Text Summarization Assignment using FastAPI

# 1. Introduction

This project involves building a Text Summarization API using FastAPI, where an input text is processed   
and summarized using an external API, Gemini API. The app takes a text and generates a summary of that text   
by sending a POST request to the Gemini API.

# 2. Steps Followed

## 2.1 Setting Up the Environment

- Installed Python, FastAPI, and Uvicorn for local development.  
- Used `requirements.txt` for managing dependencies, including `fastapi`, `uvicorn`, and `requests`.

## 2.2 Creating the FastAPI Application

- Designed the FastAPI app with a POST endpoint `/summarize-text`.  
- Defined the `SummarizeRequest` class using `pydantic.BaseModel` to handle the input.  
- Integrated the Gemini API for summarization with parameters like `text` and `max\_tokens`.  
- The API was called using the `requests` library to send the input to Gemini and receive the summarized text.

## 2.3 Testing the Application

- The app was tested locally using Visual Studio Code.  
- The app was run with the command `uvicorn app:app --reload`.  
- It was tested by sending POST requests with different text inputs and received the summarized output successfully.

# 3. Key Features

- The app uses FastAPI with a single endpoint `/summarize-text`.  
- It accepts a JSON body with `text` (the input text) and `max\_tokens` (optional).  
- The response is a summarized version of the input text generated by the Gemini API.

# 4. Deployment Steps

- The application was prepared for deployment by ensuring all dependencies were listed in `requirements.txt`.  
- After confirming local success, the app was ready to be hosted via FastAPI servers or deployed in cloud environments.

# 5. Steps Taken with commands

1. To create a virtual environment under folder '/Text\_Summarisation', execute the command in the VS Code terminal:  
   
 python -m venv venv  
   
2. To activate the above virtual environment, run:  
   
 venv\Scripts\activate  
   
3. To install the required packages, run:  
   
 pip install -r requirements.txt  
   
4. To run the application, execute:  
   
 uvicorn app:app --reload  
   
 - This command provides a URL to access the app locally.  
 - To access the \*\*FastAPI UI\*\*, append `/docs` at the end of the URL (e.g., `http://localhost:8000/docs`).

# 6. Challenges and Solutions

- One challenge was ensuring the proper API integration, but this was resolved by following documentation on sending requests with the correct parameters and parsing the returned JSON for the summarized text.  
- Another challenge was managing dependencies and ensuring the correct versions of libraries, which was solved   
by using a virtual environment and the `requirements.txt`.

# 7. Conclusion

The Text Summarization project using FastAPI and the Gemini API was successfully implemented, with a clear separation between the frontend API request and backend logic for summarization.   
The solution works efficiently for different text inputs and returns an accurate summary generated by the Gemini API.