****

**Course: PYTHON PROGRAMMING**

**PYTHON MINI PROJECT**

**VIRTUMATE**

**(a virtual assistant built using Python)**

**BY**

**PREETHA S- 21BTRCL078**

**SHREYA BITLA – 21BTRCL097**

**SNIGDHA V- 21BTRCL102**

# June – 2023 Global Campus

Jakkasandra Post, Kanakapura Taluk, Ramanagara District - Pin Code: 562 112



# Global Campus

Jakkasandra Post, Kanakapura Taluk, Ramanagara District - Pin Code: 562 112

**SEMESTER: 4**

**Python Programming**



# 2022-2023

Register No.: 21BTRCL078

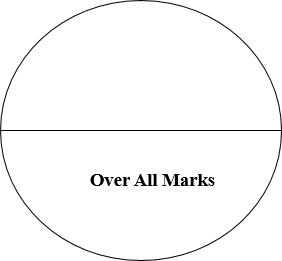
21BTRCL097

21BTRCL102

It is hereby certified that this is the bonafide record of work done by

Ms. Preetha S (21BTRCL078), Ms. Shreya Bitla (21BTRCL097) and

Ms. SnigdhaV (21BTRCL102) during the academic year 2022-2023 and submitted for the University Project work held on \_01/06/2023\_.



**Staff-in-charge Head of the Department**

**Internal Examiner External Examiner**

|  |  |  |
| --- | --- | --- |
| **SL NO.** | **TITLE** | **PAGE NO.** |
| 1. | Introduction | 2 |
| 2. | Implementation | 5 |
| 3. | Screenshots | 8 |
| 4. | Video of implementation | 10 |
| 5. | References | 11 |

# TABLE OF CONTENT

**Virtumate**

In today's fast-paced and technologically advanced world, AI assistants have become increasingly popular as they offer personalized and efficient solutions to users' queries and tasks. By leveraging the power of natural language processing (NLP), machine learning, and integration with external APIs and databases, this project aims to develop an intelligent assistant capable of understanding and responding to user inputs.

The primary goal of the AI assistant is to provide users with a seamless and interactive experience, enabling them to perform a wide range of tasks, access information, and receive assistance in real-time. The assistant will be designed to understand natural language queries, thereby eliminating the need for users to learn specific commands or follow a rigid interaction pattern. This intuitive and user-centric approach enhances the accessibility and usability of the AI assistant.

Python, a versatile and widely used programming language, is chosen as the foundation for this project. Python offers numerous libraries and frameworks for NLP, machine learning, and web development, making it an ideal choice for developing an AI assistant. The flexibility and ease of use provided by Python enable efficient implementation and integration of various components required for the assistant's functionality.

The project encompasses several key objectives, including the development of an AI assistant capable of interpreting user queries, implementing NLP algorithms to understand user intent, employing machine learning techniques for improved accuracy and responsiveness, integrating with external APIs and databases for fetching real-time information, creating a user-friendly interface for seamless interaction, and continuously improving the assistant's capabilities

1

**Chapter 1**

**INTRODUCTION**

The AI assistant serves as a virtual companion, capable of engaging in conversational interactions with users. By analyzing and interpreting natural language queries, the assistant aims to comprehend the intent behind user inputs and generate appropriate responses. This project seeks to empower the AI assistant with a diverse range of functionalities, making it an indispensable tool for users in their daily lives. The choice of Python as the programming language for this project is strategic due to its versatility, ease of use, and extensive libraries and frameworks available for NLP and machine learning. Python's clean syntax and robust ecosystem facilitate the development process, enabling efficient implementation of complex algorithms and integration with external APIs and databases.

Furthermore, the successful completion of this project can contribute to the advancement of AI technology. Insights gained through the development process can be applied to future research and development efforts, facilitating the evolution of intelligent systems and opening up new possibilities for human-computer interaction.

Overall, this project represents an exciting endeavor to create an AI assistant using Python, aiming to empower users with a sophisticated and intuitive tool that enhances productivity, provides information, and simplifies daily tasks. The potential applications and benefits of such an AI assistant are vast, promising to reshape the way individuals engage with technology and paving the way for a more efficient and intelligent future.

2

**OBJECTIVES:**

The main objectives of this project are as follows:

* Develop an AI assistant capable of understanding and responding to user queries.
* Implement natural language processing algorithms to analyze and interpret user inputs.
* Utilize machine learning techniques to improve the assistant's accuracy and responsiveness.
* Integrate with various APIs and databases to fetch relevant information and perform tasks.
* Create a user-friendly interface for seamless interaction with the AI assistant.
* Continuously improve and refine the assistant's capabilities based on user feedback and usage patterns.

**LIMITATIONS:**

Scope and Complexity: Developing a fully functional AI assistant that can understand and respond accurately to a wide range of user queries and perform complex tasks is a challenging task. The scope of the project may need to be limited to specific domains or tasks to ensure a manageable level of complexity.

Data Availability and Quality: The performance and effectiveness of the AI assistant heavily rely on the availability and quality of training data. Obtaining a diverse and representative dataset for training the assistant can be a challenging task. Limited or biased data may result in reduced accuracy and performance.

Integration Challenges: Integrating the AI assistant with external APIs and databases to access real-time information and perform tasks can introduce technical challenges. Inconsistent APIs, changes in data formats, and limitations imposed by external systems may affect the assistant's functionality and reliability.

3

Ethical Considerations: AI assistants raise ethical concerns regarding privacy, data security, and the potential for biased responses. Safeguarding user data, ensuring transparency in data usage, and addressing biases in the AI assistant's training data and decision-making process are crucial aspects that require careful attention.

Technical Constraints: The implementation of certain advanced features, such as real-time speech recognition or natural language generation, may pose technical constraints based on hardware limitations or t he availability of specialized libraries and resources

4

**Chapter 2**

IMPLEMENTATION

PYTHON CODE:

#importing necessary packages

import speech\_recognition as sr

import pyttsx3

import pywhatkit

import datetime

import wikipedia

import pyjokes

import sys

listener = sr.Recognizer() #Recognising your voice

engine = pyttsx3.init()

voices = engine.getProperty('voices')

engine.setProperty('voice', voices[1].id) #changing the voice

def talk(text):

engine.say(text)

engine.runAndWait()

def take\_command():

try:

with sr.Microphone() as source:

#talk('Hi! I am your personal virtual assistant alpha.')

#talk('how can i help you?')

talk('Please give your command')

5

print('listening.....')

voice = listener.listen(source) #listening to the source

command = listener.recognize\_google(voice, language = "en-IN")

command = command.lower()

print(command)

if 'alpha' in command:

command = command.replace('alpha', '')

except:

pass

return command

def run\_alpha():

command = take\_command()

#print(command)

if 'stop' in command:

sys.exit()

elif 'play' in command:

song = command.replace('play', '')

talk('playing ' + song)

pywhatkit.playonyt(song)

elif 'time' in command:

time = datetime.datetime.now().strftime('%I:%M %p')

print(time)

6

talk('The Current time is ' + time)

elif 'who' in command:

person = command.replace('who', '')

info = wikipedia.summary(person, 1)

print(info)

talk(info)

elif 'joke' in command:

joke = (pyjokes.get\_joke())

print(joke)

talk(joke)

else:

talk('Please say that command again.')

talk('Hi! I am your personal virtual assistant alpha.')

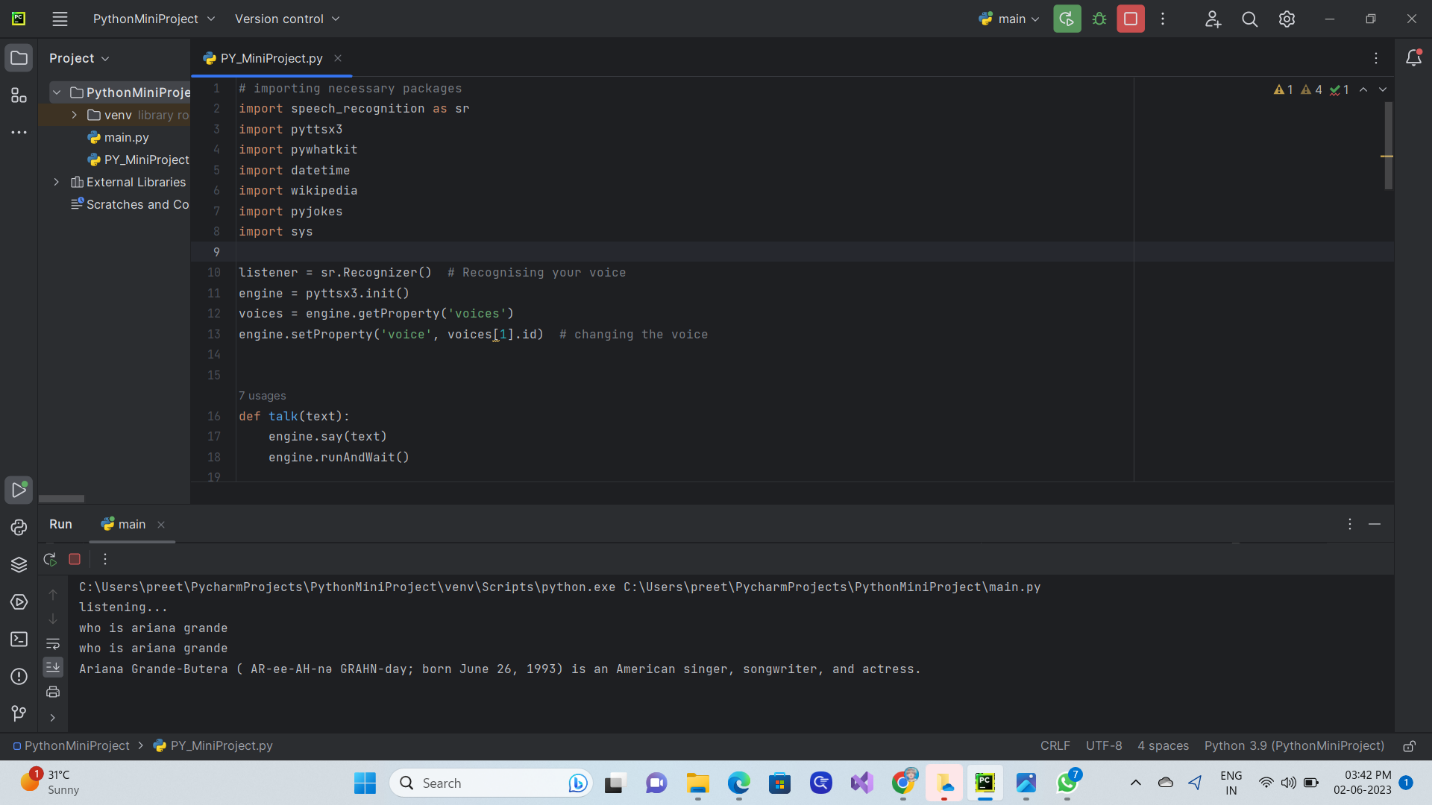
while True:

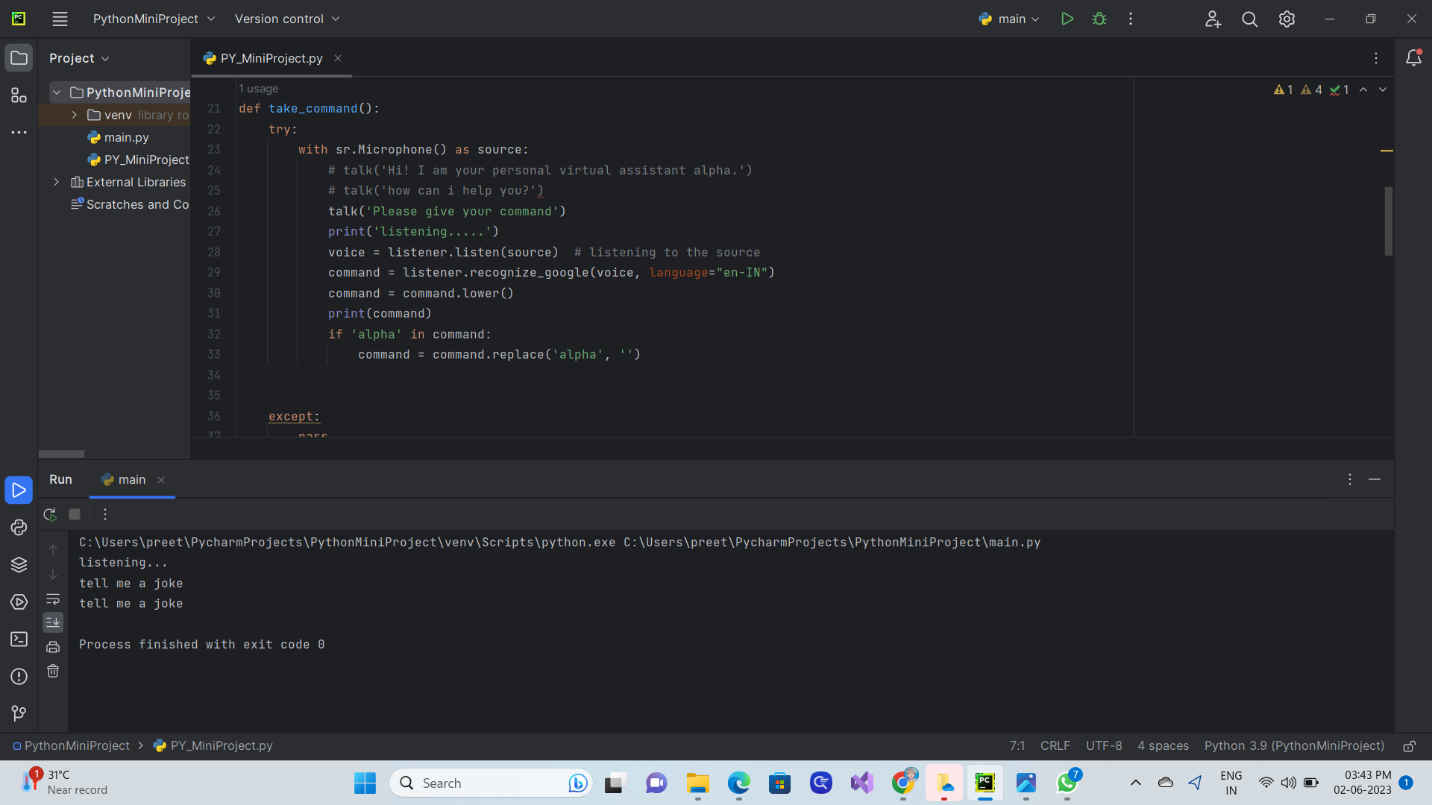
    run\_alpha()

7

**SCREENSHOTS:**

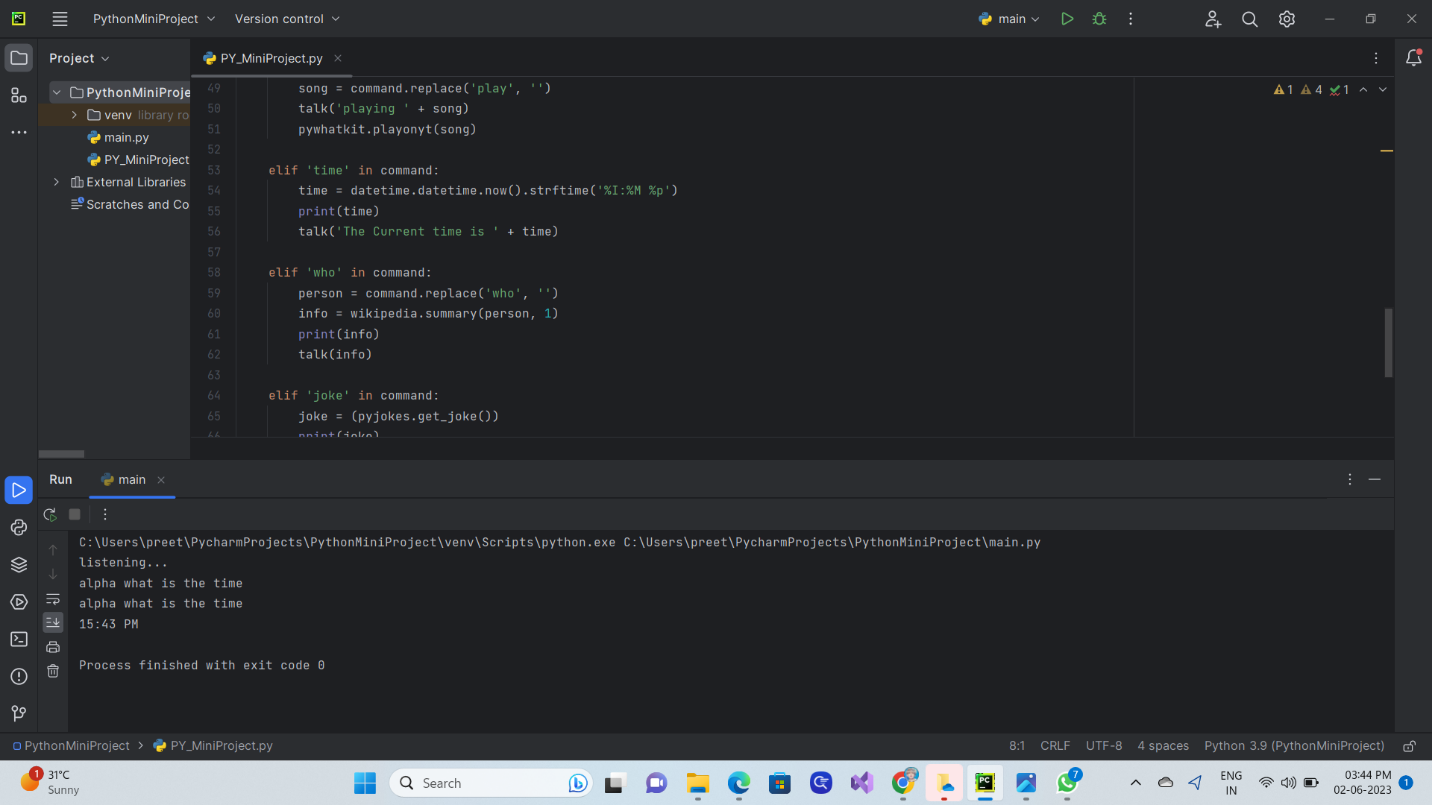
Creating a try block and using the microphone as the audio source and calling speech recognizer to listen to the source:

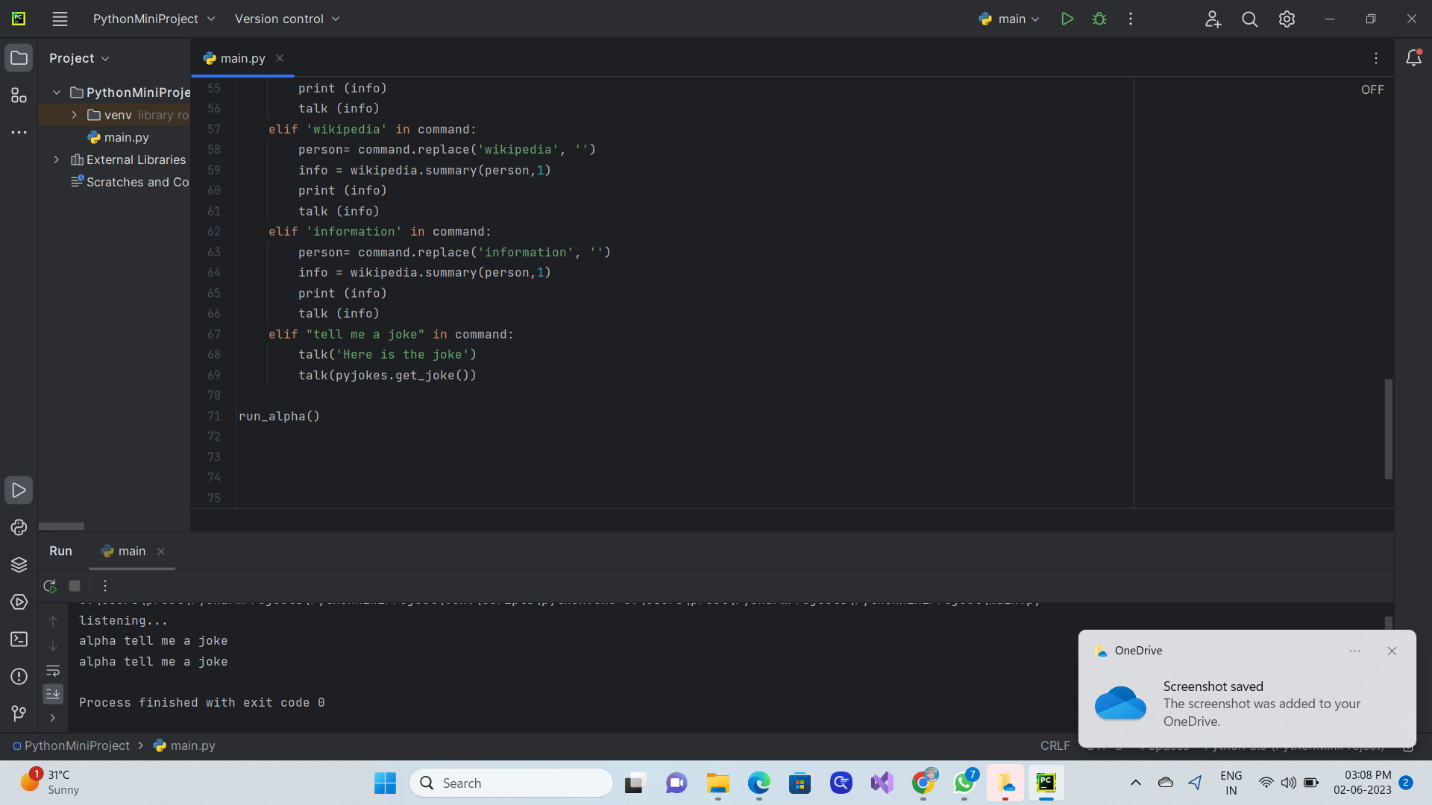


Asking the assistant to search for a person and the results from Wikipedia is the output:

8

For current time:

  
Calling the run\_alpha() method:

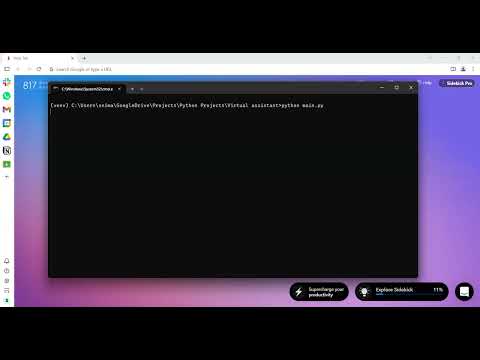


VIDEO WITH EXPLANATION IS UPLOADED IN THE GOOGLE CLASSROOM

9

**VIDEO OF IMPLEMENTATION OF VIRTUMATE**

Click the play button below to watch the video:

**[](https://www.youtube.com/embed/cblwog5MWGE?feature=oembed)**

**Youtube link for the video:** <https://youtu.be/cblwog5MWGE>

10

**REFERENCES:**

**Documentation/Official Documentation**: For detailed information on the python libraries and frameworks used:

1. Wikipedia library- https://pypi.org/project/wikipedia/

2. pyjokes library- https://pypi.org/project/pyjokes/

3. speech recognition library- https://pypi.org/project/SpeechRecognition/

4. python text to speech conversion library- https://pypi.org/project/pyttsx3/

5. pywhatkit library- https://pypi.org/project/pywhatkit/

**Online tutorials**: Youtube - https://www.youtube.com/watch?v=AWvsXxDtEkU

**Blog posts**: Geeks For Geeks - https://www.geeksforgeeks.org/build-a-virtual-assistant-using-python/

Title: Build a Virtual Assistant Using Python

Author: Abhishek Srivastav

11

**-THE END-**