Smart SDLC - Project Documentation

1. Introduction

- **Project Title: Smart SDLC**

- **Team leader: Amala sweety**
- **Team member : Abinaya.I**
- **Team member: Divya.M**
- **Team member : Aishwarya.K**

Smart SDLC (Software Development Life Cycle) is an AI-powered tool designed to automate and assist in requirement analysis and code generation. It simplifies the software development workflow by analyzing requirements, generating insights, and assisting developers with auto-generated code structures.

2. Project Overview

• Purpose:
The purpose of Smart SDLC is to enhance the efficiency of software development by leveraging AI to interpret software requirements and generate useful insights and code snippets. This tool bridges the gap between requirement analysis and code implementation.

• Features:
- Requirement Analysis (text or PDF upload)
- Automated Code Generation
- AI-based suggestion system
- Gradio-based User Interface
- Real-time analysis and feedback

3. Architecture

• Frontend: Gradio-based UI with tabs for Requirement Analysis and Code Generation
• Backend: Python-based logic for requirement parsing and code generation
• AI Integration: Hugging Face models for natural language understanding and code assistance
• Hosting: Hugging Face Spaces for accessibility and deployment

4. Setup Instructions

Prerequisites:
- Python 3.9 or later
- pip and virtual environment tools
- Internet access

Installation Process:
- Clone the repository
- Install dependencies from requirements.txt
- Run the Gradio app (app.py)
- Open the Hugging Face Space URL

5. Folder Structure

app.py : Main application entry point
- requirements.txt : Dependencies
- /ui : Frontend Gradio components
- /api : Backend modules for requirement parsing and code generation
- /assets : Screenshots and static files

6. Running the Application

- Launch the backend by running: python app.py
- Open Hugging Face Spaces hosted link
- Upload PDF or enter requirements text
- View analyzed requirements
- Generate code snippets automatically

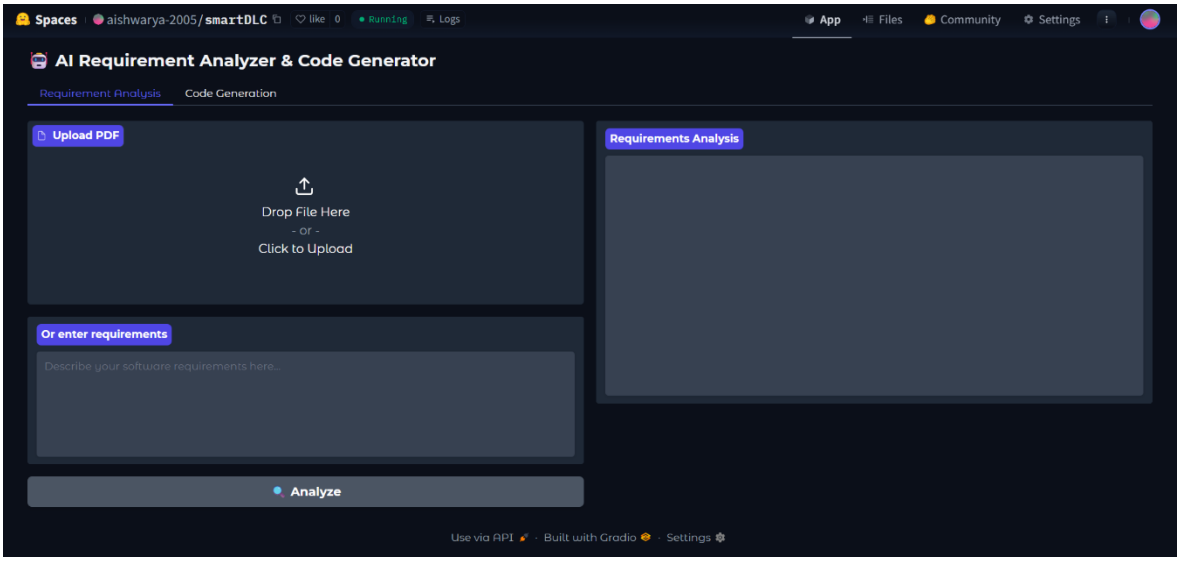
7. User Interface

The Smart SDLC interface includes:
- Tab for Requirement Analysis (PDF upload or text input)
- Tab for Code Generation
- Output panels for displaying analysis and generated code
- Clean design for easy navigation and usage

8. Testing

Testing performed includes:
- Unit Testing: Requirement parsing logic
- Manual Testing: Interface workflow
- API Testing: Model response accuracy
- Edge Case Handling: Invalid inputs, incomplete requirements

9. Screenshots and demo link



Demo link

https://aishwarya-k-2005.github.io/smartsdlc-1/

10.Future Enhancements

- Support for multiple programming languages
- Integration with IDEs for real-time coding assistance
- Advanced NLP for deeper requirement understanding
- Collaboration features for teams