Project Design Phase-II

Technology Stack (Architecture & Stack)

Date	12 June 2025
Team ID	LTVIP2025TMID31167
Project Name	SmartSDLC – AI-Enhanced Software Development Lifecycle
Maximum Marks	4 Marks

Technical Architecture

The SmartSDLC platform follows a modular AI-driven architecture. The system processes unstructured software requirements and automates several SDLC stages including requirement classification, code generation, bug fixing, test creation, code summarization, and chatbot support. The system uses IBM Granite LLM, deployed via Google Colab using a Gradio-based frontend.

Table-1: Components & Technologies

S.No	Component	Description	Technology Used
1.	User Interface	Web interface to interact with application	HTML, CSS, Gradio, JavaScript
2.	Application Logic-1	SDLC phase classification and task orchestration	Python
3.	Application Logic-2	Natural Language to Code, Bug Fixing, Summarization, Chatbot	IBM Watsonx Granite LLM
4.	Application Logic-3	Chatbot conversation handling and state management	LangChain (optional extension), Python
5.	Database	Temporary in-memory data handling	Pandas DataFrame (in-memory), SQLite (optional)
6.	Cloud Database	Not used in prototype, planned for production	IBM DB2, IBM Cloudant (planned)
7.	File Storage	Upload/parse PDF files	Google Colab local storage
8.	External API-1	(Future Scope) e.g., GitHub Copilot integration	GitHub REST APIs (optional)

9.	External API-2	Not applicable currently	–
10.	Machine Learning Model	AI-based generation, classification, summarization	IBM Watsonx Granite LLM (granite-3.3-2b-instruct)
11.	Infrastructure (Server / Cloud)	Executed and hosted for demo in cloud environment	Google Colab (Backend), Gradio Share (Frontend URL)

Table-2: Application Characteristics

S.No	Characteristics	Description	Technology Used
1.	Open-Source Frameworks	Frameworks/libraries used for UI, model, and PDF processing	Gradio, PyPDF2, Transformers
2.	Security Implementations	Input validation, secure file upload handling	Python Exception Handling, File Limits
3.	Scalable Architecture	Modular, component-wise expansion possible	Microservices-ready with REST-compatible APIs
4.	Availability	Hosted using Google Colab; always accessible via shared URL	Gradio Share Service
5.	Performance	Optimized model prompts, efficient text extraction, lightweight frontend	Transformers pipeline, GPU-backed Colab

References

- C4 Model Architecture
- IBM AI Architecture Patterns
- IBM Cloud Architecture
- AWS Architecture Examples
- How to Draw Useful Architecture Diagrams