# Project Name:

# Voice Assistant using Python

Name: YEDUVAKA LIKHITHA

Designation: Python Development Intern

Intern ID: CC44573

Batch: August 2023

Mail ID: [20891a4914@raghuenggcollege.in](mailto:20891a4914@raghuenggcollege.in)

# Context Introduction and Usage:

In today's digital world, voice assistants have become an integral part of our lives. They provide convenience by allowing users to interact with technology using natural language. As a Python Development Intern, your objective would be to create a Voice Assistant that can understand and respond to user commands, enhancing the user's experience with technology.

# The Voice Assistant can be used in various scenarios:

* Automation of tasks: Users can ask the assistant to perform actions like sending emails, opening applications, setting reminders, etc.
* Information retrieval: The assistant can fetch weather updates, news, or answer general knowledge queries.
* Language practice: Users can interact with the assistant to improve language skills or practice conversations.
* Entertainment: The assistant can tell jokes, play music, or engage in interactive storytelling.

# Objective:

The main objective of this project is to develop a functional Voice Assistant using Python. This involves:

* Speech recognition: Converting spoken language into text.
* Natural language processing (NLP): Understanding the meaning of the text input.
* Response generation: Creating appropriate responses in natural language.
* Integration with external services: Fetching data from APIs, performing actions on the system, etc.

# Project Prerequisites:

Before starting the project, you should have a solid understanding of the following:

* Python programming fundamentals.
* Basics of natural language processing and speech recognition.
* Familiarity with working with APIs to fetch data.
* Experience with using libraries such as `speech\_recognition`, `pyttsx3`, and `requests` in Python.

# Software Requirements:

Development Environments (IDEs):

* Visual Studio Code (VS Code): A popular, free source code editor.
* PyCharm: A dedicated Python IDE with advanced features.
* Thonny: A beginner-friendly Python IDE.

The project requires the use of specific Integrated

Choose one of these IDEs based on your familiarity and comfort level.

# Key Steps:

1. Set up the chosen IDE and create a new Python project.

2. Install necessary packages (`speech\_recognition`, `pyttsx3`, etc.) using a package manager like `pip`.

3. Implement speech recognition to convert spoken words into text.

4. Utilize NLP techniques to understand user intents and commands.

5. Integrate with external APIs to fetch information or perform actions.

6. Generate natural language responses using text-to-speech conversion.

7. Create a user-friendly interface for interaction.

Remember to test your Voice Assistant thoroughly and iterate on its functionality to make it as robust and user-friendly as possible.

Certainly! Here are the steps you can follow to create a Voice Assistant using Python for your Python Development Intern project. Remember, the steps provided are a general guideline, and you can adapt and expand them as needed for your specific project.

# Step 1: Set Up Your Development Environment:

* Choose one of the IDEs you mentioned (VS Code, PyCharm, Thonny) and install it on your computer.
* Create a new Python project within the chosen IDE.

# Step 2: Install Required Packages:

* Open a terminal within your IDE and install the necessary packages using the following commands:

pip install speechrecognition pyttsx3 requests

# Step 3: Implement Speech Recognition:

* Use the `speech\_recognition` library to capture and recognize speech input from the user.
* Implement code to convert the spoken words into text.

# Step 4: Natural Language Processing (NLP):

* Utilize NLP techniques to understand the user's intent from the text input.
* Decide on the commands your assistant will support and create patterns for recognizing them.

# Step 5: API Integration:

* Integrate your assistant with external APIs to fetch information or perform actions. For example, you could use APIs for weather updates, news, or data retrieval.
* Use the `requests` library to make API requests and handle responses.

# Step 6: Text-to-Speech Conversion:

* Implement the `pyttsx3` library to generate natural language responses in audio format.
* Convert the text responses generated by your assistant into speech output.

# Step 7: Interaction Loop:

* Create a loop that continuously listens for user input and responds accordingly.
* Implement logic to handle different commands and user interactions.

# Step 8: User Interface:

* If you want a graphical user interface (GUI), you can use libraries like `tkinter` to create a simple interface for user interaction.

# Step 9: Testing and Refining:

* Test your Voice Assistant thoroughly to ensure that it understands commands correctly and provides accurate responses.
* Iterate and refine your code based on user testing and feedback.

# Step 10: Documentation:

* Document your project by creating a README file that explains how to set up and use your Voice Assistant.
* Include information about the functionalities, supported commands, and any other relevant details.

# Source code:

import speech\_recognition as sr

import pyttsx3

import requests

# Initialize speech recognition and text-to-speech engines

recognizer = sr.Recognizer()

engine = pyttsx3.init()

def get\_weather(city):

# Replace with an actual weather API endpoint

url = f"https://api.example.com/weather?city={city}"

response = requests.get(url)

data = response.json()

weather\_description = data["weather"][0]["description"]

temperature = data["main"]["temp"]

return f"The weather in {city} is {weather\_description} with a temperature of {temperature}°C."

def main():

while True:

try:

with sr.Microphone() as source:

print("Listening...")

audio = recognizer.listen(source)

user\_input = recognizer.recognize\_google(audio).lower()

print("User said:", user\_input)

if "weather" in user\_input:

city = input("Which city's weather would you like to know? ")

response = get\_weather(city)

elif "hello" in user\_input:

response = "Hello! How can I assist you?"

else:

response = "I'm sorry, I didn't understand that."

print("Assistant:", response)

engine.say(response)

engine.runAndWait()

except sr.UnknownValueError:

print("Sorry, could not understand audio.")

if \_\_name\_\_ == "\_\_main\_\_":

main()

# Summary:

In summary, this project involves building a voice Assistant using Python for your internship. It requires understanding speech recognition, natural language processing, API integration, and using the chosen IDE (VS Code, PyCharm, or Thonny). By completing this project, you’ll gain practical experience in python development, NLP, and voice-based interactions.

# OUTPUT:



