# Project Report: SmartSDLC – AI-Enhanced Software Development Lifecycle

|  |  |
| --- | --- |
| Date | 29 jun 2025 |
| Team ID | LTVIP2025TMID29143 |
| Project Name | Project - SmartSDLC-AI-Enhanced Software Development Lifecycle |
| Maximum Marks |  |

## 1. INTRODUCTION

### 1.1 Project Overview

SmartSDLC is an AI-powered platform that enhances every phase of the Software Development Lifecycle (SDLC). It leverages IBM Watsonx, Granite Foundation Models, and modern frameworks like FastAPI and Streamlit to accelerate software design, improve quality, and automate repetitive tasks.

### 1.2 Purpose

The goal is to provide a smart, scalable, and intelligent development environment that supports requirements analysis, design suggestions, code generation, testing, and documentation with AI assistance.

## 2. IDEATION PHASE

### 2.1 Problem Statement

Traditional SDLC processes are often time-consuming, error-prone, and require extensive manual effort. There is a lack of integration between various stages leading to delays and inconsistencies.

### 2.2 Empathy Map Canvas

- Says: Developers want better tools and automation.  
- Thinks: AI can reduce manual effort.  
- Feels: Frustrated with repetitive tasks.  
- Does: Searches for smart development tools.

### 2.3 Brainstorming

- Use LLMs for generating requirements.  
- Automate test case creation.  
- Integrate AI with Streamlit for live interaction.  
- Provide code explanations, performance analysis, and versioning.

## 3. REQUIREMENT ANALYSIS

### 3.1 Customer Journey Map

User logs in → Selects SDLC phase → Inputs data → Receives AI-generated suggestions → Applies/Edits → Downloads or saves the result

### 3.2 Solution Requirement

- FastAPI Backend  
- Streamlit Frontend  
- IBM Watsonx API  
- SQLite / NoSQL DB

### 3.3 Data Flow Diagram

1. User input via UI  
2. Processed by FastAPI  
3. Watsonx Granite model inference  
4. Output to frontend

### 3.4 Technology Stack

- Frontend: Streamlit  
- Backend: FastAPI, Python  
- AI: IBM Watsonx + Granite LLM  
- Database: SQLite / MongoDB

## 4. PROJECT DESIGN

### 4.1 Problem Solution Fit

SmartSDLC addresses automation gaps in software lifecycles using AI capabilities to reduce effort and improve accuracy.

### 4.2 Proposed Solution

A unified platform that provides intelligent features per SDLC phase such as:  
- Requirement gathering  
- Architecture generation  
- Code assistance  
- Test case creation  
- Performance metrics

### 4.3 Solution Architecture

Frontend (Streamlit) ↔ API Layer (FastAPI) ↔ AI Engine (Watsonx) ↔ DB

## 5. PROJECT PLANNING & SCHEDULING

### 5.1 Project Planning

- Week 1: Requirement gathering and setup  
- Week 2–3: Backend and AI integration  
- Week 4: UI development  
- Week 5: Testing and feedback  
- Week 6: Final deployment

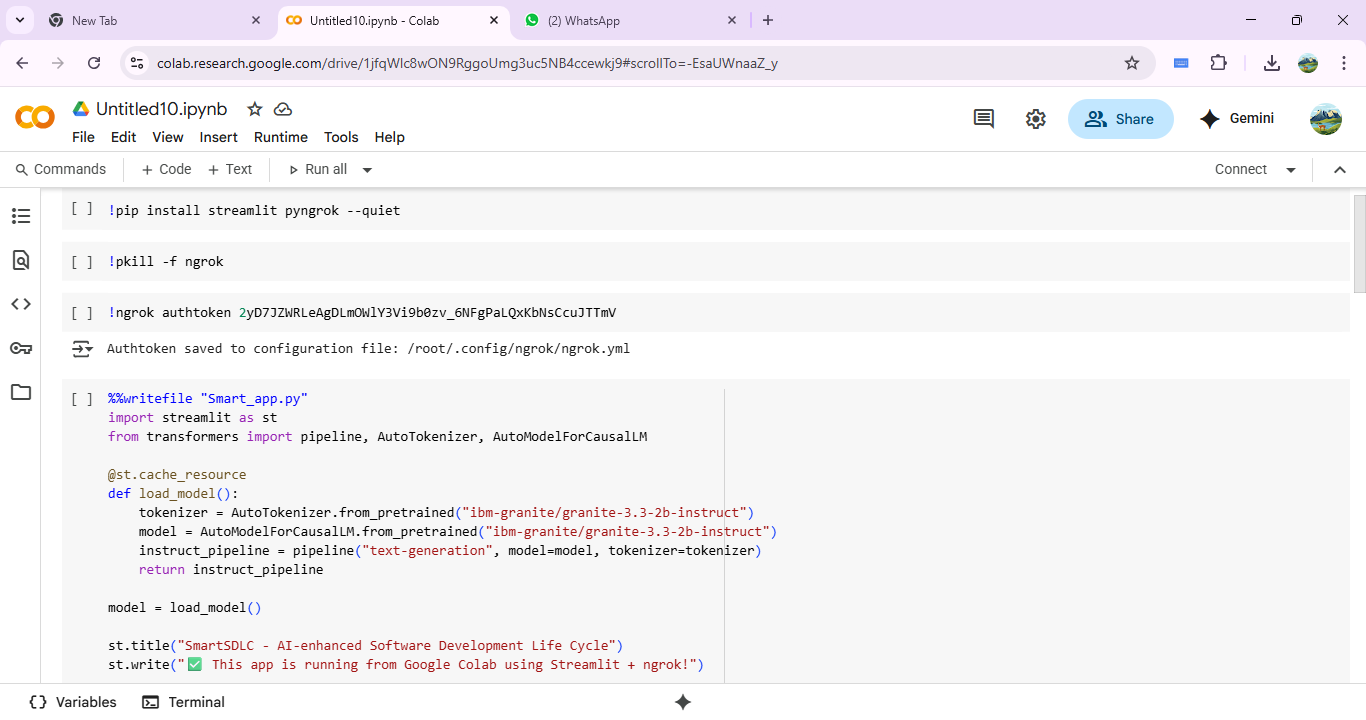
## 6. FUNCTIONAL AND PERFORMANCE TESTING

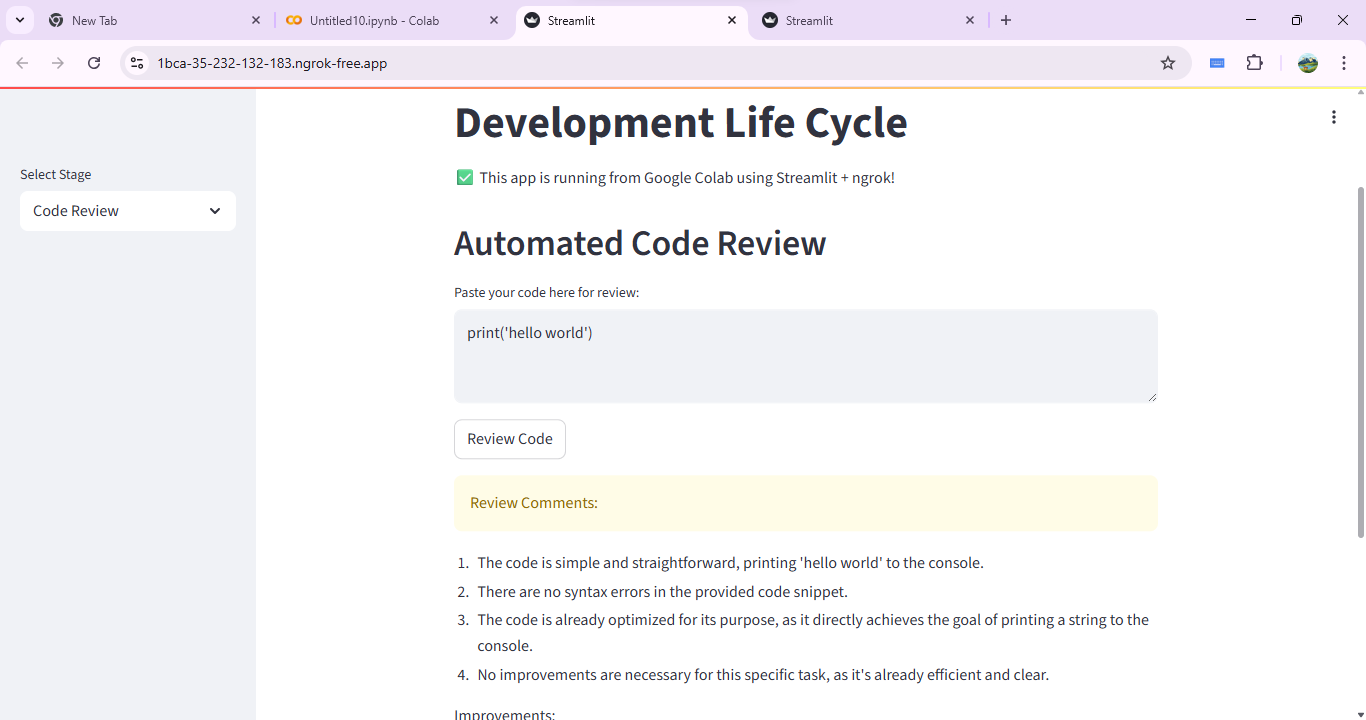
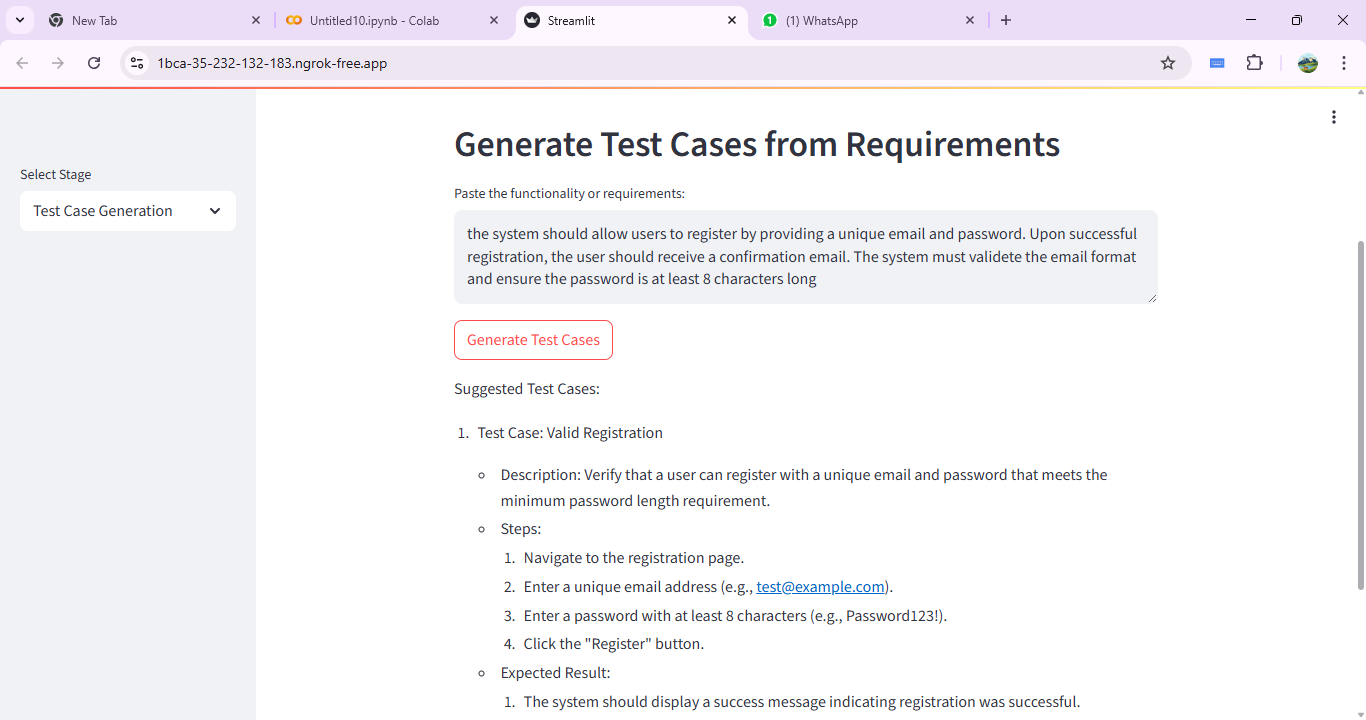
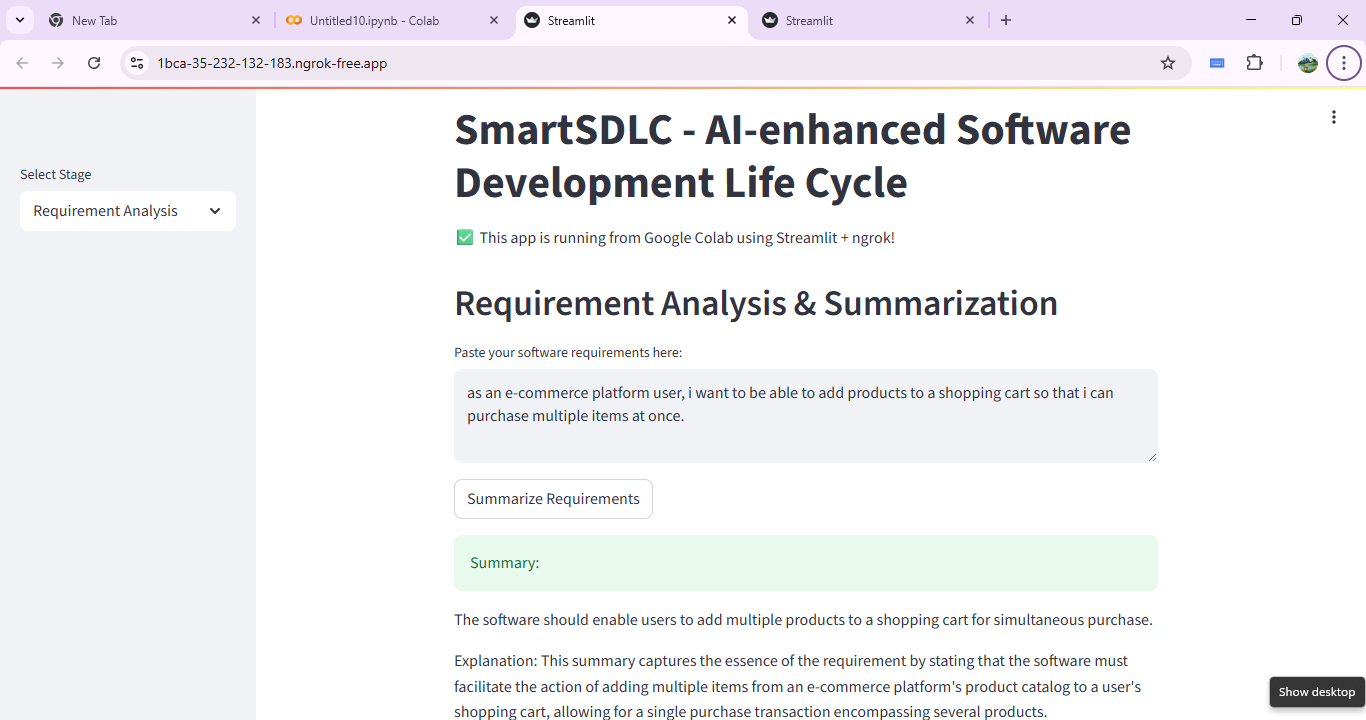
### 6.1 Performance Testing

- API latency < 2 sec  
- Response accuracy benchmarked using predefined prompts  
- UI load test with 50 concurrent users

## 7. RESULTS

### 7.1 Output Screenshots





## 8. ADVANTAGES & DISADVANTAGES

### Advantages

- AI-driven automation  
- Time-saving and productivity boost  
- Easy integration of modules

### Disadvantages

- Requires internet for Watsonx API  
- Initial learning curve for new users

## 9. CONCLUSION

SmartSDLC brings intelligence to software development by automating key processes, improving quality, and reducing the manual workload of developers.

## 10. FUTURE SCOPE

- Integrate CI/CD pipeline tools  
- Add voice assistant  
- Expand to mobile app version

## 11. APPENDIX

- Source Code: Available on GitHub  
- Dataset Link: <https://github.com/GeethaDundu/SmartSDLC-AI-Enhanced-Software->Development-Life-Cycle/blob/main/SmartSDLC.ipynb  
- GitHub & Project Demo Link: <https://github.com/GeethaDundu/SmartSDLC-AI-Enhanced> Software-Development-Life-Cycle