SmartSDLC : AI-Enhanced Software Deevelopment Lifecycle

**Team Members** :

Kodali. Navya - 22HU1A4235

Korumalli. Sasi Vardhini - 22HU1A4238

Khagga. Varsha Priya - 22HU1A4234

Kotari. Sravani - 22HU1A4239

Kalyanam. Bhavana - 22HU1A4232

Faculty Mentor: Manoj

# 1. Introduction

**Smart SDLC** (Software Development Life Cycle) refers to a modern, efficient approach to software development that emphasizes **agility, automation, collaboration, and continuous improvement**. It integrates smart tools and practices to streamline each phase—planning, development, testing, deployment, and maintenance—for faster, higher-quality software delivery.

**2. Tools & Technologies**

- IBM Granite (LLM API from IBM watsonx.ai)  
- Python 3, Flask  
- HTML, CSS, JavaScript (AJAX)  
- VS Code  
- Replit / Render for deployment

**3. Technical Architecture**

The system consists of a front end interface (HTML/CSS/JS) where the user types a query. The back end is a Flask server that sends the user query to the IBM Granite model via Open AI - compatible API, fetches the response, and returns it to the user in real time.

# IMG_2564. Modules Implemented

- Model Selection and Architecture: IBM Granite 13B Chat model is used.  
- Core Functionality: Text-based question-answering system.  
- Application Logic: Flask handles the API call and response formatting.  
- UI Design: Simple chat interface for end users.  
- Deployment: Easily deployable on platforms like Render or Replit.

# 5. Sample Code Snippet

Back end logic to query IBM Granite:

response = openai.ChatCompletion.create(  
 model="ibm/granite-13b-chat-v2",  
 messages=[  
 {"role": "system", "content": "You are a helpful medical assistant."}

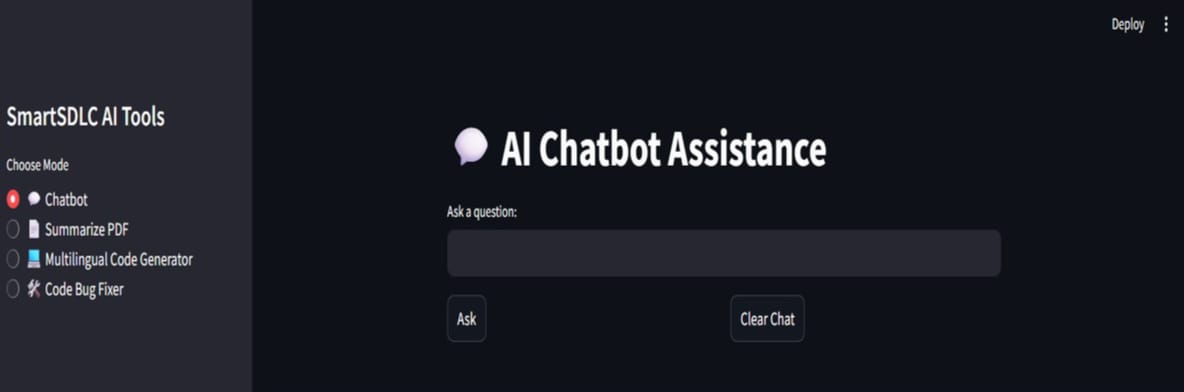
{"role": "user", "content": user\_input}

]

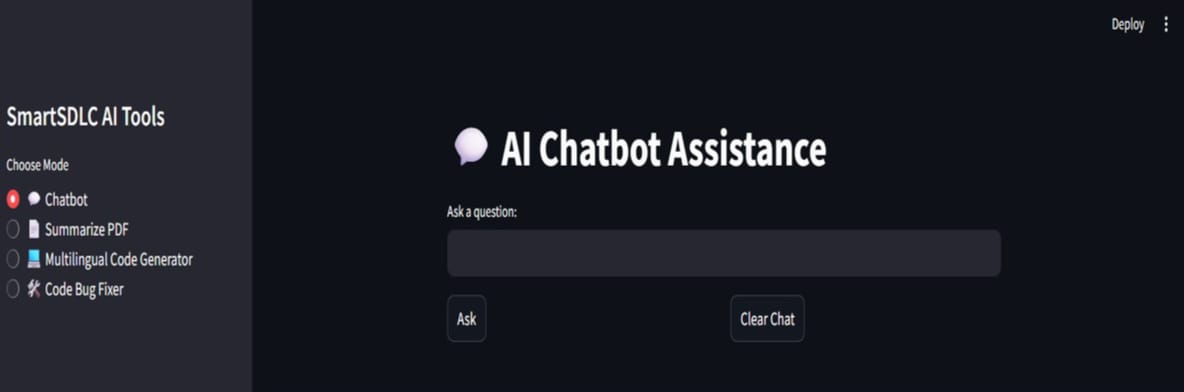
}

**6. Output (Demo)**

Starting interface of the Website:



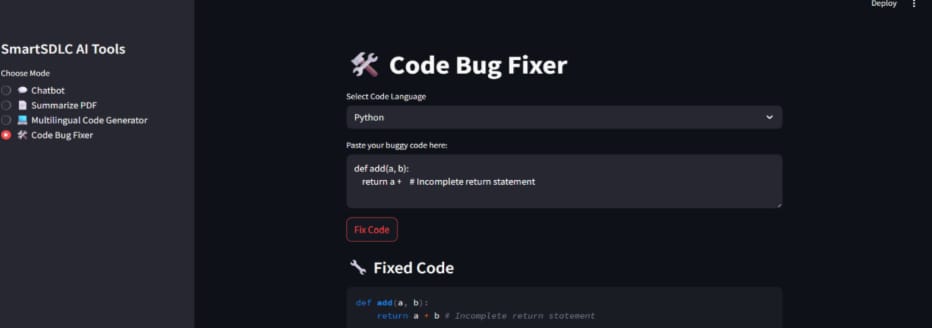
Output of the AI ChatBot Assistance:



Output for Pdf Summarizer:

# WhatsApp Image 2025-06-27 at 8.17.48 PM

Output of the Multilingual Code Generator:



# Advantages & Disadvantages

**Advantages :**

**-** Automation

**-** Faster Delivery

**-** Improved Collaboration

**-** Higher Quality Software

**-** Scalabilty

**Disadvantages:**

**-** High Initial Cost

**-** Tool Complexity

**-** Over dependence on Tools

**-** Security Risks

**-** Requires Skilled Workforce

# 8. Conclusion

The Smart SDLC platform represents a significant advancement in the automation of the Software Development Life cycle by integrating AI - powered intelligence into each phase - from requirement analysis to code generation, testing, bug, fixing, and documentation.

Overall, Smart SDLC not only improves productivity and accuracy but also foundation for future enhancements like CI/CD integration, team collaboration, version control, and cloud deployment

**9.Future Scope**

* Add multilingual support
* Connect with telemedicine services
* Integrate voice input and wearable data

# Apendix

* Source Code: Included
* Dataset Link: IBM cloud Model Documentation
* GitHub & Demo Link: GitHub.com/KodaliNavya-dev/SmartSDLC