SmartSDLC\_Project/

│

├── Code/

│ └── main.py

│

├── Dataset/

│ └── requirements\_sample.pdf

│

├── Outputs/

│ └── sample\_output.txt

│

├── Docs/

│ ├── SmartSDLC\_Presentation.pptx

│ └── SmartSDLC\_Documentation.docx

│

├── Project Templates/

│ ├── Ideation Phase/

│ ├── Requirement Analysis/

│ ├── Project Design Phase/

│ ├── Project Planning Phase/

│ ├── Performance Testing/

│ ├── Final Report Template.pdf

│ └── FSD Documentation Format.pdf

│

├── README.md

└── requirements.txt

markdown

Copy code

## ▶️ How to Run 1. \*\*Install dependencies\*\* ```bash pip install -r requirements.txt

* **Run FastAPI backend**

bash

Copy code

uvicorn Code.main:app --reload

* **Use Streamlit frontend** (Optional UI)

bash

Copy code

streamlit run frontend.py

**🤖 Future Work**

* Integrate real IBM Watsonx API
* Add database for persistence and version tracking
* Role-based access control
* Cloud deployment (AWS/GCP)

**📬 Contact**

For questions or contributions, reach out to the project maintainer.

yaml

Copy code

--- ## 📘 Project Documentation (Text Format) ```text # SmartSDLC – Project Documentation ## 1. Introduction SmartSDLC is an AI-powered automation tool for the Software Development Lifecycle (SDLC). It converts unstructured user inputs into structured deliverables like code, tests, and documents. It uses IBM Watsonx, NLP, and Generative AI for end-to-end SDLC support. --- ## 2. Modules ### 🔹 1. Requirement Upload & Classification - \*\*Input\*\*: Raw PDF requirements - \*\*Output\*\*: Grouped user stories by SDLC phase - \*\*Tech\*\*: PyMuPDF, Watsonx ### 🔹 2. AI Code Generator - Converts user

stories into boilerplate or production code. - Supports Python, JS, and more. ### 🔹 3. Bug Fixer - Takes buggy code and returns corrected version with explanation. ### 🔹 4. Test Case Generator - Auto-generates `unittest` or `pytest` cases from logic or requirements. ### 🔹 5. Code Summarizer - Translates code into human-readable summary (useful for onboarding/dev docs). ### 🔹 6. AI Chatbot Assistant - LangChain-based chat assistant for real-time help during development. --- ## 3. Tools & Technologies - \*\*FastAPI\*\*: Backend API handling - \*\*Streamlit\*\*: Simple UI - \*\*PyMuPDF\*\*: PDF parsing - \*\*IBM Watsonx (Granite-20B)\*\*: AI classification and generation - \*\*LangChain\*\*: Conversational agent - \*\*Pinecone\*\*: Vector database (optional) - \*\*Transformers\*\*: NLP support - \*\*Scikit-learn, Pandas, Matplotlib\*\*: Data processing and visualization --- ## 4. How

to Use 1. Upload a requirements document. 2. View classified sentences grouped by SDLC stage. 3. Click to generate code, test cases, or summaries. 4. Use AI Chatbot for guidance. 5. Save/export all outputs. --- ## 5. Sample Workflow 1. Upload: `requirements\_sample.pdf` 2. Output: - Requirements → "System shall have login" - Design → "Use a responsive layout" - Development → Code snippet auto-generated 3. Debug and summarize with AI 4. Export Final Report from UI --- ## 6. Future Scope - Full integration with JIRA/Confluence - Real-time collaboration & commenting - Dataset finetuning for specific domains - Dockerized deployment with CI/CD pipeline --- ## 7. Conclusion SmartSDLC enhances modern software engineering by reducing human effort and increasing delivery speed and accuracy. Its modular, AI-driven approach makes it suitable for enterprise adoption, education,

and agile teams.