**Virtual Assistant project using python:**

**Project Purpose**

A voice assistant that:

* **Listens** to what the user says 🎤
* **Understands** it as text 🧠
* **Acts** (e.g., tells time, opens Google/YouTube, tells a joke) ⚡
* **Speaks back** to the user 🔊

**🧱 How to Explain Each Part:**

**🔹 1. Import Modules**

import pyttsx3

import speech\_recognition as sr

import datetime

import webbrowser

import pyjokes

📘 **Explain:**  
We import Python libraries:

* pyttsx3: for talking (text-to-speech)
* speech\_recognition: for listening to your voice
* datetime: to get current time
* webbrowser: to open websites
* pyjokes: to tell funny jokes

**🔹 2. Initialize Voice Engine**

engine = pyttsx3.init()

📘 **Explain:**  
We create a voice engine so the computer can speak back.

**🔹 3. talk(text) Function**

def talk(text):

engine.say(text)

engine.runAndWait()

📘 **Explain:**  
This function takes any text and speaks it aloud using the engine.

✅ **Example:**

talk("Hello, I am your assistant.")

**🔹 4. listen() Function**

def listen():

listener = sr.Recognizer()

with sr.Microphone() as source:

print("🎤 Listening...")

listener.adjust\_for\_ambient\_noise(source)

audio = listener.listen(source)

📘 **Explain:**

* It turns on the microphone and listens to your voice.
* Then it converts voice to text using Google API.

Try/Catch handles errors like no internet or unclear speech:

try:

command = listener.recognize\_google(audio)

command = command.lower()

print("👂 You said:", command)

return command

except sr.UnknownValueError:

talk("Sorry, I didn't understand. Please try again.")

return ""

except sr.RequestError:

talk("Sorry, there was a network error.")

return ""

**🔹 5. Main Function: run\_assistant()**

def run\_assistant():

talk("Hi! How can I help you?")

command = listen()

📘 **Explain:**  
Starts by greeting the user and calling listen() to get the command.

**🔹 6. Command Processing (if-elif)**

if 'time' in command:

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

talk('The time is ' + time)

📘 **Explain:**  
If user says "time", get the current time and speak it.

elif 'search' in command:

search\_term = command.replace('search', '')

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

webbrowser.open(url)

talk('Here is what I found for ' + search\_term)

📘 **Explain:**  
If user says "search something", it opens Google search.

elif 'open youtube' in command:

webbrowser.open('https://www.youtube.com')

talk('Opening YouTube.')

📘 **Explain:**  
Opens YouTube when the user says “open YouTube”.

elif 'joke' in command:

joke = pyjokes.get\_joke()

talk(joke)

📘 **Explain:**  
Grabs a random joke from the pyjokes library and speaks it.

elif 'stop' in command or 'exit' in command:

talk("Goodbye!")

exit()

📘 **Explain:**  
If user says “stop” or “exit”, the assistant ends.

else:

talk("Sorry, I can't do that yet.")

📘 **Explain:**  
Handles anything else the assistant doesn’t understand.

**🔹 7. Run the Assistant**

run\_assistant()

📘 **Explain:**  
Starts the whole thing. This line calls the run\_assistant() function.

**Explanation:**

| **Step** | **Method** |
| --- | --- |
| 1. **Explain the goal** | “You’ll build an assistant like Alexa that listens and talks back” |
| 2. **Run live demo** | Show a sample interaction first |
| 3. **Break the code** | Explain function-by-function as above |
| 4. **Interactive testing** | Let students try different commands |
| 5. **Customize** | Let them add features (e.g., open Facebook, play music, etc.) |

**🧪 Mini Practice for Students**

Ask them to:

* Add a new command: "open facebook"
* Change the voice speed or gender using pyttsx3

**✅ Summary Diagram**

text

CopyEdit

[User speaks] → [Microphone → speech\_recognition → Text]

↓

[Check command: time, search, joke...]

↓

[Action] → [Speak result with pyttsx3]

Complete code:

import pyttsx3  
import speech\_recognition as sr  
import datetime  
import webbrowser  
import pyjokes  
  
  
*# Initialize the speaker*engine = pyttsx3.init()  
  
def talk(text):  
 engine.say(text)  
 engine.runAndWait()  
  
def listen():  
 listener = sr.Recognizer()  
 with sr.Microphone() as source:  
 print("Listening...")  
 voice = listener.listen(source)  
 command = listener.recognize\_google(voice)  
 command = command.lower()  
 return command  
  
def run\_assistant():  
 command = listen()  
 print("You said:", command)  
  
 if 'time' in command:  
 time = datetime.datetime.now().strftime('%I:%M %p')  
 talk('The time is ' + time)  
  
 elif 'search' in command:  
 search\_term = command.replace('search', '')  
 url = 'https://www.google.com/search?q=' + search\_term  
 webbrowser.open(url)  
 talk('Here is what I found for ' + search\_term)  
  
 elif 'open youtube' in command:  
 webbrowser.open('https://www.youtube.com')  
 talk('Opening YouTube')  
  
 elif 'joke' in command:  
 joke = pyjokes.get\_joke()  
 talk(joke)  
  
 else:  
 talk("Sorry, I didn't understand that.")  
  
run\_assistant()