Project Design Phase-II
Technology Stack (Architecture & Stack)

Date	27 June 2025
Team ID	LTVIP2025TMID29288
Project Name	SmartSDLC – AI-Enhanced Software Development Lifecycle
Maximum Marks	4 Marks

A. Technical Architecture Overview

The architecture integrates:

- **Frontend (Streamlit)** for user interaction.
- **Backend (FastAPI)** for routing and logic handling.
- **IBM Watsonx (Granite Models)** and **LangChain** for AI capabilities.
- **External services** like GitHub for code repository actions.
- **PDF processing** via PyMuPDF and feedback logging via file/DB.

The architecture includes local deployment for initial phases and is designed for future cloud scalability.

B. Components & Technologies

S.No	Component	Description	Technology
1	User Interface	Web UI interface for interacting with SmartSDLC features	Streamlit (Python), HTML/CSS
2	Application Logic-1	SDLC automation logic (PDF parsing, prompt generation, AI query, etc.)	Python (FastAPI), PyMuPDF
3	Application Logic-2	AI-based code generation, summarization, bug fixing	IBM Watsonx Granite Models
4	Application Logic-3	AI chatbot for SDLC guidance and Q&A	LangChain, Watsonx AI
5	Database	Stores user feedback, login info, session details	JSON/File-based (or SQLite optional)
6	Cloud Database	For future cloud persistence of data (optional)	IBM Cloudant or IBM DB2
7	File Storage	Requirement documents and generated outputs	Local Filesystem
8	External API-1	AI Foundation Model API (IBM Watsonx)	IBM Watsonx APIs
9	External API-2	GitHub Integration for repo operations	GitHub REST APIs
10	Machine Learning Model	NLP + Code Generation + Classification	granite-13b-chat-v1, granite-20b-code-instruct
11	Infrastructure	Runs locally, but scalable to cloud deployments	Local Server (Uvicorn), IBM Cloud

---

C. Application Characteristics

S.No	Characteristics	Description	Technology
1	Open-Source Frameworks	Frameworks used for full-stack development	FastAPI, Streamlit, LangChain, PyMuPDF
2	Security Implementations	Hashed passwords, API key management using environment variables (.env)	bcrypt, dotenv, OAuth (planned), HTTPS
3	Scalable Architecture	Microservices with clear routing and modular services	FastAPI modular routers, service layers
4	Availability	Accessible locally, cloud-compatible with CI/CD and Docker in future plans	Uvicorn, IBM Cloud Foundry (future)
5	Performance	Fast inference via optimized AI calls, async FastAPI routes, lightweight UI	Async FastAPI, Caching (planned), Streamlit