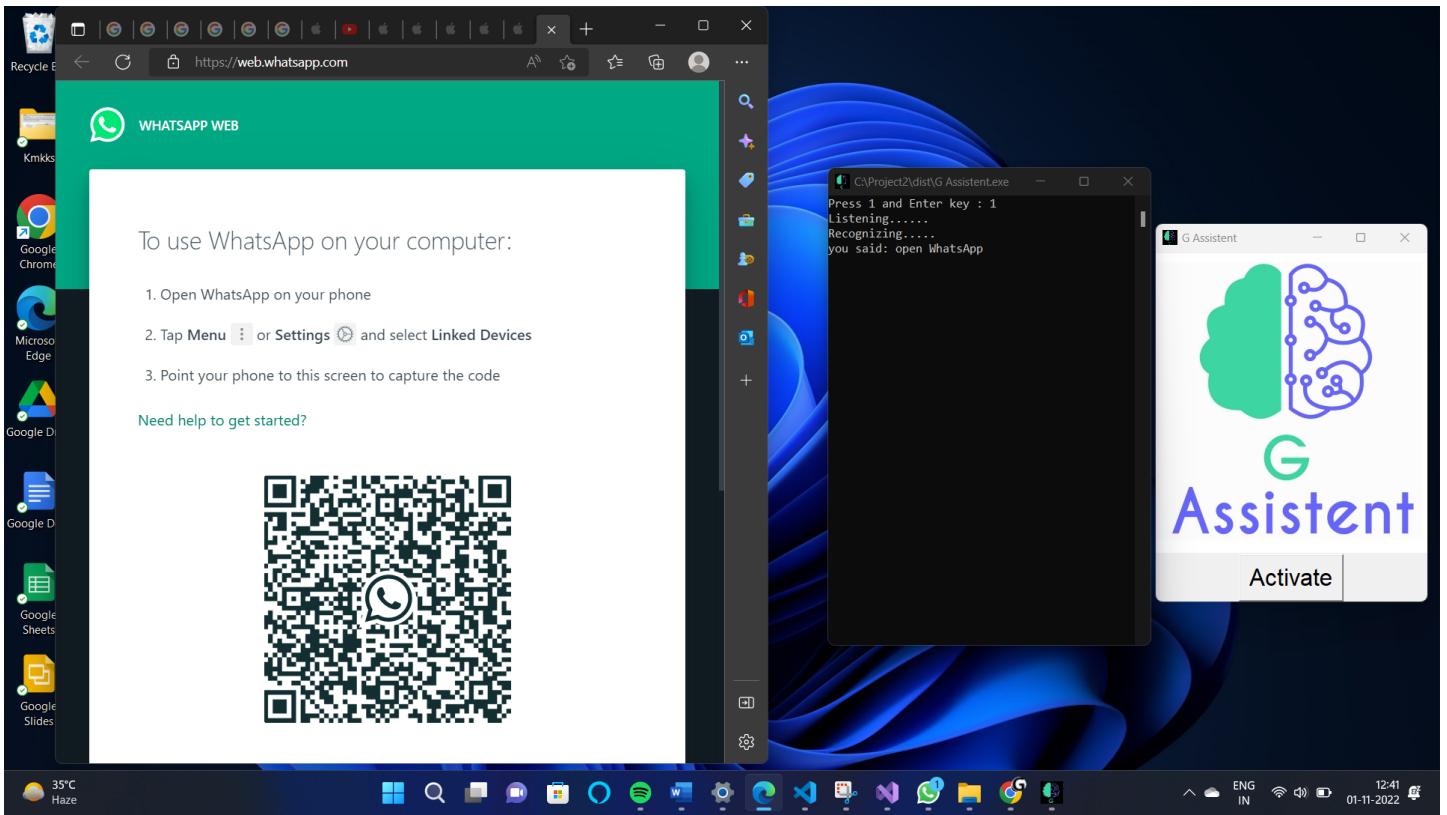
- What is my IP Address



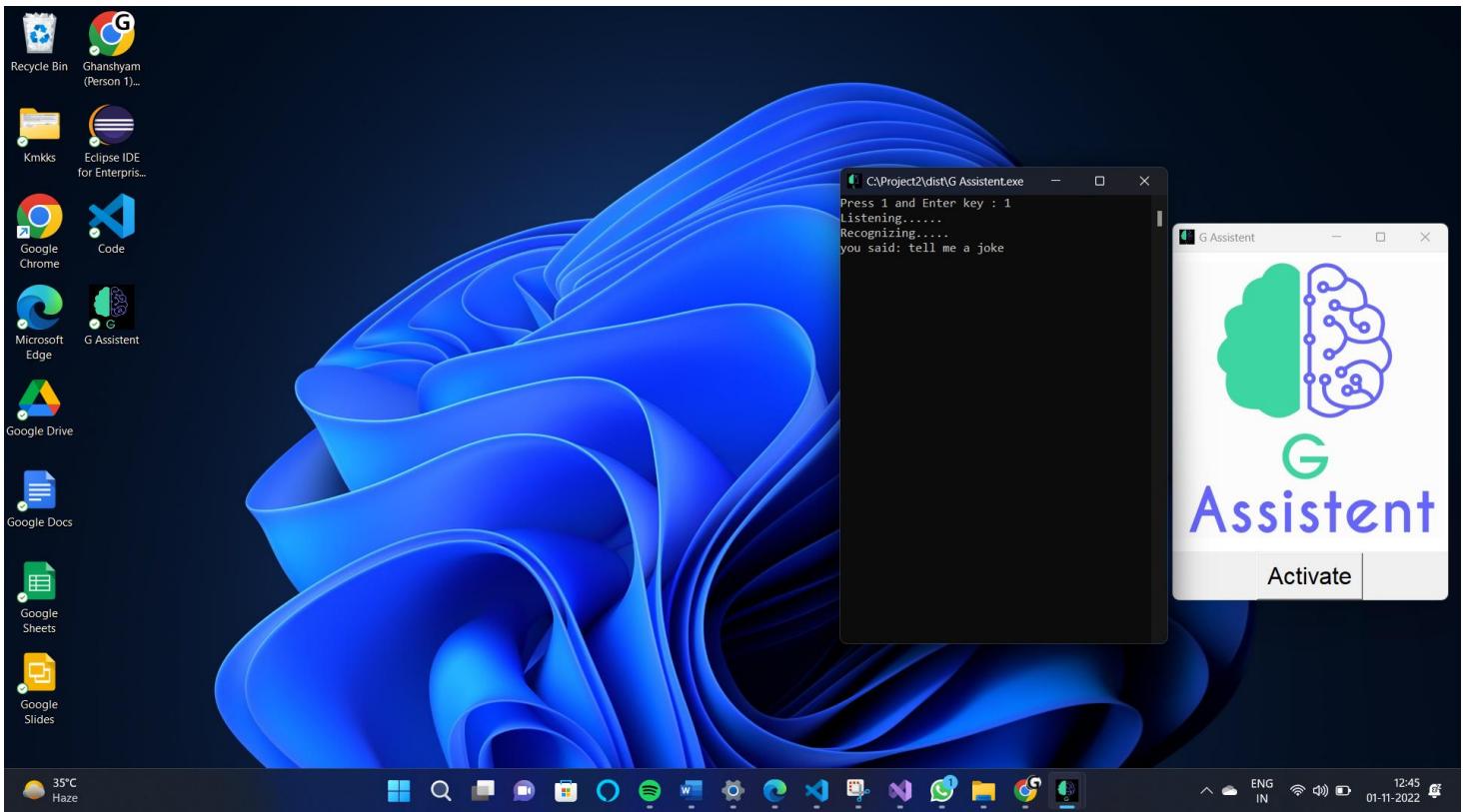
- Elon Musk according to wikipedia



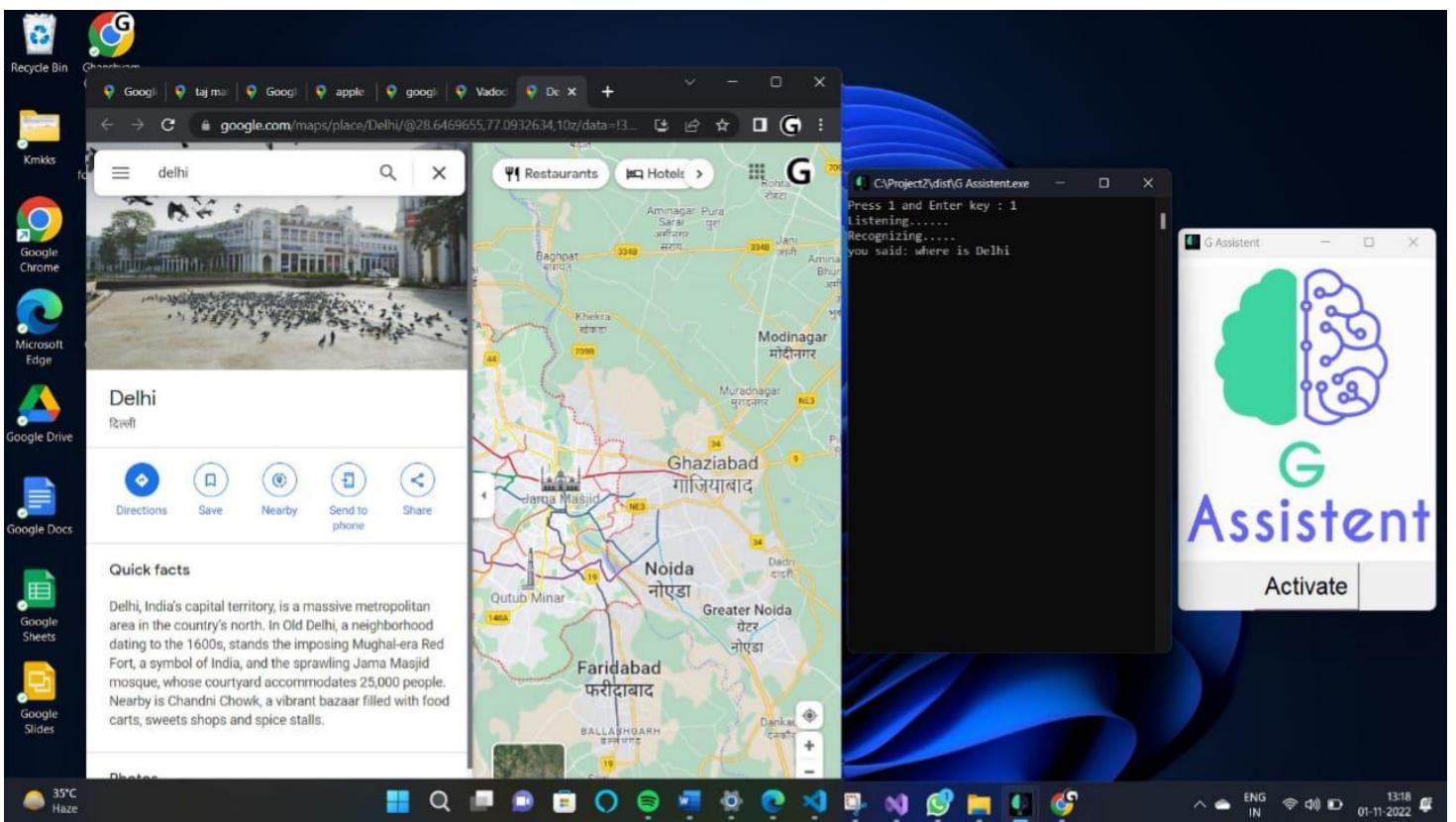
• open WhatsApp



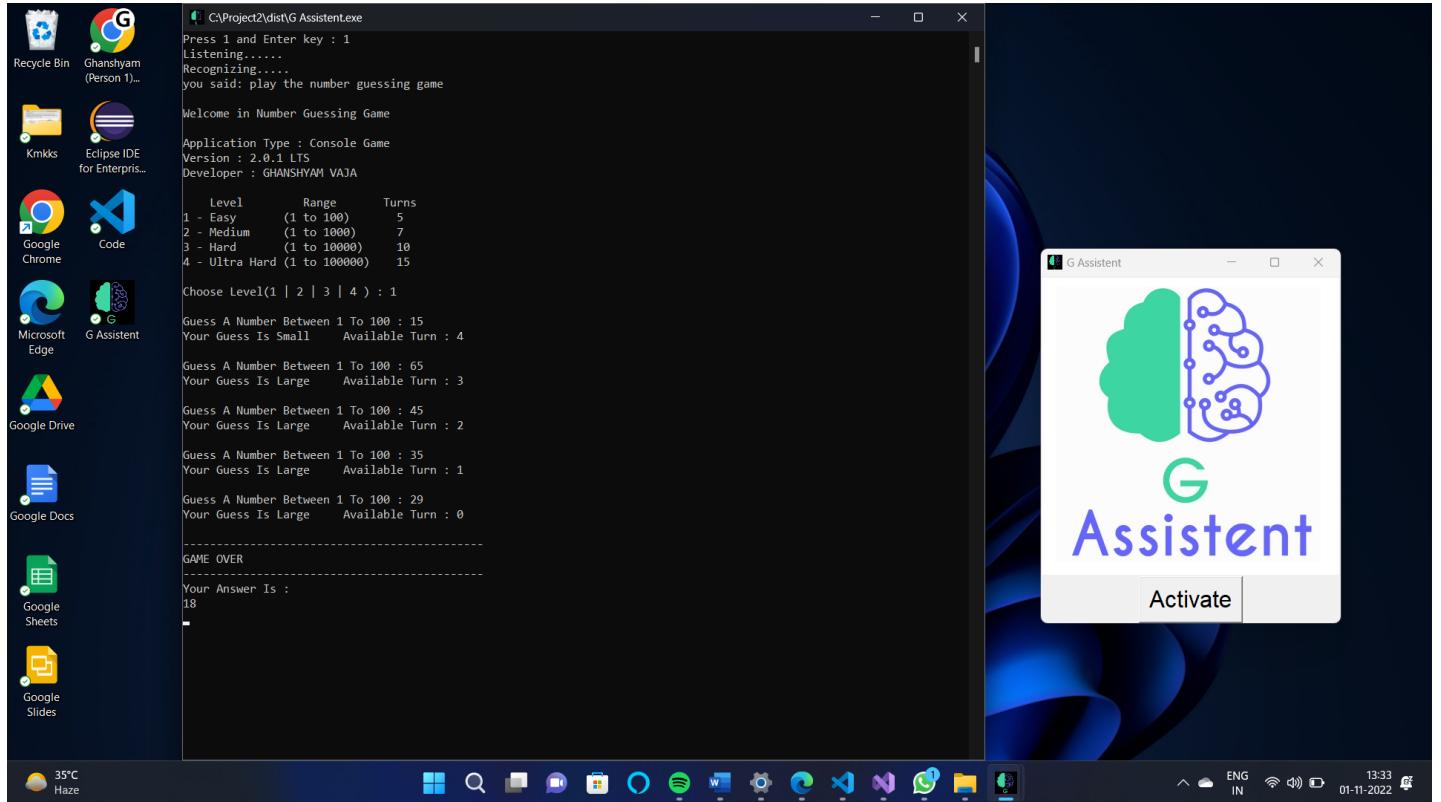
• tell me a joke



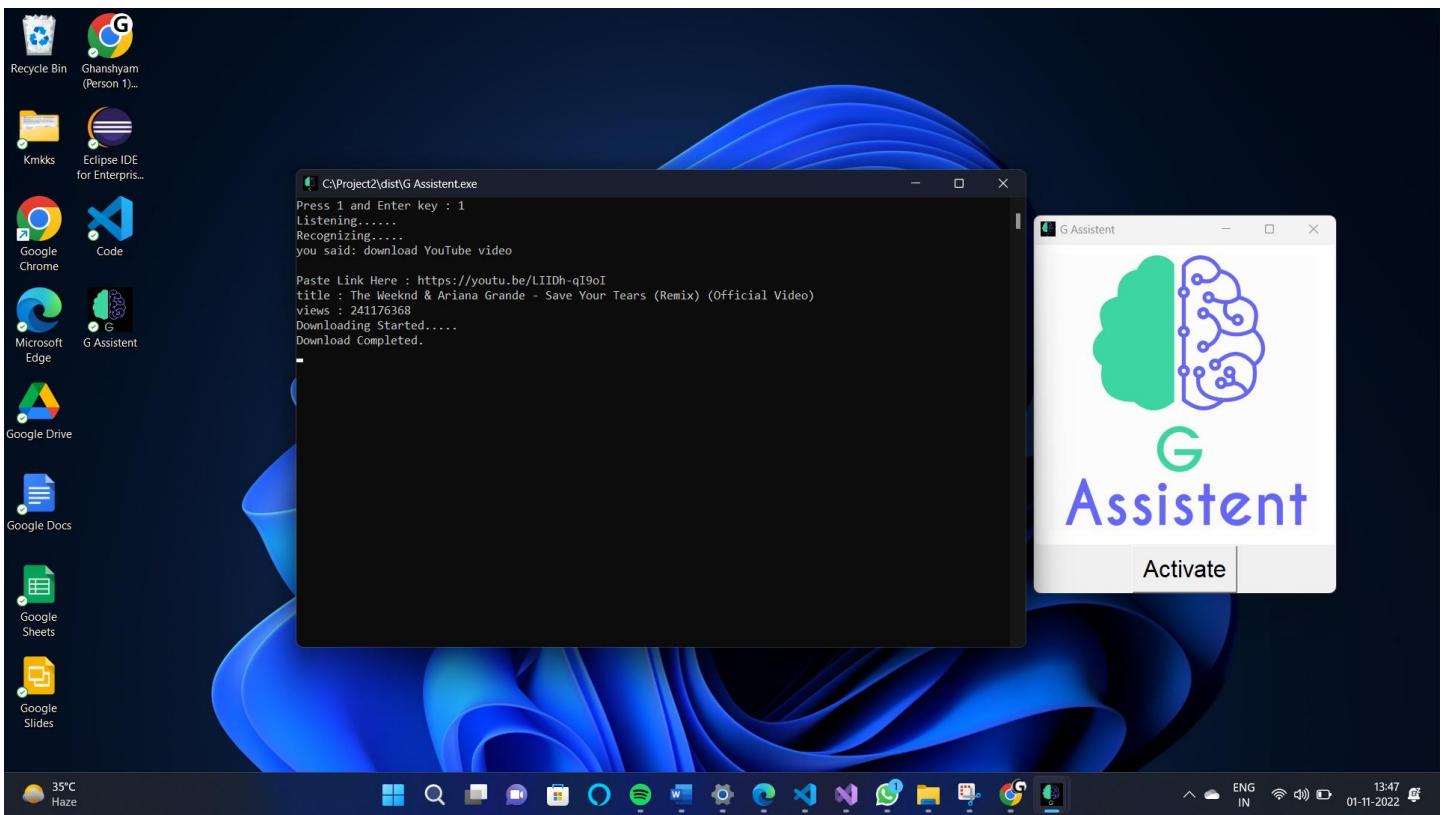
- where is Delhi



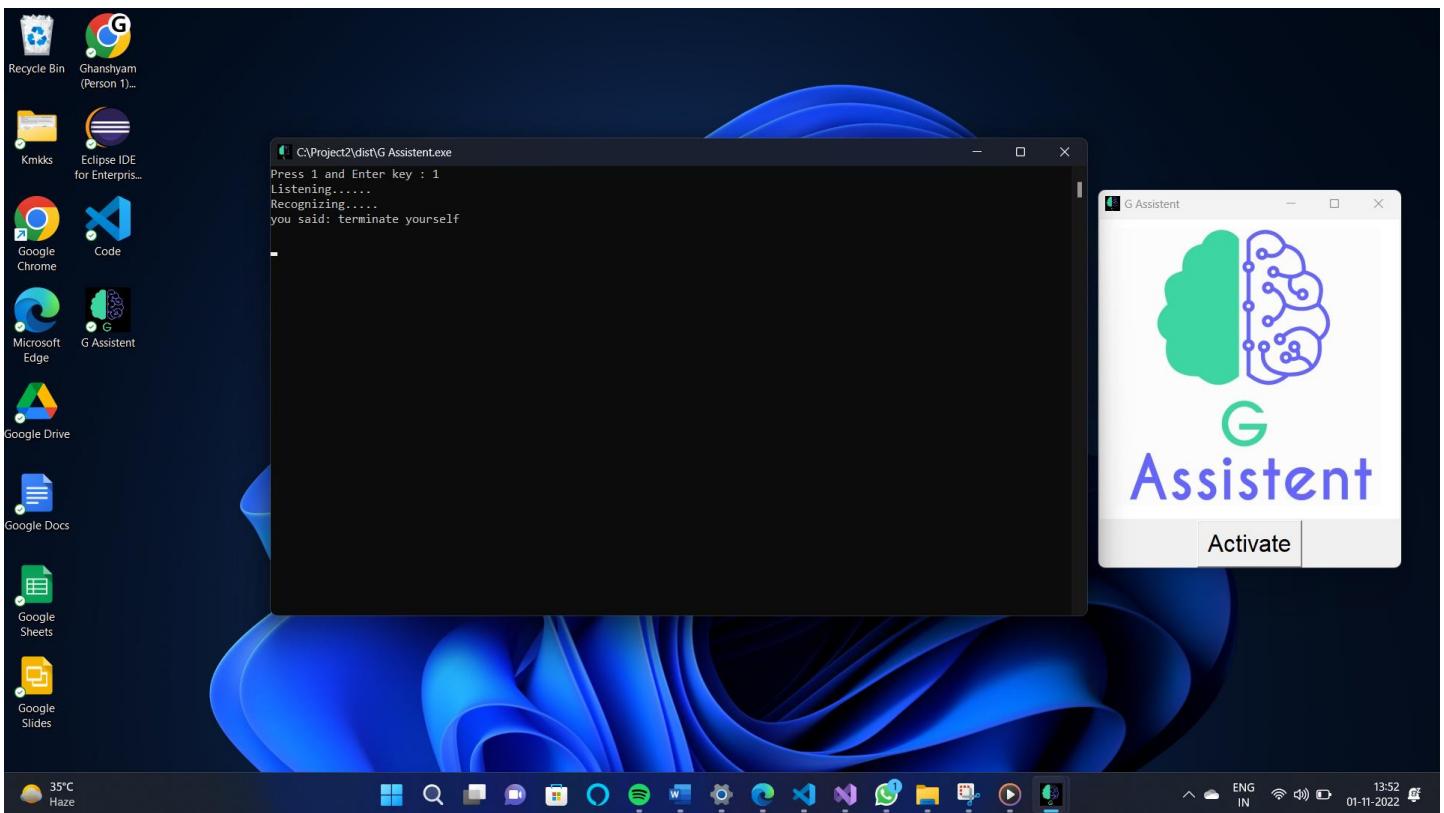
- play the number guessing game



- download Youtube video



- terminate yourself



13. Limitations

Voice recognition software turns speech into text -- we talk into the system and it transcribes what we say. This is useful for people with **visual impairments and those with physical problems that make typing on a keyboard difficult**. Others may use a system because **they find talking easier than typing or simply because it's fun**. Voice recognition technology is not perfect, however, and comes with a few Limitations.

- Lack of Accuracy and misinterpretation
- Time Costs and Productivity
- Accents and Speech Recognition
- Background Noise Interference
- Lack of Advance Level Machine Learning
- Lack of Advance Voice Processing
- Limited Size of Database Storage

14. Future Enhancements

We can do many future enhancements in this project Like,

- Apply Machine Learning,
- Increase Accuracy,
- Increase Natural Language Processing Capability,
- Add own voices,
- Reduce Execution Time of commands,
- Control Home Appliances Through this Assistant using Automation,
- Weather Prediction etc.....

15. Bibliography

I have used

- **Python 3.10.8** for Write Beckend Logic,
- **Tkinter** For GUI and
- **MySQL 8.0** for Database. Here I have used remote database provided by **db4free.net**.

16. Reference

[Python official site](#)

[PyPI](#)

[GeeksForGeeks](#)

[W3Schools](#)

[Stackoverflow](#)

[JavaTPoint](#)