AI Text Summarizer – Full Stack Production Project Guide

1. Project Introduction

This project is a production-level AI Text Summarizer built using a full-stack approach. Users can paste long articles or documents and receive a concise, meaningful summary generated using a Generative AI model.

2. Purpose of the Project

The purpose of this project is to understand how Generative AI models are integrated into real-world web applications. It helps in learning backend development, frontend development, database integration, and AI model handling together.

3. Technologies and Tools Used

Backend:
• Python 3.11
• FastAPI
• Hugging Face Transformers (facebook/bart-large-cnn)
• SQLModel + SQLite

Frontend:
• React (Vite)
• Tailwind CSS

Tools:
• VS Code
• Git & GitHub
• Uvicorn Server

4. System Architecture

The application follows a client–server architecture.

Frontend (React UI) → Backend API (FastAPI) → AI Model (Hugging Face) → Database (SQLite) → Response to Frontend

5. Application Flow (Flowchart Explanation)

1. User enters a long paragraph or document in the UI.
2. Frontend sends a POST request to the backend API.

3. Backend validates the input text.
4. Long text is split into smaller chunks.
5. Each chunk is summarized using the AI model.
6. All partial summaries are combined into a final summary.
7. The summary is saved into the database.
8. Final summary is returned to the frontend and displayed to the user.

6. Working of the Project (Step-by-Step)

Step 1: User pastes long text in the web interface.
Step 2: React frontend sends the text to the FastAPI backend.
Step 3: Backend applies chunking logic for long inputs.
Step 4: Hugging Face model generates summaries.
Step 5: Final summary is created and stored in SQLite database.
Step 6: Summary is displayed back on the UI.

7. How to Use This Project (For GitHub Users)

1. Clone the GitHub repository.
2. Set up a Python virtual environment for the backend.
3. Install backend dependencies.
4. Run the backend using Uvicorn.
5. Install frontend dependencies.
6. Run the frontend using npm run dev.
7. Open the application in the browser and test summarization.

8. Conclusion

This AI Text Summarizer project demonstrates how Generative AI can be used in a real full-stack application. It is suitable for portfolios, internships, and learning purposes, and serves as a strong foundation for advanced GenAI projects.