SmartSDLC – Software Development Lifecycle AI Assistant

Introduction

In today's fast-paced software industry, managing the Software Development Lifecycle (SDLC) efficiently is crucial to delivering high-quality products. SmartSDLC is an innovative application that integrates Artificial Intelligence (AI) into the development workflow. It assists developers and students in different phases of the SDLC by offering automated code generation, bug fixing, testing, and document classification.

This project leverages IBM Watson AI services to make the development process easier and more interactive. Whether you're a beginner learning how SDLC works or a developer trying to speed up coding tasks, SmartSDLC provides tools that help you achieve your goals faster and more accurately.

Team-members:

- Vikram anand
- Vignesh
- Sivasakthi
- Sanjay

Project Goals

- Simplify the software development process.
- Provide AI-powered solutions for common development challenges.
- Improve learning and understanding of SDLC phases.
- Help students by automating repetitive tasks and providing instant feedback.

Architecture

The SmartSDLC application follows a client-server architecture with the following components:

Frontend

- Built using Streamlit (Python framework) for a responsive and user-friendly interface.

- Users can interact with AI tools via forms and buttons.
- Handles user requests and displays AI-generated results instantly.

Backend

- Developed using FastAPI (Python framework).
- Handles all API requests such as code generation, bug fixing, and PDF classification.
- Integrates with IBM Watson AI services for natural language understanding and classification.

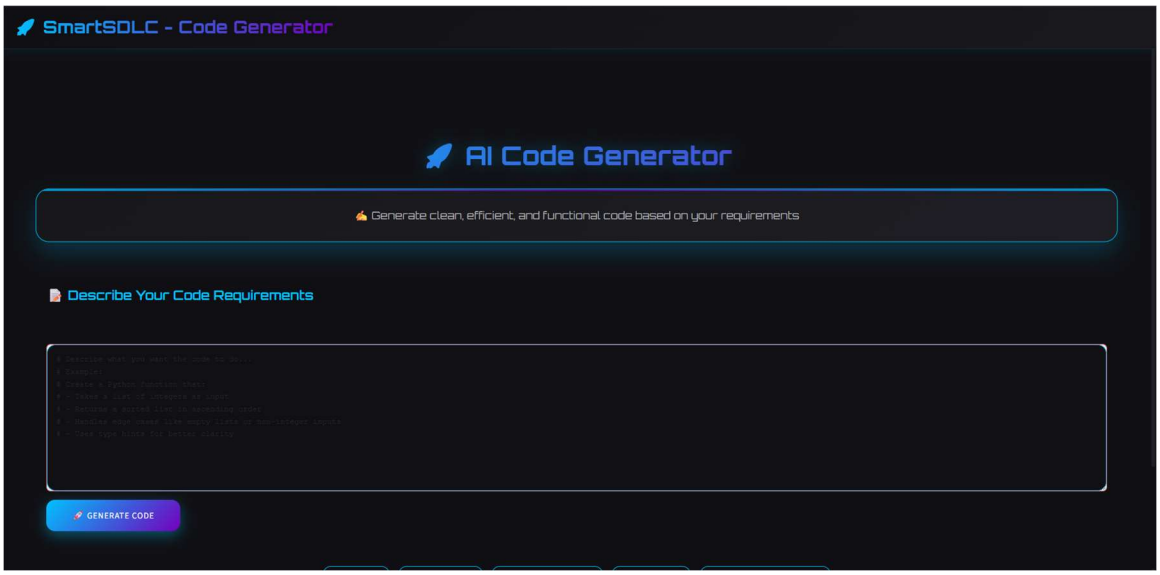
Flow Diagram

User → Frontend → Backend → IBM Watson AI → Backend → Frontend → User

Features – Detailed Overview

1. AI Code Generator

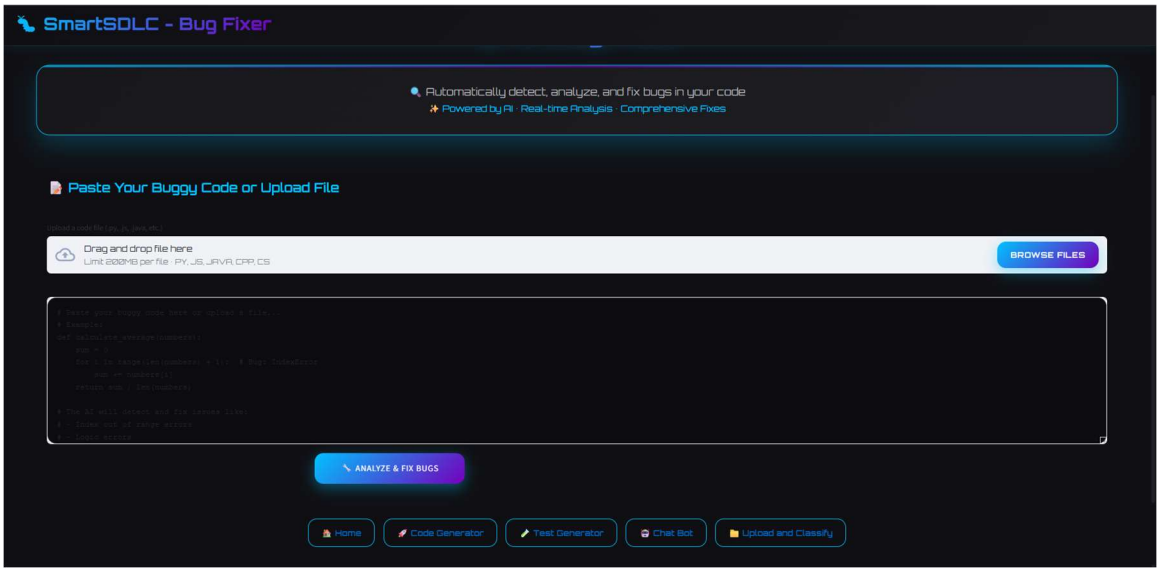
The AI Code Generator helps you by generating functional code snippets based on your input. For example, if you write “function to calculate factorial,” the tool will automatically create the Python code that implements the function



2. Bug Fixer

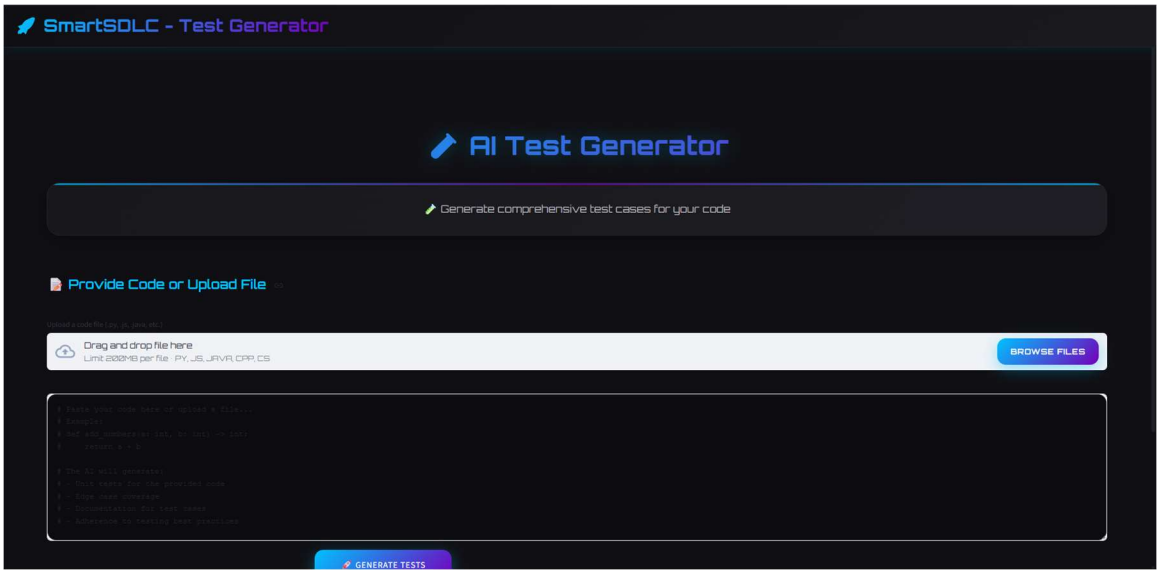
Identifying and fixing bugs can be time-consuming. With the Bug Fixer, you can paste your code, and the AI will detect potential issues, logic errors, or incorrect

syntax. It then suggests fixes along with explanations.



3. Test Case Generator

Testing is an integral part of SDLC. The Test Case Generator helps you create test cases for functions and modules automatically. It saves time and helps you cover edge cases effectively.



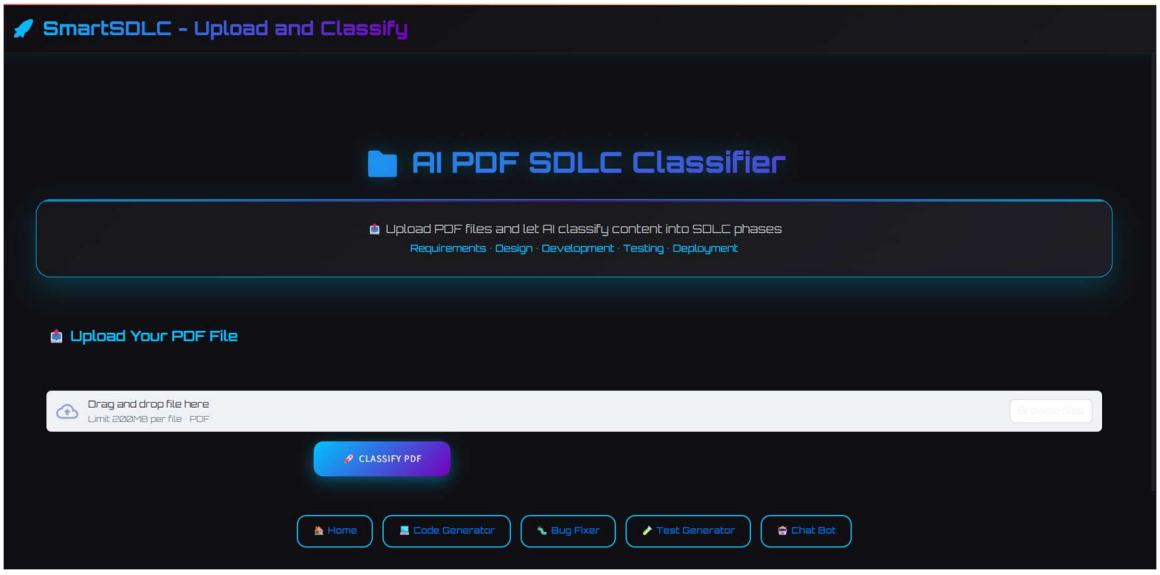
4. Chat Bot

The built-in chatbot is designed to answer development-related questions and provide guidance on SDLC processes. You can ask about methodologies like Agile,

Waterfall, or specific programming concepts like recursion or object-oriented principles.

5. PDF Classification

This feature helps in organizing your project or assignment files. You can upload a PDF document, and the AI will classify it into SDLC phases such as planning, design, testing, or maintenance. This helps students quickly organize their work based on content.



Setup & Installation – Step by Step

Installation Steps

Step 1 – Clone the Repository

Open the terminal and run:

```
git clone https://github.com/Vikram-45/Smart-sdlc.git
cd Smart-sdlc
```

This will download the project files to your system.

Step 2 – Install Backend Dependencies

Navigate to the backend folder:

```
cd backend
pip install -r requirements.txt
```

This will install all required Python libraries like FastAPI, requests, etc.

Step 3 – Install Frontend Dependencies

Open a new terminal window and run:

```
cd frontend
pip install streamlit
```

This will install streamlit libraries and tools needed for the frontend.

Step 4 – Configure API Keys

You need to create an account on IBM Cloud and get the API credentials for Watson services. After obtaining the API key, update the configuration file in the backend.

backend/.env

Replace placeholder values with your actual credentials.

Step 5 – Run the Backend Server

Start the backend server by running:

```
python app.py
```

The backend will start listening on http://127.0.0.1:8000.

Step 6 – Run the Frontend Server

Start the frontend server by running:

Streamlit run login.py

The frontend will open in your default browser at http://localhost: 8501

Usage – How to Work with SmartSDLC

1. Accessing the Interface

Open your browser and go to:
http://localhost: 8501

You'll see the SmartSDLC dashboard with various options such as Code Generator, Bug Fixer, Test Case Generator, Chat Bot, and PDF Classification.

2. Generating Code

1. Click on "Code Generator."
2. Enter your requirement in the input box.
3. Click "Generate."
4. The AI will display the code snippet which you can copy and use.

3. Fixing Bugs

1. Click on "Bug Fixer."
2. Paste your code in the text area.
3. Click "Analyze."
4. The tool will highlight errors and suggest corrections.

4. Generating Test Cases

1. Click on "Test Case Generator."
2. Enter the module or function name.
3. Click "Generate Test Cases."
4. The tool will list relevant test cases.

5. Using the Chat Bot

1. Click on "Chat Bot."
2. Type your query like "Explain Agile methodology."
3. Get instant answers and explanations.

6. PDF Classification

1. Click on "Upload PDF."
2. Choose a PDF file from your computer.

3. Click “Classify.”
4. The tool will classify it into one of the SDLC categories and display the result.

Conclusion

SmartSDLC is a comprehensive tool designed to assist students and developers in understanding and implementing software development processes. By integrating AI-powered solutions like code generation and bug fixing, it reduces development time and improves efficiency. This project not only helps in practical learning but also enhances problem-solving skills and technical understanding.