

Multilingual AI Assistant

An AI-Powered Voice Assistant with Multilingual Capabilities
(GENERATIVE AI)

[Github-Link](#)

Project Overview

Objective: The project aims to create an AI assistant capable of understanding and responding in multiple languages. The assistant takes spoken input from the user, processes it, and delivers a response in the form of synthesized speech. The goal is to offer a seamless, real-time conversational experience.

Core Features:

- Voice Input and Output: Users interact with the assistant using their voice, both for input and output, making the experience more natural and accessible.
- Multilingual Support: The assistant can understand and respond in various languages, catering to a diverse user base.
- Real-Time Interaction: The system is designed to respond quickly, providing a smooth and efficient user experience.

Tech Stack

- Programming Language: The project is developed in Python, known for its versatility and rich ecosystem of libraries.
- Frameworks & Libraries:
 - Streamlit: Used to create the web application interface. Streamlit allows for the rapid development of interactive applications.
 - SpeechRecognition: This library is used to capture and convert voice input into text.
 - Google Generative AI (Gemini Model): Utilized for generating responses based on the user's input text. This model is known for its ability to generate coherent and contextually relevant responses.
 - gTTS (Google Text-to-Speech): This library converts text responses into synthesized speech, allowing for audio playback.
- APIs: The project leverages Google APIs for both speech recognition and text generation, ensuring high accuracy and reliability.
- Deployment Platform: The application is hosted on an AWS EC2 instance, providing scalability and flexibility for handling different workloads.

System Architecture

Workflow Overview:

1. Voice Input: The user speaks into a microphone, and the system captures this audio input.
2. Speech-to-Text Conversion: The captured audio is converted into text using Google's SpeechRecognition API. This step involves processing the audio to accurately transcribe spoken words into written form.
3. Text Processing and Generation: The transcribed text is sent to the Google Generative AI (Gemini model), which generates a relevant response. The model processes the input and generates coherent and contextually appropriate text.
4. Text-to-Speech Conversion: The generated text is then converted into an audio response using gTTS. This step synthesizes the response into a human-like voice, which is played back to the user.
5. Output: The system plays the audio response to the user. Additionally, users can download the audio file for offline use.

- Components Overview:
 - Frontend: The user interacts with the assistant through a web interface created using Streamlit. The interface includes buttons for initiating voice input and displays the text response along with audio playback.
 - Backend: Python scripts manage the core functionalities, including voice recognition, response generation, and text-to-speech conversion.
 - Deployment: The application is deployed on AWS EC2, providing a robust and scalable platform for hosting the service.

User Interaction Flow

- Interaction Steps:
 - a. Initiating Interaction: The user clicks a button on the Streamlit interface to start the interaction. This action triggers the system to begin listening for voice input.
 - b. Processing Input: Once the user speaks, the system captures the audio and converts it into text. This text is then processed to generate a response using the AI model.
 - c. Generating Response: The generated response is displayed as text on the interface and converted into speech for audio playback.
 - d. Delivering Output: The synthesized speech is played back to the user. Additionally, users have the option to download the audio file for later use.

DEPLOYMENT ON AWS EC2

Deployment Process:

- a. Launching the EC2 Instance: Set up an EC2 instance with appropriate specifications, including the instance type and security configurations.
- b. Environment Setup:
 - Updated the system packages and installed necessary software like Python, git, and other dependencies.
 - Cloned the project repository from GitHub and installed the required Python packages.
- c. Running the Application:
 - Temporarily ran the application using Streamlit for testing.
 - Configured the application to run as a background process for continuous operation.
- d. Access and Management: The application is accessible via a public IP address or domain, with the Streamlit app running on port 8501 by default. We set up security groups to manage access and ensure the system's security.

SOURCE CODE

```
src > helper.py > ...
8
9
10    load_dotenv()
11    GOOGLE_API_KEY = os.getenv("GOOGLE_API_KEY")
12    os.environ['GOOGLE_API_KEY'] = GOOGLE_API_KEY
13
14
15    def voice_input():
16        # Create a recognizer instance
17        r = sr.Recognizer()
18
19        with sr.Microphone() as source:
20            print("Listening...")
21            audio = r.listen(source)
22
23        try:
24            text = r.recognize_google(audio) # Using Google Speech Recognition
25            print("You said: " + text)
```

SOURCE

Microsoft Windows [Version 10.0.19045.4651]
(c) Microsoft Corporation. All rights reserved.

(mul) E:\project-8\Multi-lingual-Ai-Assistant>streamlit run app.py

You can now view your Streamlit app in your browser.

Local URL: <http://localhost:8501>

Network URL: <http://192.168.0.149:8501>

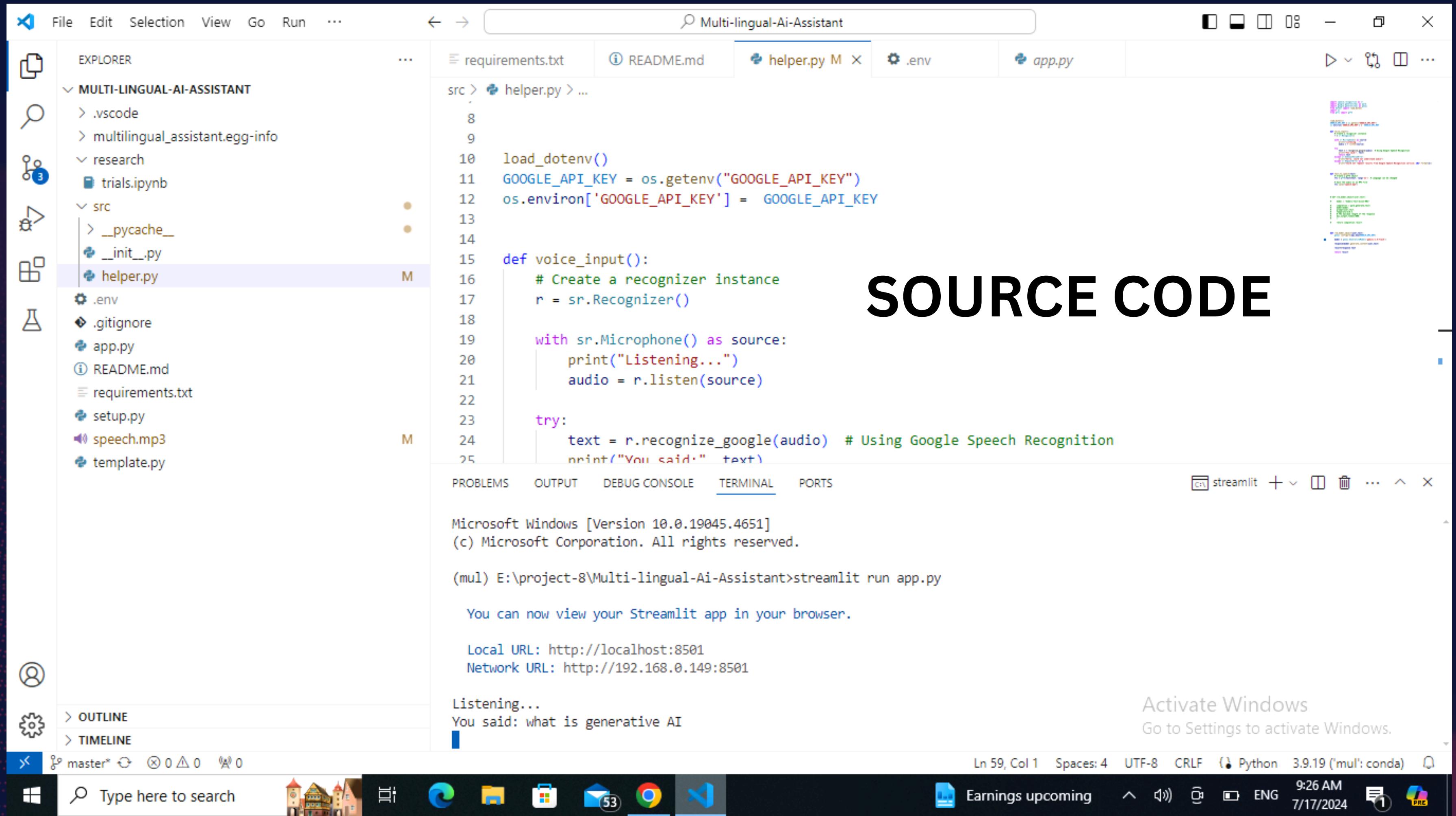
Listening...

You said: what is generative AI

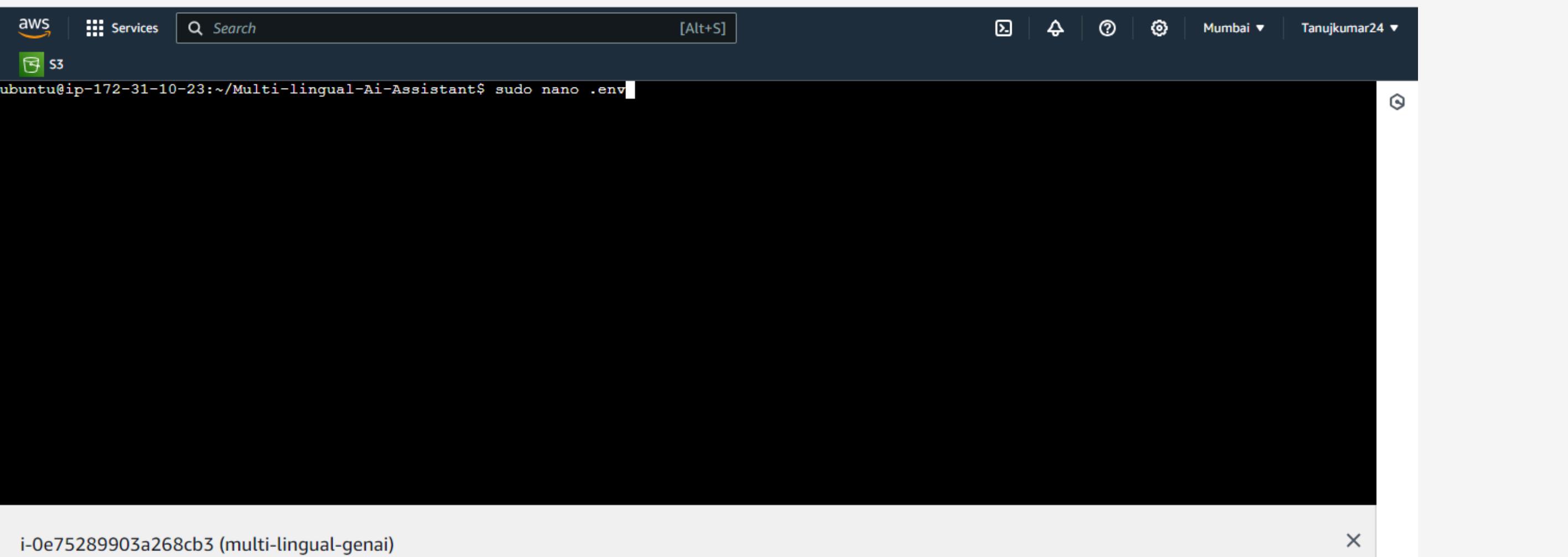
Activate Windows

Go to Settings to activate Windows

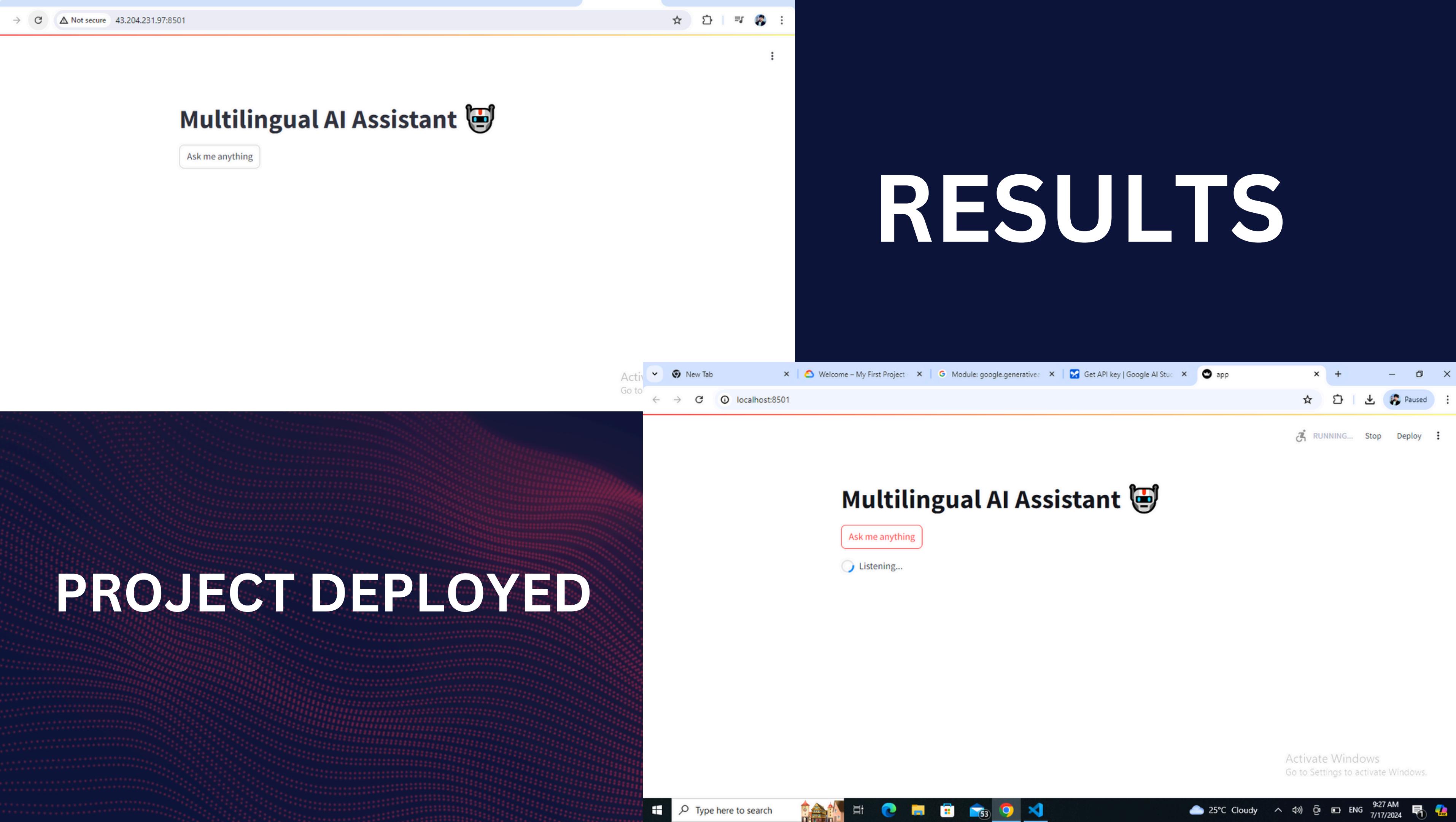
Ln 59, Col 1 Spaces: 4 UTF-8 CRLF (1) Python 3.9.19 ('mul': conda) Q



WORKING WITH AMAZON EC2



```
ubuntu@ip-172-31-10-23:~$ cd Multi-lingual-Ai-Assistant/
ubuntu@ip-172-31-10-23:~/Multi-lingual-Ai-Assistant$ ls
README.md app.py multilingual_assistant.egg-info requirements.txt research setup.py src template.py
ubuntu@ip-172-31-10-23:~/Multi-lingual-Ai-Assistant$ touch .env
```



Multilingual AI Assistant 🤖

Ask me anything

RESULTS

Multilingual AI Assistant 🤖

Ask me anything

Listening...

Activate Windows
Go to Settings to activate Windows.



output

what is Generative Ai?