**Project Design Phase-II**

**Solution Requirements (Functional & Non-functional)**

|  |  |
| --- | --- |
| Date | 27 June 2025 |
| Team ID | LTVIP2025TMID34066 |
| Project Name | SmartSDLC – AI-Enhanced Software Development Lifecycle |
| Maximum Marks | 4 Marks |

**Functional Requirements:**

The Smart SDLC must provide a conversational AI interface that allows developers to interact naturally, making it intuitive to access support across various stages of the software development lifecycle. It should automatically **fix bugs**, **generate clean code**, and **transform natural language requirements into structured specifications**, enabling faster and more accurate development.

The assistant must also support **test case generation** for both unit and integration testing using AI-driven logic, helping teams ensure code reliability with minimal manual effort. It should provide clear, beginner-friendly **explanations of programming concepts** and **code snippets**, supporting continuous learning and onboarding for junior developers.

An integrated **chatbot** must answer technical doubts on programming languages, frameworks, and development tools in real-time, enhancing productivity. Additionally, the platform should track and analyse user feedback and usage data to **continuously improve model accuracy and relevance.**

|  |  |  |
| --- | --- | --- |
| **FR No.** | **Functional Requirement (Epic)** | **Sub Requirement (Story / Sub-Task)** |
| FR-1 | User Registration | Registration through Form  Registration through Gmail  Registration through LinkedIn |
| FR-2 | User Confirmation | Confirmation via Email  Confirmation via OTP |
| FR-3 |  |  |
| FR-4 |  |  |
|  |  |  |
|  |  |  |

**Non-functional Requirements:**

Following are the non-functional requirements of the proposed solution.

|  |  |  |
| --- | --- | --- |
| **FR No.** | **Non-Functional Requirement** | **Description** |
| NFR-1 | **Usability** | Simple, clean UI using Streamlit or similar Python frameworks; beginner-friendly design for developers and students; tooltips and help sections for each functionality. |
| NFR-2 | **Security** | Secure data handling using TLS/SSL; role-based access to developer vs. admin functions; no storage of user code or credentials without consent. |
| NFR-3 | **Reliability** | Stable backend with retry logic for IBM watsonx calls; fallback responses in case of API failure; logs for debugging and monitoring. |
| NFR-4 | **Performance** | Lightweight LLM prompts for faster responses; async processing for generation-heavy tasks like test case generation; optimized queries. |
| NFR-5 | **Availability** | Deployed via cloud platforms like AWS/GCP with load balancing support; responsive during multiple requests; high uptime via managed services. |
| NFR-6 | **Scalability** | FastAPI + Docker structure allows scaling features independently; backend designed for future integration with CI/CD tools, IDE plugins, and GitHub. |