

# **Project Title : AI-Powered SmartSDLC Assistant for Software Development Automation**

**Team ID : NM2025TMID11592**

Team Size : 4

Team Leader : pooja p

Team member : Deepika S Type your text

Team member : jayamani p

Team member : Kokila p

## **1. INTRODUCTION**

### **1.1 Project Overview**

The SmartSDLC project aims to automate and enhance various stages of the software development life cycle (SDLC) using Artificial Intelligence. By integrating tools such as LangChain, IBM Watsonx, and FastAPI, this system supports developers in tasks like requirement analysis, code generation, test case creation, bug fixing, documentation, and GitHub integration. The system acts as a virtual assistant that streamlines the development workflow, improves productivity, and reduces manual effort.

### **1.2 Purpose**

The purpose of this project is to develop an AI-powered tool that can assist developers and project managers by automating repetitive tasks in the SDLC. The goal is to increase software quality, reduce time-to-market, and empower teams with intelligent suggestions and actions at each development stage.

## **2. IDEATION PHASE**

### **2.1 Problem Statement**

Traditional SDLC activities such as writing boilerplate code, creating test cases, documenting logic, and debugging are time-consuming and error-prone. Developers spend valuable time on repetitive tasks rather than focusing on innovation. There is a need for a smart assistant that can automate and support these activities intelligently.

## 2.2 Empathy Map Canvas

| Category       | Details  |
|----------------|--|
| Sees           | Repetitive tasks, tight deadlines                                |
| Hears          | Complaints about delays, bugs, missing documentation             |
| Thinks & Feels | Stressed, time-constrained, frustrated                           |
| Says & Does    | "I wish this part was automated", uses Google for fixes          |
| Pains          | Time loss, inconsistency, manual effort                          |
| Gain           | AI-generated suggestions, faster delivery, improved code quality |
|                |  |

## 2.3 Brainstorming Ideas

- LLM-based requirement analyzer
- Natural language to code generation
- Test case generator
- Bug detector and fixer
- Auto documentation tool
- GitHub integration for version control
- AI chatbot assistant for developers

## 3. REQUIREMENT ANALYSIS

### 3.1 Customer Journey Map

| Step | User Action              | System Response        |
|------|--------------------------|------------------------|
| 1.   | Developer logs in        | Shows Dashboard        |
| 2.   | Enters Requirements      | AI analyzes input      |
| 3.   | Requests Code/tests/docs | AI generates output    |
| 4.   | Review Suggestions       | Can accept or modify   |
| 5.   | Pushes to Github         | System commits files   |
| 6.   | Follows up with chatbot  | Gets help/debug advice |

### 3.2 Solution Requirements

#### Functional Requirements

- Accepts textual requirements
- Generates code/test/docs
- Detects bugs in code
- Integrates with GitHub
- Chatbot for Q&A support

#### Non-Functional Requirements

- Scalable and responsive
- Modular and maintainable code
- Accuracy in AI predictions
- Secure API communication

### 3.3 Data Flow Diagram

User → FastAPI → LangChain → IBM Watsonx → Response → GitHub/API/UI

### 3.4 Technology Stack

- **Backend:** FastAPI (Python)
- **AI Integration:** LangChain + IBM Watsonx
- **LLMs:** IBM Watsonx LLMs / Open Source Models
- **Frontend (optional):** React / Streamlit
- **Version Control:** GitHub
- **Deployment:** Localhost or cloud VM
- **Database (optional):** SQLite / JSON logs

## 4. PROJECT DESIGN

### 4.1 Problem-Solution Fit

SmartSDLC solves the problem of slow, manual SDLC processes by intelligently automating key stages. It aligns with developer pain points and offers a smart, supportive assistant that reduces effort and speeds up delivery.

### 4.2 Proposed Solution

An AI assistant for SDLC built using FastAPI backend that connects to LLMs through LangChain and Watsonx. It supports end-to-end functions from requirement input to GitHub integration, with optional chatbot features.

### 4.3 Solution Architecture

[User]

↓

[Frontend (optional)] → [FastAPI Backend] → [LangChain] → [Watsonx LLM]

↓

[GitHub/Output]

## 5. PROJECT PLANNING & SCHEDULING

### 5.1 Project Planning

| Week   | Activities   |
|--------|--|
| Week 1 | Requirement gathering, empathy map, tool selection |
| Week 2 | Backend setup (FastAPI), LangChain integration     |
| Week 3 | AI prompt tuning, test case & code generation      |
| Week 4 | GitHub API integration, chatbot assistant, testing |

## 6. FUNCTIONAL AND PERFORMANCE TESTING

### 6.1 Performance Testing

- API response time: < 3 seconds for typical prompts
- LLM response accuracy tested across 10+ queries
- Stress tested using multiple sequential inputs
- GitHub API validated with commit operations
- Manual validation for code/test/doc generation


## 7. RESULTS

### 7.1 Output Screenshots

## SmartSDLC – AI Requirement Analyzer

Upload a **requirement document** (PDF) to automatically classify SDLC phases and generate user stories using IBM Watsonx AI.

Upload a PDF file

 Drag and drop file here  
Limit 200MB per file • PDF

Browse files

## SmartSDLC – AI Requirement Analyzer


Upload a **requirement document** (PDF) to automatically classify SDLC phases and generate user stories using IBM Watsonx AI.

Upload a PDF file

 Drag and drop file here  
Limit 200MB per file • PDF

Browse files

 Hema's Resume.pdf 116.5KB 

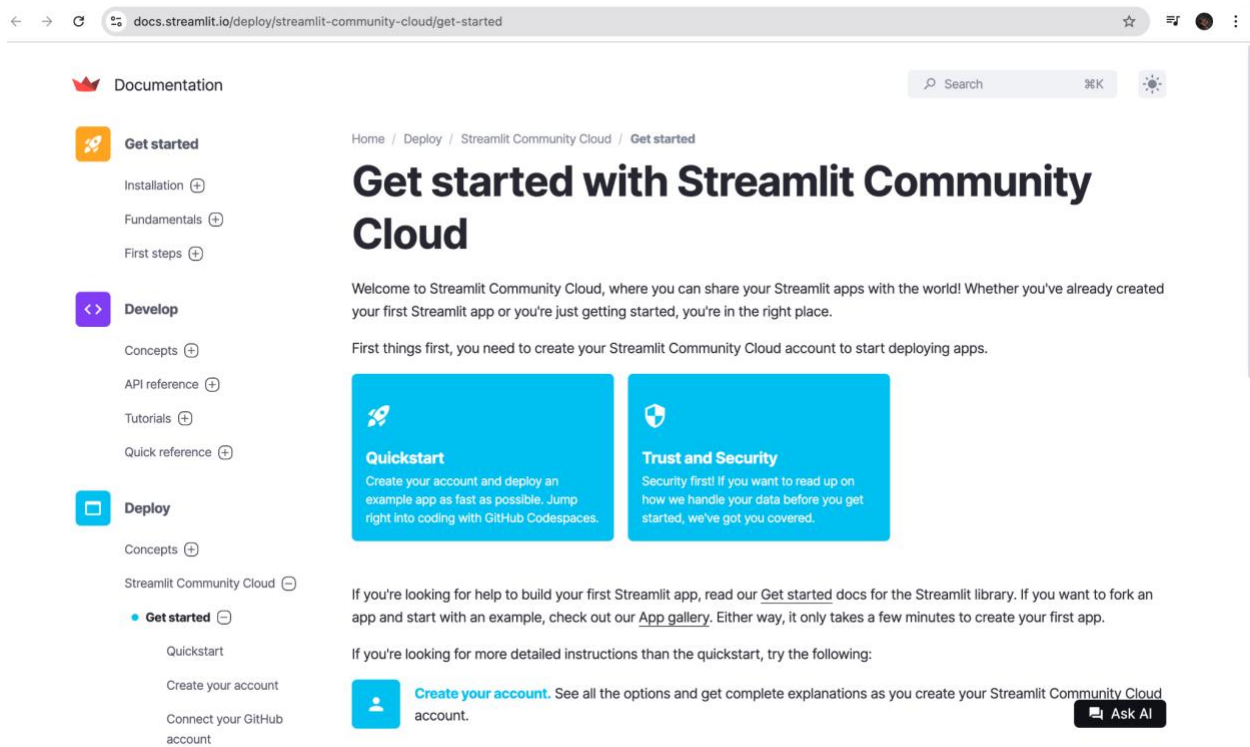
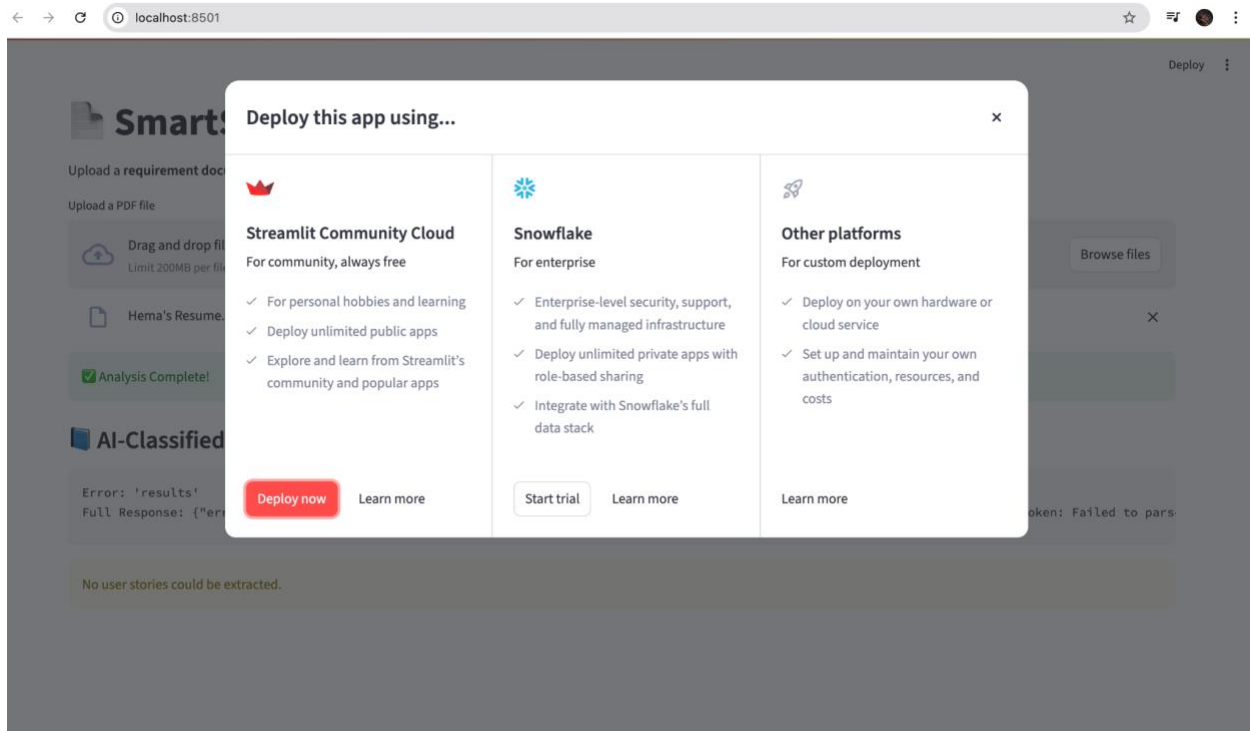
 Analysis Complete!

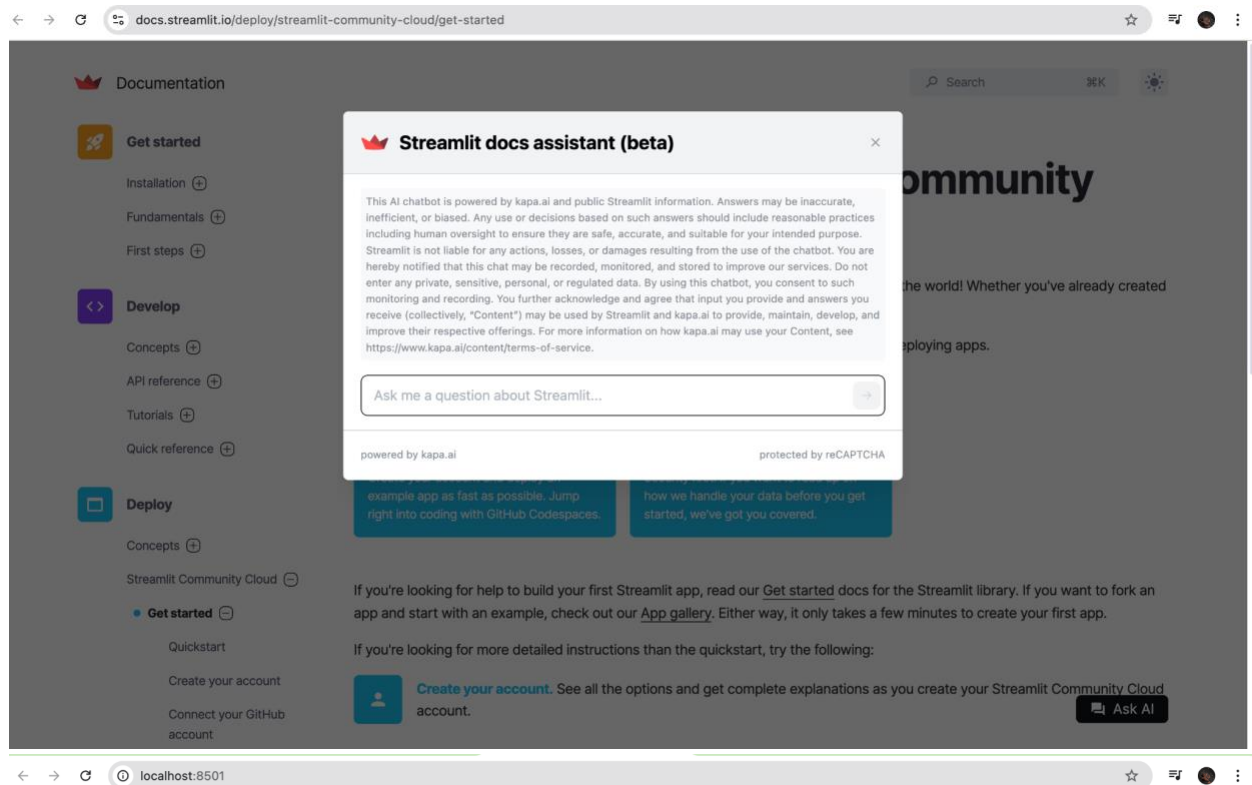
### AI-Classified Text:

Error: 'results'

Full Response: {"errors":[{"code":"authentication\_token\_not\_valid","message":"Failed to authenticate the request due to invalid token: Failed to pars

No user stories could be extracted.





## SmartSDLC – AI Requirement Analyzer

Upload a **requirement document (PDF)** to automatically classify SDLC phases and generate user stories using IBM Watsonx AI.

Upload a PDF file

Drag and drop file here  
Limit 200MB per file • PDF

Hema's Resume.pdf 116.5KB

✓ Analysis Complete!

### AI-Classified Text:

```
Error: 'results'
Full Response: {"errors":[{"code":"authentication_token_not_valid","message":"Failed to authenticate the request due to invalid token: Failed to pars
```

No user stories could be extracted.

Deploy

- Rerun R
- Settings
- Print
- Record a screencast
- About
- Developer options
- Clear cache C

## 8. ADVANTAGES & DISADVANTAGES

### Advantages:

- Reduces manual effort



- Improves SDLC speed and consistency
- Smart, AI-generated support
- Easily extensible to other domains

**Disadvantages:**

- Limited by LLM capabilities
- May require prompt tuning
- Online access required for APIs
- GitHub actions need configuration

## **9. CONCLUSION**

The SmartSDLC project demonstrates how AI can revolutionize software development by automating the SDLC pipeline. It reduces human effort, increases quality, and creates a future-ready development environment leveraging modern LLMs and intelligent systems.

## **10. FUTURE SCOPE**

- UI improvements for production use
- Role-based access and login
- CI/CD pipeline integration
- Support for multi-language coding
- Integration with issue trackers like Jira
- Add XAI for explainability of code predictions