

## **Program coding:**

### **1. Input Data**

```
import speech_recognition as sr
import pyttsx3
import datetime
import requests
```

#### **# Initialize the text-to-speech engine**

```
engine = pyttsx3.init()
engine.setProperty('rate', 150) # slower speech for elders
```

```
def speak(text):
    engine.say(text)
    engine.runAndWait()
```

#### **# Get voice input**

```
def listen():
    recognizer = sr.Recognizer()
    with sr.Microphone() as source:
        print("Listening...")
        recognizer.adjust_for_ambient_noise(source)
        audio = recognizer.listen(source)

    try:
        command = recognizer.recognize_google(audio).lower()
        print("User said:", command)
        return command
    except:
        speak("Sorry, I didn't catch that.")
        return ""
```

#### **# Get time**

```
def tell_time():
    time = datetime.datetime.now().strftime('%I:%M %p')
    speak(f"The current time is {time}")
```

#### **# Get date**

```
def tell_date():  
    date = datetime.datetime.now().strftime('%A, %B %d, %Y')  
    speak(f"Today is {date}")
```

**# Sample weather function (you can expand using  
OpenWeatherMap API)**

```
def tell_weather():  
    speak("Today is sunny and warm.") # Static for now
```

**# Main assistant function**

```
def run_assistant():  
    speak("Hello! How can I help you today?")  
while True:  
    command = listen()  
  
    if "time" in command:  
        tell_time()  
    elif "date" in command:  
        tell_date()  
    elif "weather" in command:  
        tell_weather()  
    elif "stop" in command or "exit" in command:  
        speak("Goodbye! Take care.")  
        break  
    else:  
        speak("Sorry, I don't understand that yet.")
```

**# Run the assistant**

```
run_assistant()
```

## Output:

The screenshot displays the Visual Studio Code environment with the following components:

- File Explorer:** Shows the file `assistant.py`.
- Run and Debug:** The `RUN` tab is active, showing a `Run and Debug` button and instructions to create a `launch.json` file. The `BREAKPOINTS` section shows `Uncaught Exceptions` checked.
- Code Editor:** Contains the `assistant.py` script. The code defines `tell_date()`, `tell_weather()`, and `run_assistant()` functions. `run_assistant()` is a `while True` loop that listens for user input and calls the appropriate function based on the input.
- Terminal:** Shows the output of the program, including the `Listening...` prompts and user input: `what is the time`, `tell me date`, `how is the weather`, and `stop`. The prompt `PS D:\VCA>` is visible at the bottom.
- Taskbar:** Shows the Windows taskbar with the Start button, Search bar, and various application icons. The system tray shows the date and time as 14-05-2025, 20:54.

```
35 def tell_date():
36     speak(f"Today is {date}")
37
38
39 # Sample weather function (you can expand using OpenWeatherMap API)
40 def tell_weather():
41     speak("Today is sunny and warm.") # Static for now
42
43 # Main assistant function
44 def run_assistant():
45     speak("Hello! How can I help you today?")
46     while True:
47         command = listen()
48
49         if "time" in command:
50             tell_time()
51         elif "date" in command:
```

Listening...  
User said: what is the time  
Listening...  
User said: tell me date  
Listening...  
User said: how is the weather  
Listening...  
User said: stop  
PS D:\VCA>