**Project Design Phase-II**

**Technology Stack (Architecture & Stack)**

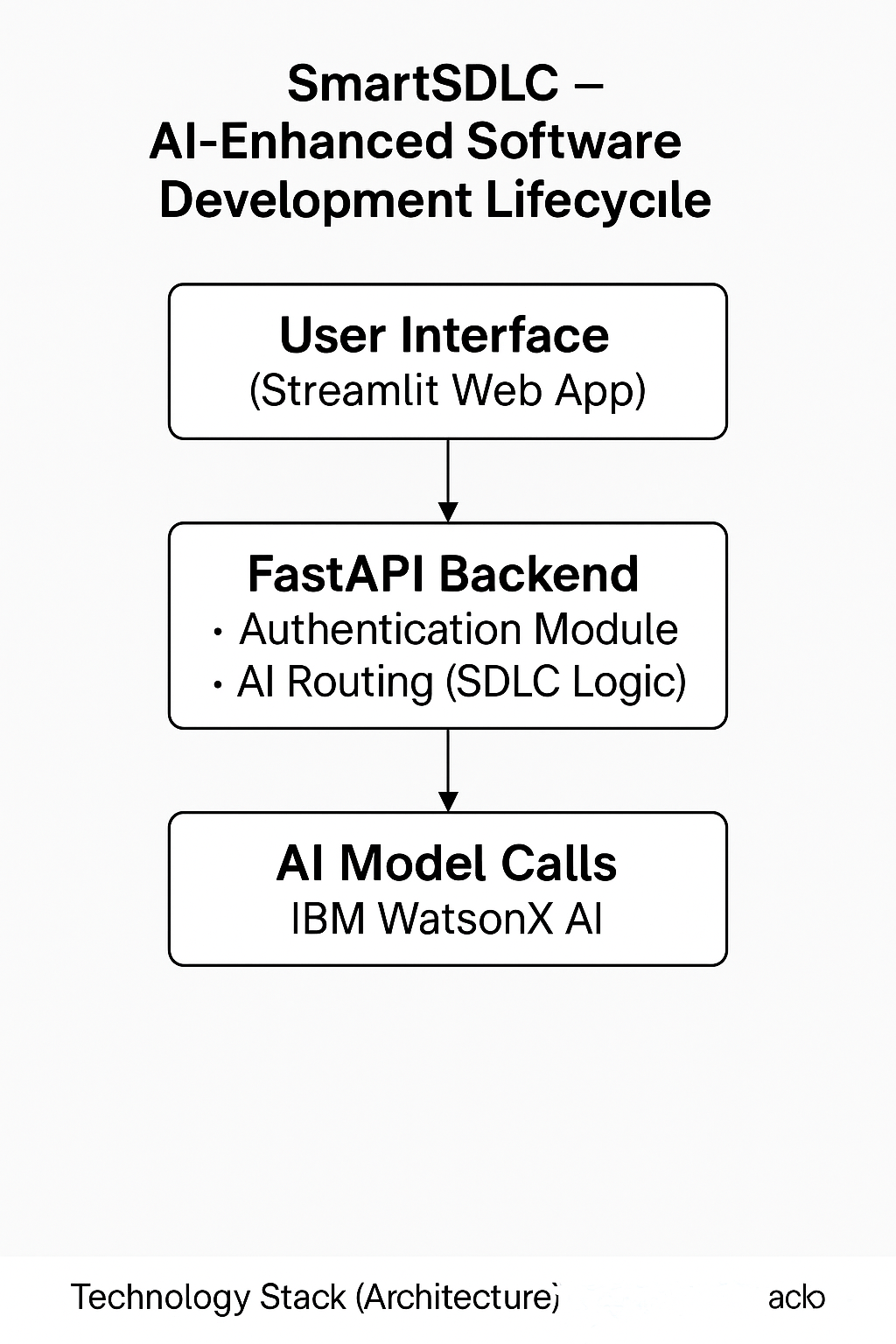
|  |  |
| --- | --- |
| Date | 25 June 2025 |
| Team ID | LTVIP2025TMID30212 |
| Project Name | SmartSDLC – AI-Enhanced Software Development Lifecycle |
| Maximum Marks | 4 Marks |

**Technical Architecture:**

SmartSDLC is a modular AI-enhanced platform that integrates Streamlit for the frontend, FastAPI for backend services, and IBM Watsonx for AI-driven logic. It follows a scalable microservices-inspired architecture with logical separation of frontend, backend APIs, and AI services.

**Example: Order processing during pandemics for offline mode**

**Reference:** [**https://developer.ibm.com/patterns/ai-powered-backend-system-for-order-processing-during-pandemics/**](https://developer.ibm.com/patterns/ai-powered-backend-system-for-order-processing-during-pandemics/)



**Table-1 : Components & Technologies:**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Component** | **Description** | **Technology** |
|  | User Interface | Web app for users to interact with SDLC features | Streamlit (Python), HTML, CSS |
|  | Application Logic-1 | Handles user prompt routing, SDLC operations | Python, FastAPI |
|  | Application Logic-2 | AI text/code processing using generative AI | BM Watsonx Granite 3.3 8B Instruct Model |
|  | Application Logic-3 | Chatbot assistant for user queries | |  | | --- | |  |  |  | | --- | | LangChain + WatsonX API | |
|  | Database | No traditional DB used | N/A |
|  | Cloud Database | Database Service on Cloud | N/A |
|  | File Storage | Temporary file handling (PDF uploads) | Local FileSystem |
|  | External API-1 | IBM Foundation Model API for text generation | WatsonX LLM API |
|  | External API-2 | None used | N/A |
|  | Machine Learning Model | Generates test cases, summaries, and fixes code | IBM WatsonX Foundation Models |
|  | Infrastructure (Server / Cloud) | Application runs locally with option for cloud deployment | Localhost / IBM Cloud |

**Table-2: Application Characteristics:**

| **S.No** | **Characteristics** | **Description** | **Technology** |
| --- | --- | --- | --- |
|  | Open-Source Frameworks | FastAPI, Streamlit, Pydantic | Python-based OSS |
|  | Security Implementations | SHA-256 password check, .env for secrets | hashlib, dotenv |
|  | Scalable Architecture | Modular design allows separation of concerns; loosely coupled services | 3-Tier Architecture |
|  | Availability | Can be containerized and deployed on cloud platforms | IBM Cloud, Docker (optional extension) |
|  | Performance | Lightweight framework, async routes, reusable service layer | FastAPI, Streamlit, Uvicorn |