**Ideation Phase**

**Brainstorm & Idea Prioritization Template**

|  |  |
| --- | --- |
| Date | 16 June 2025 |
| Team ID | LTVIP2025TMID29152 |
| Project Name | SmartSDLC – AI-Enhanced Software Development Lifecycle |
| Maximum Marks | 4 Marks |

### ****Team Members & Roles :****

|  |  |
| --- | --- |
| **Name** | **Role** |
| **Vanaja Pulapa** | Frontend Developer (Streamlit), Project Integration |
| **Lakshmidurga Sathi** | Backend Developer (FastAPI), IBM Watsonx AI Integration |
| **Sonti Naga Tulasi** | PDF Upload & SDLC Phase Classification Module |
| **Yalla Manasa Siri** | Testing, Bug Fixer, and Code Summarization Modules |

**Step-1: Team Gathering, Collaboration and Select the Problem Statement :**

**Problem Statement:**  
Manual SDLC processes such as code writing, test creation, debugging, and document classification are repetitive and time-consuming. Students and developers face difficulty performing these tasks efficiently and accurately without automation.

**Motivation:**  
SmartSDLC aims to automate major SDLC stages using IBM Watsonx, reducing time and effort while increasing productivity. It provides features like code generation, test case generation, bug fixing, summarization, and PDF classification through a simple and unified AI interface.

**Step-2: Brainstorm, Idea Listing and Grouping :**

#### ****Initial Ideas:****

* Convert natural language tasks into working code using AI
* Generate test cases automatically
* Auto-detect and fix bugs from uploaded code
* Summarize complex codebases
* Upload and classify PDF content into SDLC phases
* Design an intuitive user interface with Streamlit
* Build a backend using FastAPI to connect frontend and AI
* Add secure login, sign-up, and password reset with OTP

#### ****Grouped into Modules:****

1. **PDF Classifier Module** – Upload PDFs and classify SDLC content
2. **Code Generator Module** – Convert task input to code
3. **Test Case Generator Module** – Auto-generate tests
4. **Bug Fixer Module** – Detect and fix bugs using AI
5. **Summarizer Module** – Generate summaries of source code
6. **Authentication Module** – Secure sign-up/login and password reset
7. **Frontend UI Module** – Built with Streamlit
8. **Backend API Module** – FastAPI for processing and model integration

**Step-3: Idea Prioritization :**

|  |  |  |
| --- | --- | --- |
| **Module** | **Priority** | **Reason** |
| Code Generator | High | Core AI functionality for smart development |
| PDF Classifier | High | Helps understand requirements from uploaded files |
| Test Case Generator | High | Saves time and improves code quality |
| Bug Fixer | Medium | Important but model-dependent |
| Summarizer | Medium | Useful for understanding large codebases |
| Authentication Module | High | Ensures secure access |
| Frontend UI (Streamlit) | High | Essential for a user-friendly interface |
| Backend API (FastAPI) | High | Required to handle all feature logic and AI calls |