**Ideation Phase**

**Brainstorm & Idea Prioritization Template**

| Date | 25 JUNE 2025 |
| --- | --- |
| Team ID | LTVIP2025TMID21159 |
| Project Name | SmartSDLC – AI-Powered Software Development Lifecycle Optimization |
| Maximum Marks | 4 Marks |

**Brainstorm & Idea Prioritization Template:**

**Brainstorming** fosters a collaborative environment to explore innovative ideas. This template records the creative ideation process that led to the SmartSDLC system—a generative AI platform automating key phases of the SDLC.

Reference: <https://www.mural.co/templates/brainstorm-and-idea-prioritization>

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

* **Problem Identified**:  
  Traditional SDLC processes are often time-consuming, manual, and prone to human error. Developers spend excessive time on repetitive tasks such as writing boilerplate code, debugging, and test creation, slowing down overall software delivery.
* **Objective**:  
  To build an AI-powered platform that leverages NLP and generative models to automate critical SDLC stages, improving efficiency, consistency, and developer productivity.

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

|  |  |
| --- | --- |
| **Idea** | **Group** |
| Automate requirement classification using LLMs | Requirements |
| Use LLMs to generate code from plain English | Code Generation |
| Auto-fix bugs by understanding code context | Debugging |
| Generate test cases automatically from code logic | Testing |
| Summarize large codebases for better understanding | Documentation |
| Integrate an SDLC-aware chatbot for developer assistance | Chat Interface |
| Build a Streamlit UI for non-technical users | User Interface |
| Use fallback API calls if GPU is unavailable | Reliability |
| Show GPU status in the UI | System Info |
| Add structured error handling and logging | Reliability & Debugging |

**Step-3: Idea Prioritization**

|  |  |  |  |
| --- | --- | --- | --- |
| **Idea** | **Impact** | **Feasibility** | **Priority** |
| Requirement classification using transformers | High | High | High |
| Code generation with LLMs | High | Medium | High |
| Bug fixing module | Medium | High | High |
| Test case generation | Medium | High | High |
| Code summarization | Medium | Medium | Medium |
| SDLC chatbot | Medium | Medium | Medium |
| Streamlit frontend UI | High | High | High |
| Fallback mechanism (Groq API) | Medium | High | Medium |
| GPU status display | Low | High | Low |
| Structured error responses + Logging | High | High | High |