**SmartSDLC – AI-Enhanced Software Development Lifecycle**

**1.Introduction**

• NM Team ID : NM2025TMID08084

• Project title : SmartSDLC – AI-Enhanced Software Development Lifecycle

• Team leader : MANASA D (CBBF14B0CA1AC54A0BF496BE1246758F )

• Team member : LAWRENCIA S (F6336CA640F4719AC5F43984378FBE63 )

• Team member : LEKA G (D756183293248C8BF8B4C3C3D9A6F86A)

• Team member : MATHANIKA M (614BEF8227A4F9EAAAA46D4573E17CCF)

• Team member : MEENA P ( 2CA4666F375400484084C61E039CEEA7 )

**2. Project Overview**

• Purpose :

SmartSDLC uses the Granite model from Hugging Face to speed up software development. It lets users upload PDFs, generate clear requirements, turn prompts into code, create tests, fix bugs, write docs, and chat with an AI helper. The project is deployed in Google Colab using Granite for easy setup and reliable performance.

• Features:

Requirements Generation  
Code Generation  
Automated Testing  
Bug Fixing  
Documentation Creation  
AI-powered Chat Assistance

**3. Architecture**

Frontend (Gradio):  
The frontend is built with Gradio, providing an interface for requirement input, code generation, and bug fixing.

Backend (Google Colab + IBM Granite):  
Google Colab provides the execution environment. IBM Granite models from Hugging Face are used for requirement extraction, natural language processing, and code generation.

**4. Setup Instructions**

Pre requisites:

1. Gradio Framework Knowledge  
2. IBM Granite Models (Hugging Face)  
3. Python Programming Proficiency  
4. Version Control with Git  
5. Google Colab’s T4 GPU Knowledge

Installation Process:

1. Open Google Colab.  
2. Create a new notebook.  
3. Set runtime to T4 GPU.  
4. Install dependencies using '!pip install transformers torch gradio PyPDF2 -q'.  
5. Run the provided SmartSDLC code.  
6. Access the Gradio app via the generated link.

**5. Folder Structure**

project/  
├── smart\_sdlc.ipynb – Main notebook file  
├── requirements.txt – Dependencies  
├── README.md – Project documentation  
└── app/ – Optional additional scripts

**6. Running the Application**

1. Launch the Google Colab notebook.  
2. Run all cells sequentially.  
3. Wait for the model to load.  
4. Click on the Gradio-generated link to access the SmartSDLC app.

**7. API Documentation**

The SmartSDLC application uses Gradio endpoints for interaction. Example endpoints include:  
• /generate\_requirements – Generates requirements from uploaded PDFs  
• /generate\_code – Converts prompts into code  
• /fix\_bugs – Suggests fixes for code  
• /generate\_tests – Creates test cases  
• /generate\_docs – Generates documentation

**8. Authentication**

Currently, the application is deployed in an open environment. For secure deployment:  
• API keys for IBM Granite  
• OAuth2 authentication for user access  
• Role-based access (admin, developer, tester)

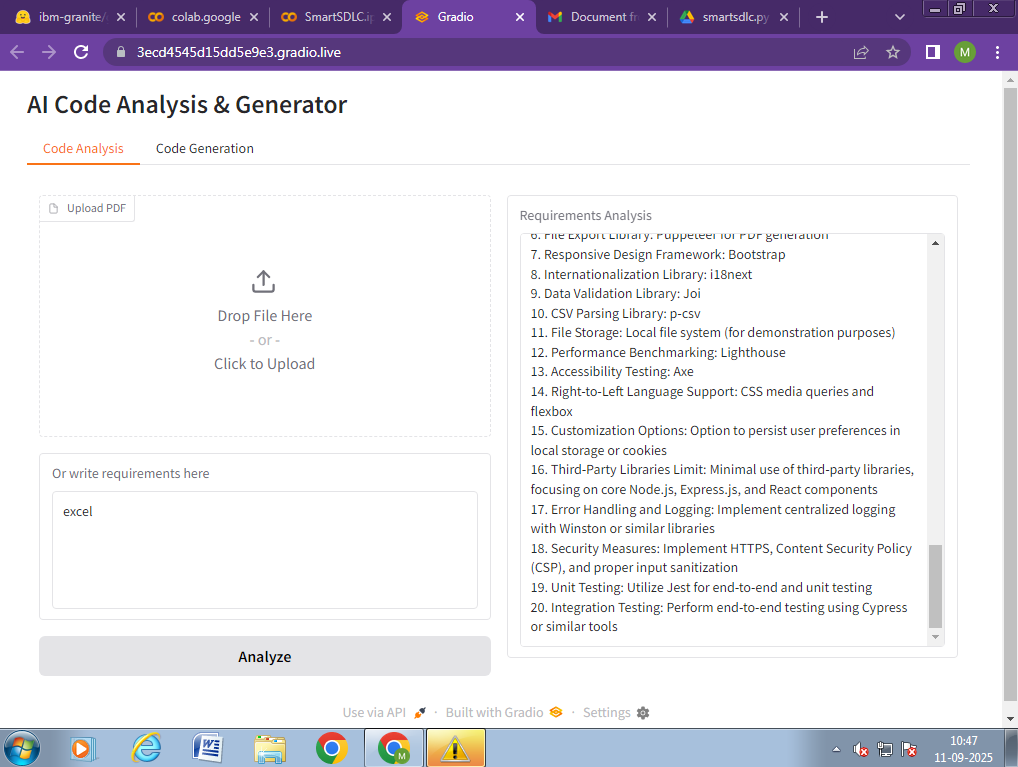
**9. User Interface**

The interface is built with Gradio. It provides:  
• Requirement input forms  
• Code generation panels  
• Bug fixing modules  
• Testing and documentation views

**10. Testing**

Testing includes:  
• Unit Testing: For requirement parsing, code generation.  
• API Testing: Using Gradio endpoints.  
• Manual Testing: Running notebook cells and verifying results.  
• Edge Cases: Invalid inputs, corrupted PDFs.

**11. Screenshots**

****

**12. Known Issues**

• Limited handling of complex software requirements.  
• Accuracy of bug fixes depends on model capabilities.  
• Requires internet for model access from Hugging Face.

**13. Future Enhancement**

• Support for larger projects and multiple modules.  
• Integration with CI/CD pipelines.  
• Enhanced test case generation.  
• Improved bug-fixing accuracy.