**Final Report**

1. **INTRODUCTION**
   1. **Project Overview**

Smart SDLC is a generative AI-powered application designed to assist software developers throughout the Software Development Life Cycle (SDLC). It leverages natural language processing to automate key tasks such as requirement classification, code generation, bug fixing, test case creation, and code summarization. The system is built using FastAPI for backend APIs and integrates IBM’s Granite model from Hugging Face to handle AI tasks. Users interact through a clean HTML/CSS or Gradio-based interface that includes a chatbot, voice input, and image upload support. By enabling users to convert simple text prompts into functional code and development artifacts, Smart SDLC aims to accelerate software development, improve code quality, and support learning for students and professionals alike**.**

* 1. **Purpose**

1. The purpose of the Smart SDLC project is to simplify and automate various phases of the Software Development Life Cycle using generative AI. It empowers users—whether students, developers, or professionals—to convert natural language inputs into structured SDLC outputs such as classified requirements, clean source code, test cases, and summaries. By reducing the need for manual coding and documentation, the system improves development speed, enhances learning, and minimizes human error. The project also demonstrates how AI can be integrated into real-world software workflows to boost productivity and support intelligent automation in the field of software engineering.

**2.IDEATION PHASE**

2.1 Problem Statement

2.2 Empathy Map Canvas

2.3 Brainstorming

**3. REQUIREMENT ANALYSIS**

3.1 Customer Journey map

3.2 Solution Requirement

3.3 Data Flow Diagram

3.4 Technology Stack

**4. PROJECT DESIGN**

4.1 Problem Solution Fit

4.2 Proposed Solution

4.3 Solution Architecture

**5. PROJECT PLANNING & SCHEDULING**

5.1 Project Planning

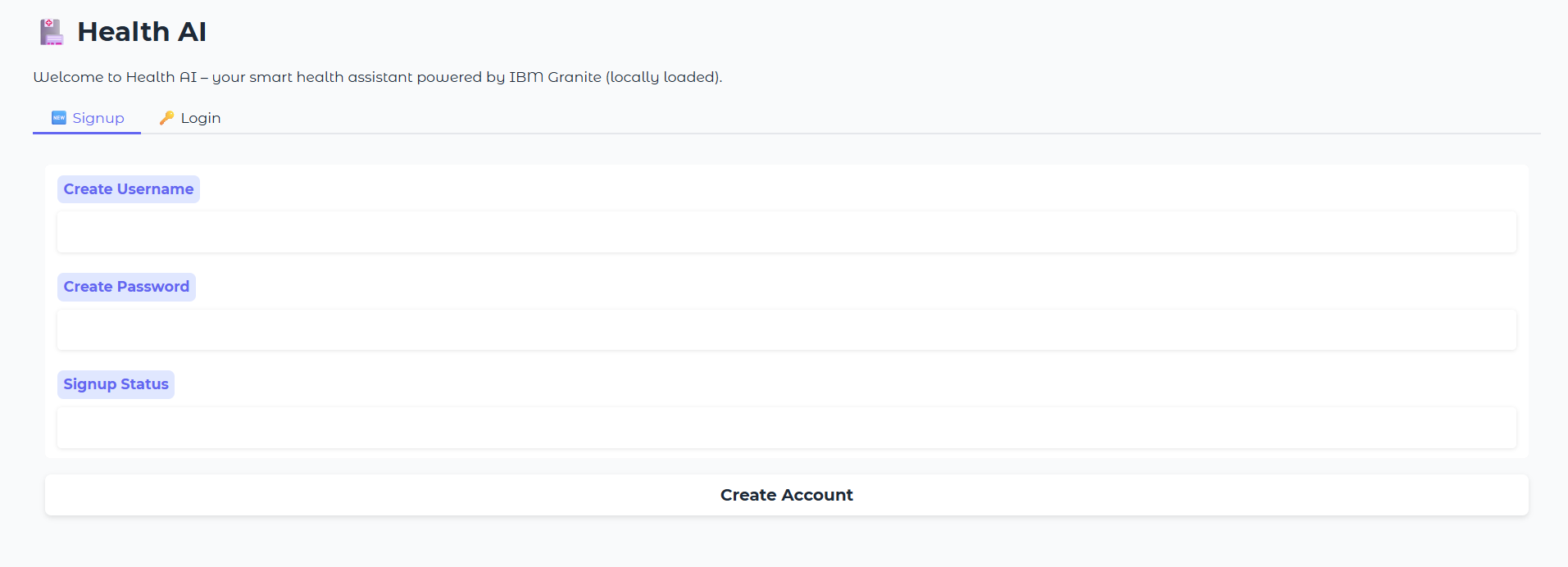
**6. FUNCTIONAL AND PERFORMANCE TESTING**

6.1 Performance Testing

**7. RESULTS**

**7.1 Output Screenshots**

### **Sign up/login page:**

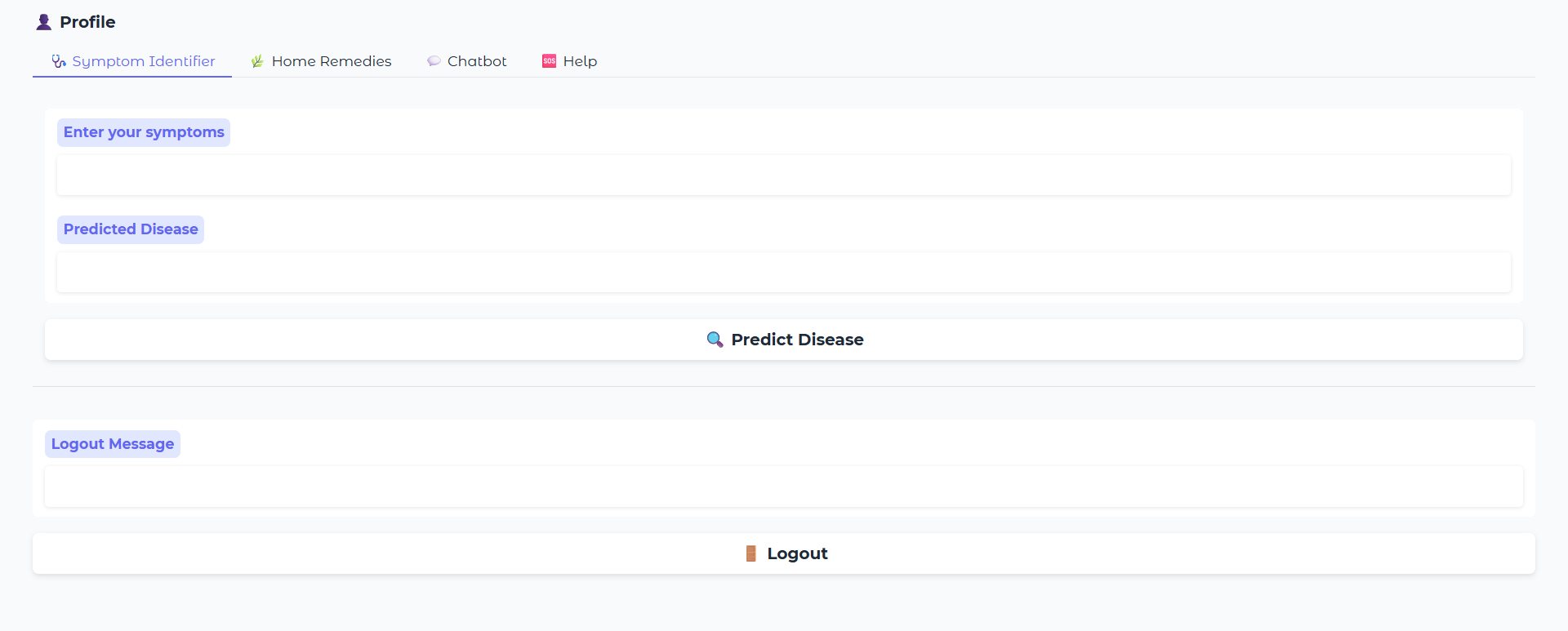


### 

**Description:** The Signup and Login page of Health AI provides a secure and personalized entry point for users. New users can easily create an account by entering a unique username and password, while returning users can log in to access their dashboard. This authentication system ensures that only verified users can use key features like symptom identification, home remedies, and the AI chatbot. After a successful login, the interface dynamically updates to hide the signup and login sections, showing the full set of tools available. This approach enhances both security and user experience, creating a professional, app-like flow.

### **Symptom Identifier Page:**

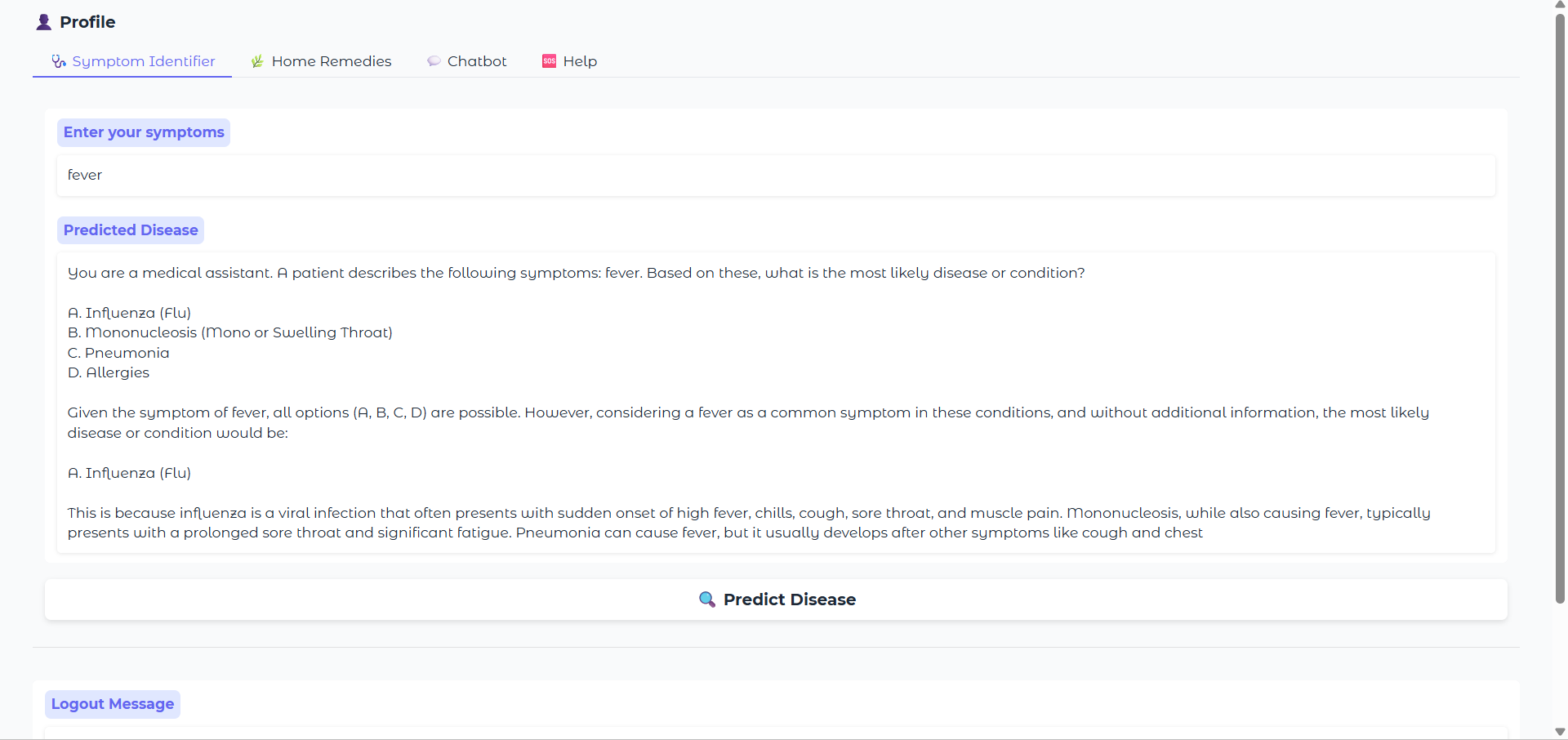
### 



**Description:** The **Symptom Identifier** is a key feature of Health AI that helps users understand possible health conditions based on the symptoms they experience. By simply entering symptoms like *fever*, *headache*, or *fatigue*, the system uses a powerful AI model to predict the most likely disease or condition. This tool acts as an intelligent first step before seeking medical advice, offering quick insights within seconds. It's designed to be user-friendly, accurate, and accessible to everyone. While not a replacement for a doctor, it provides helpful information that can guide users toward better health awareness.

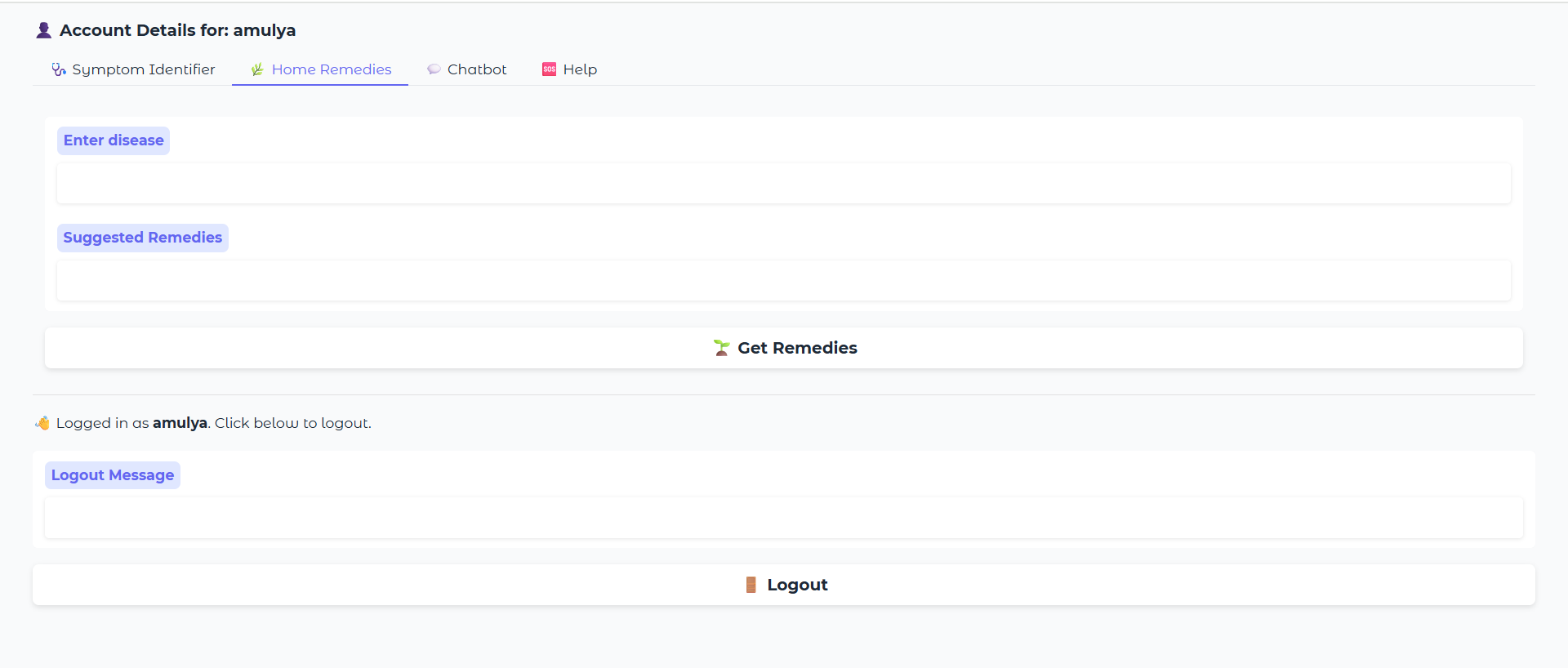
**Symptom Identifier Output:**

### 



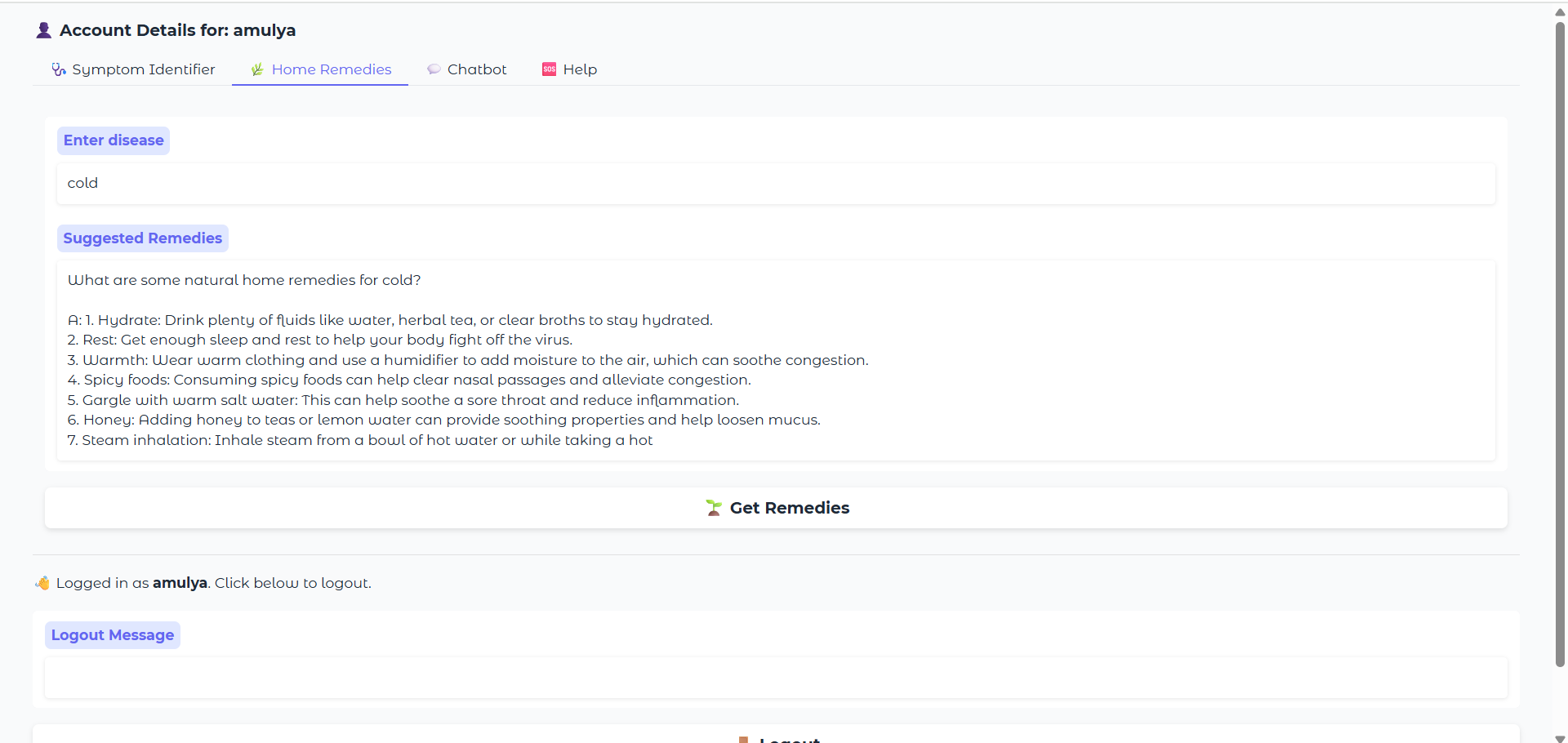
### 

### **Home Remedies Page:**

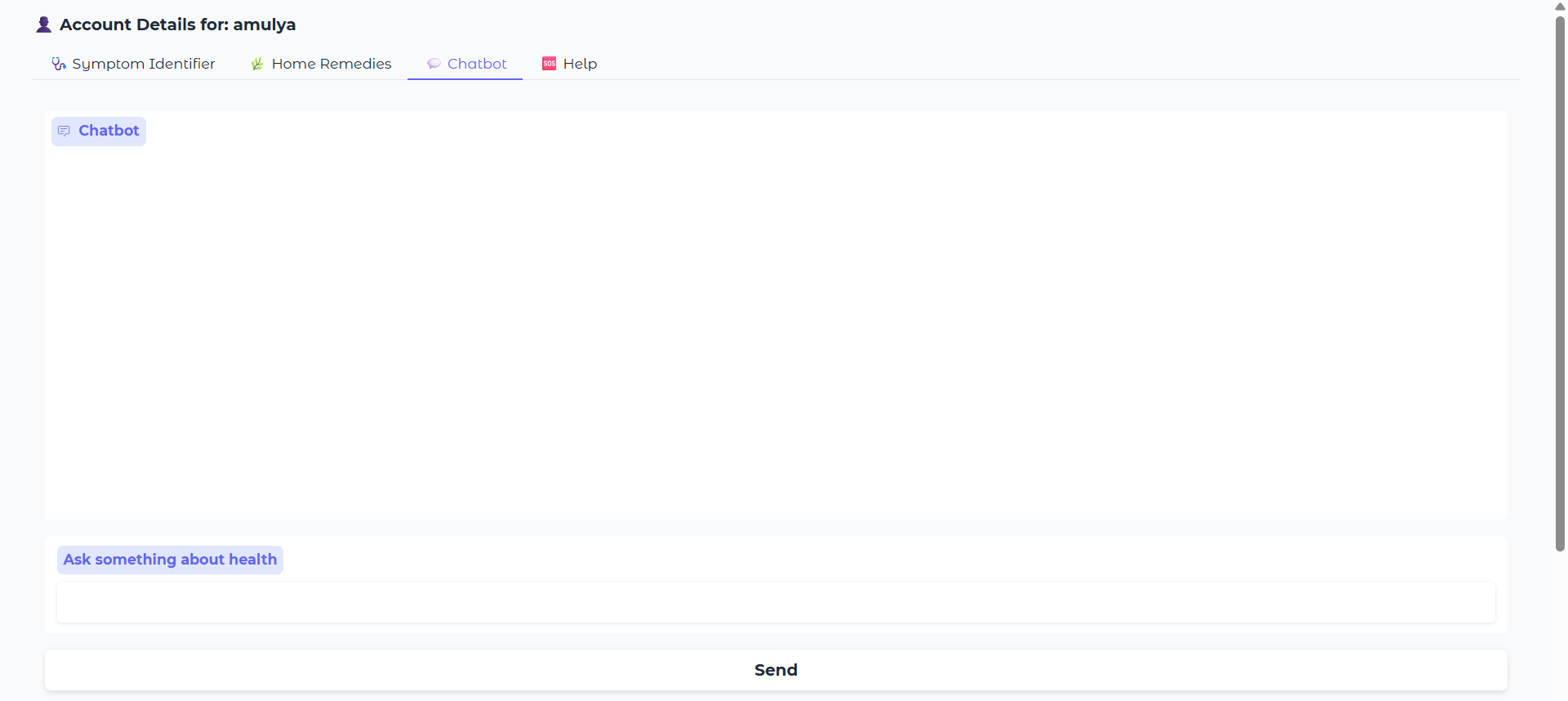


**Description:** The **Home Remedies** feature of Health AI provides natural and practical solutions for common health issues. Users can enter the name of a condition—such as *cold*, *headache*, or *indigestion*—and the AI will suggest trusted, home-based remedies. These remedies are generated using a medical language model and focus on ingredients or practices that are easily available and safe. This tool is especially useful for those looking for gentle, non-pharmaceutical relief. It encourages self-care while supporting traditional wellness practices in a modern, AI-powered way.

**Home Remedies output:**

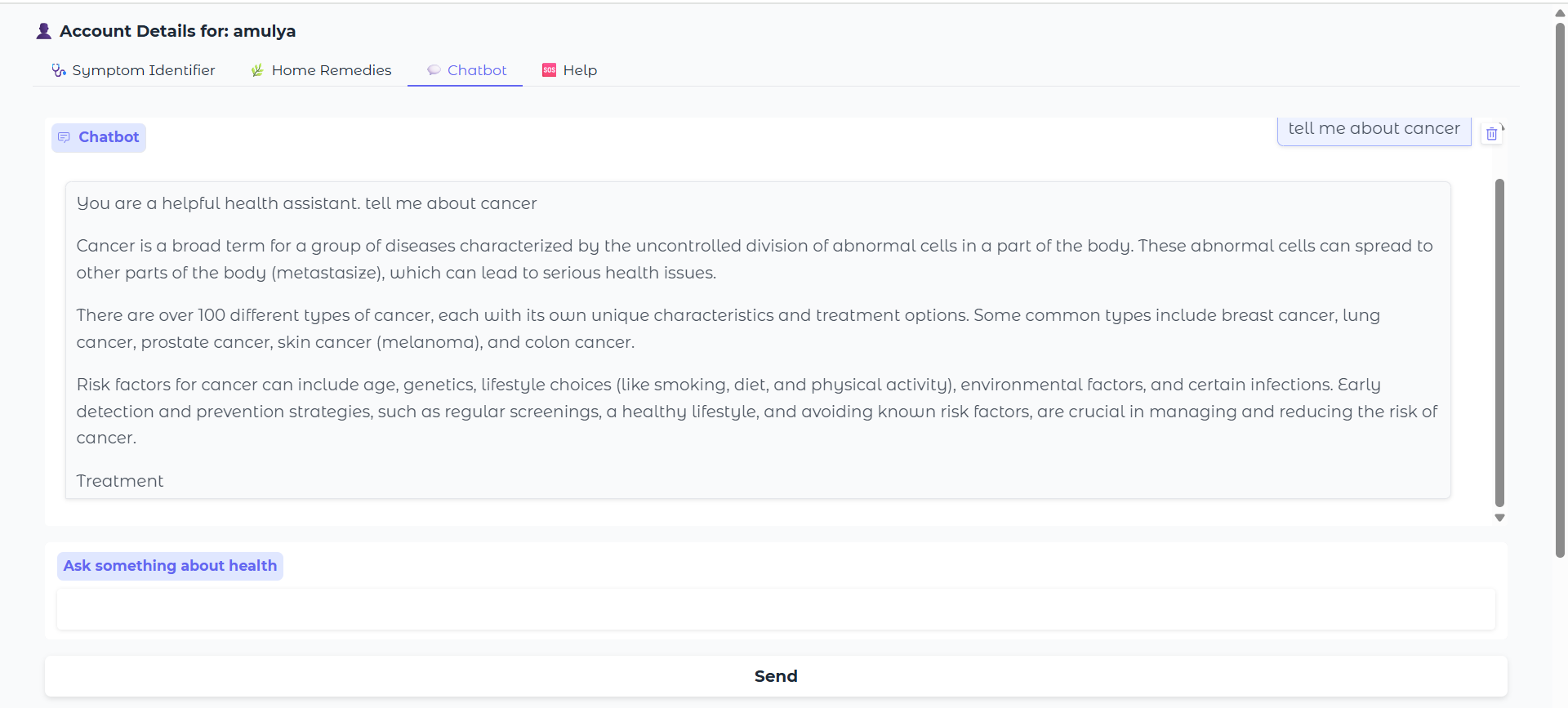
****

**Patient chat page:**

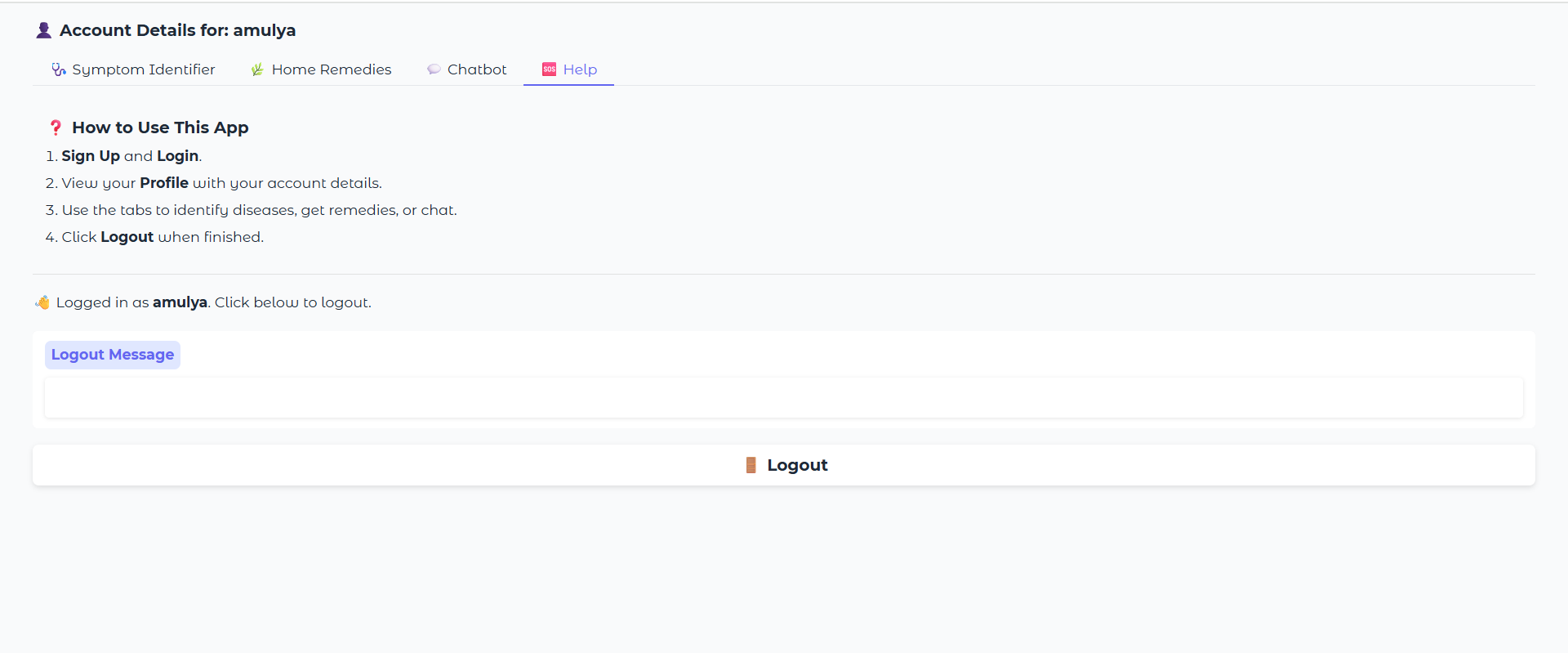
****

**Description:**The HealthBot Chat Assistant is an intelligent, conversational feature of Health AI designed to answer a wide range of health-related questions. Users can chat with the AI in natural language, asking about symptoms, remedies, healthy habits, or general medical advice. Powered by the IBM Granite AI model, the chatbot responds instantly with clear, informative answers, creating a helpful and engaging user experience. It’s like having a virtual health assistant available anytime. While it doesn’t replace professional medical consultation, it offers quick support and guidance for everyday health concerns.

**Patient chat output:**

****

**Help page:**



**Description:**The Help section of Health AI serves as a quick and easy guide for users to understand how to use the application effectively. It provides clear instructions on signing up, logging in, using the symptom identifier, exploring home remedies, and chatting with the HealthBot. This section ensures that even first-time users can navigate the platform with confidence. Designed to be simple and informative, the Help tab enhances user experience by answering common questions and explaining each feature’s purpose. It acts as a built-in support system, available whenever users need guidance.

**8. ADVANTAGES & DISADVANTAGES**

Smart SDLC offers numerous advantages that enhance the software development experience. It enables users to automate time-consuming tasks such as requirement classification, code generation, and bug fixing using simple natural language inputs. This significantly reduces development time and allows even non-programmers to prototype solutions quickly. The integrated chatbot and voice input improve usability, while the modular design allows easy extension to support new SDLC features. By using the IBM Granite model, Smart SDLC ensures high-quality AI responses and provides an educational tool for students learning software engineering concepts.

However, the system also has some limitations. It currently relies on in-memory session storage, which means user data is not saved across sessions. Since the AI outputs are based on generative models, the accuracy of code and classification can vary and should be reviewed before use in production. The platform requires a stable internet connection and cannot function offline. Additionally, integration with real-world development environments (such as IDEs or CI/CD pipelines) is not yet available but planned for future updates. Despite these challenges, Smart SDLC provides a powerful foundation for AI-driven software lifecycle automation.

**9. CONCLUSION**

The SmartSDLC platform represents a significant advancement in the automation of the Software Development Lifecycle by integrating AI-powered intelligence into each phase—from requirement analysis to code generation, testing, bug fixing, and documentation. By leveraging cutting-edge technologies like IBM Watsonx, FastAPI, LangChain, and Streamlit, the system demonstrates how generative AI can streamline traditional software engineering tasks, reduce manual errors, and accelerate development timelines.

The platform’s modular architecture and intuitive interface empower both technical and non-technical users to interact with SDLC tasks efficiently. Features such as requirement classification from PDFs, AI-generated user stories, code generation from natural language, auto test case generation, smart bug fixing, and integrated chat assistance illustrate the power of AI when applied thoughtfully within a development framework.

Overall, SmartSDLC not only improves productivity and accuracy but also sets the foundation for future enhancements like CI/CD integration, team collaboration, version control, and cloud deployment. It is a step toward building intelligent, developer-friendly ecosystems that support modern agile development needs with smart automation at its core.

**10. FUTURE SCOPE**

Smart SDLC has strong potential for future enhancements that can transform it into a fully intelligent development assistant. One key direction is the integration of persistent storage using databases like MongoDB or Firebase to allow users to save their inputs, generated code, and progress history. The platform can also be extended to support multilingual prompts, enabling non-English-speaking users to benefit from the system. Another major enhancement is the integration of CI/CD pipelines and DevOps tools (such as Jenkins or GitHub Actions), allowing the AI-generated code to be tested, versioned, and deployed automatically. Real-time code suggestions via IDE plugins (e.g., for VS Code) could make AI assistance available directly inside developers’ editors. Further improvements include training or fine-tuning custom AI models for better accuracy in requirement classification and bug detection. With these advancements, Smart SDLC could evolve into a scalable AI-powered platform suitable for students, startups, and software enterprises.

**11. APPENDIX**

**GitHub Link:** https://github.com/praveen05471/SMARTSDLC-PROJECT