Project Documentation

1. Introduction

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

• Team Leader : BAVATHARANI T

• Team Member : AFRIN A

• Team Member : ISWARYA S

• Team Member : DIVYA B

• Team Member : DHANYA SRI R

2. Project Overview

Purpose

The purpose of SmartSDLC is to transform the traditional Software Development Lifecycle (SDLC) by integrating Artificial Intelligence at every phase—requirements, design, development, testing, deployment, and maintenance.

SmartSDLC empowers developers, testers, and project managers by:
- Automating repetitive tasks such as documentation, code reviews, and testing.
- Using AI-driven insights to predict project risks, estimate timelines, and detect anomalies in code quality.
- Providing natural language interfaces for requirement gathering and policy summarization.
- Assisting with strategic planning through forecasting of project KPIs like defect density, productivity, and delivery timelines.

Ultimately, SmartSDLC bridges AI, software engineering, and project management, creating a more efficient, adaptive, and intelligent development process.

Features

• Conversational Interface
  - Key Point: AI-driven natural language interaction
  - Functionality: Enables developers, testers, and managers to query project status, request documentation, or generate reports using plain language.

• Requirement Summarization
  - Key Point: Simplified requirement understanding

- Functionality: Converts lengthy requirement documents into concise, actionable summaries.

• Effort & Timeline Forecasting
- Key Point: Predictive project analytics
- Functionality: Uses historical project data to forecast development effort, delivery time, and resource needs.

• AI Code Assistant
- Key Point: Smart development support
- Functionality: Provides code recommendations, detects anomalies, and enforces coding standards.

• Defect Prediction & Anomaly Detection
- Key Point: Early issue identification
- Functionality: Flags unusual bug patterns or performance bottlenecks in the software.

• KPI Forecasting
- Key Point: Strategic project monitoring
- Functionality: Projects KPIs such as code quality trends, bug resolution rates, and developer productivity.

• Multimodal Input Support
- Key Point: Flexible document handling
- Functionality: Accepts text, requirement PDFs, design diagrams, and CSV logs for AI-based analysis.

• User-Friendly Dashboard
- Key Point: Unified project workspace
- Functionality: Provides an intuitive interface for teams to interact with SmartSDLC, view progress, and manage tasks.

3. Architecture

- Frontend (Streamlit): Provides an interactive UI with dashboards for project KPIs, file uploads, chat assistant, task management, and reporting.
- Backend (FastAPI): Manages APIs for requirement summarization, code analysis, bug prediction, report generation, and AI-driven recommendations.
- LLM Integration (IBM Watsonx Granite): Used for natural language understanding to summarize requirements, generate documentation, assist with queries, and provide coding insights.
- Vector Search (Pinecone): Stores and retrieves project documents, requirements, and past reports using semantic search.
- ML Modules (Forecasting & Anomaly Detection): Implements forecasting (effort, cost, timeline) and detects anomalies in defect patterns or code quality using Scikit-learn.

4. Setup Instructions

Prerequisites:
- Python 3.9+
- pip & virtual environment tools
- API keys for IBM Watsonx and Pinecone
- Internet access

Installation:
1. Clone the repository
2. Install dependencies from requirements.txt
3. Create a .env file and configure credentials
4. Run backend server with FastAPI
5. Launch frontend with Streamlit
6. Upload project documents, requirements, or logs to interact with AI modules.

5. Folder Structure

app/ – FastAPI backend logic
app/api/ – Modular API routes (chat, requirement, code, bug prediction, reports)
ui/ – Streamlit frontend with dashboards and visualizations
smart_dashboard.py – Entry script for SmartSDLC dashboard
granite_llm.py – Handles LLM queries
document_embedder.py – Embeds requirements/docs in Pinecone
kpi_forecaster.py – Forecasts project KPIs
anomaly_checker.py – Detects unusual patterns
report_generator.py – Generates reports

6. Running the Application

1. Launch the FastAPI backend server.
2. Run the Streamlit dashboard.
3. Navigate via sidebar (requirements, code review, testing, KPI forecasts).
4. Upload project files (docs, CSVs, logs).
5. Interact with the AI assistant for documentation, bug prediction, or KPI analysis.
6. Download AI-generated reports.

7. API Documentation

- POST /chat/ask → Ask AI questions about project or requirements.
- POST /upload-doc → Upload requirement/design docs.
- GET /search-docs → Retrieve similar project docs.
- GET /forecast-kpi → Get AI-driven KPI forecasts.
- POST /submit-feedback → Collect team feedback.

All APIs are documented in Swagger UI.

8. Authentication

- JWT token-based authentication
- OAuth2 with enterprise credentials
- Role-based access (Admin, Developer, Tester, Manager)
- Future enhancement: session history tracking

9. User Interface

- Sidebar navigation
- KPI visualization (burndown charts, defect trends, productivity metrics)
- Tabbed layouts for requirements, code review, bug prediction, and forecasts
- AI chat assistant
- Report download capability

10. Testing

- Unit Testing: Requirement parsing & AI functions
- API Testing: Swagger UI, Postman
- Manual Testing: Document uploads, predictions
- Edge Cases: Malformed inputs, large logs, missing data

11. Screenshots

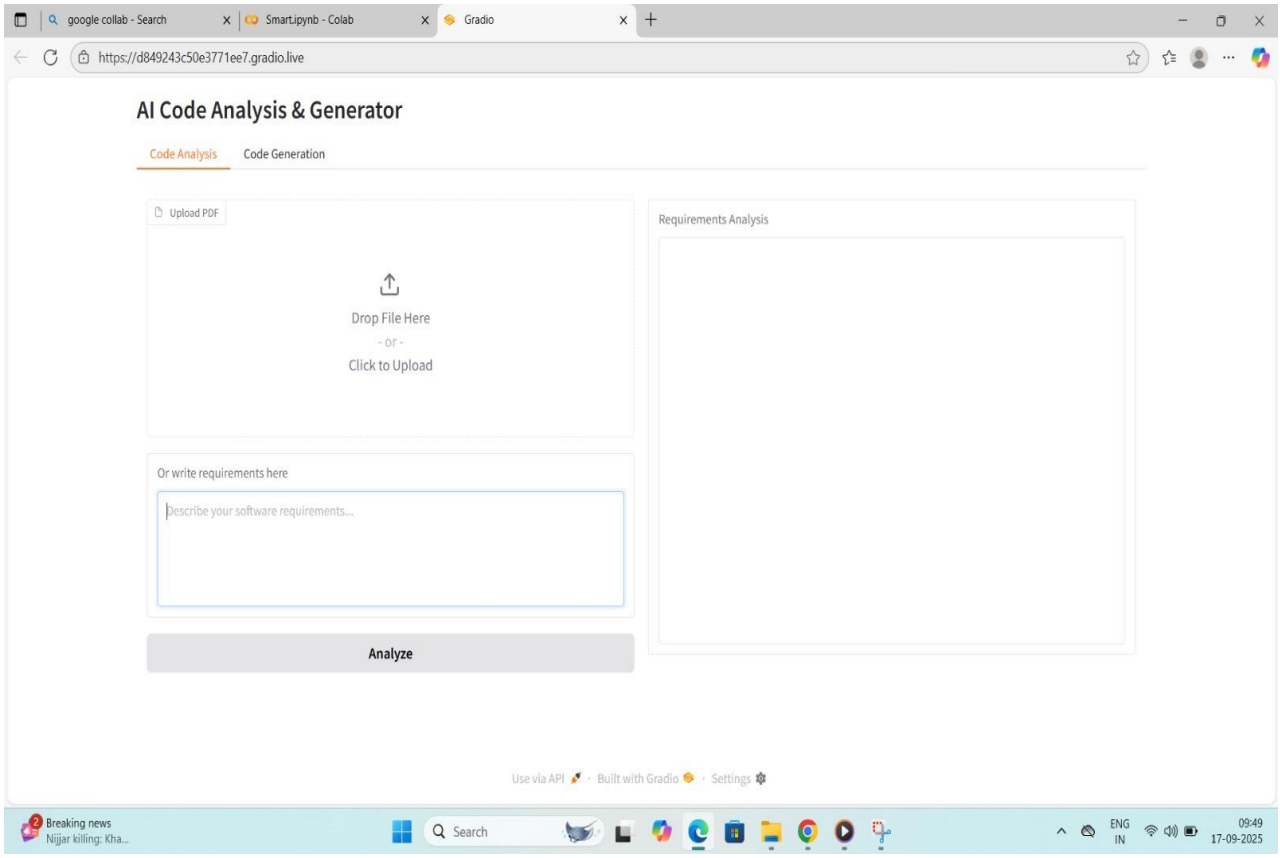
(To be added after deployment demo)

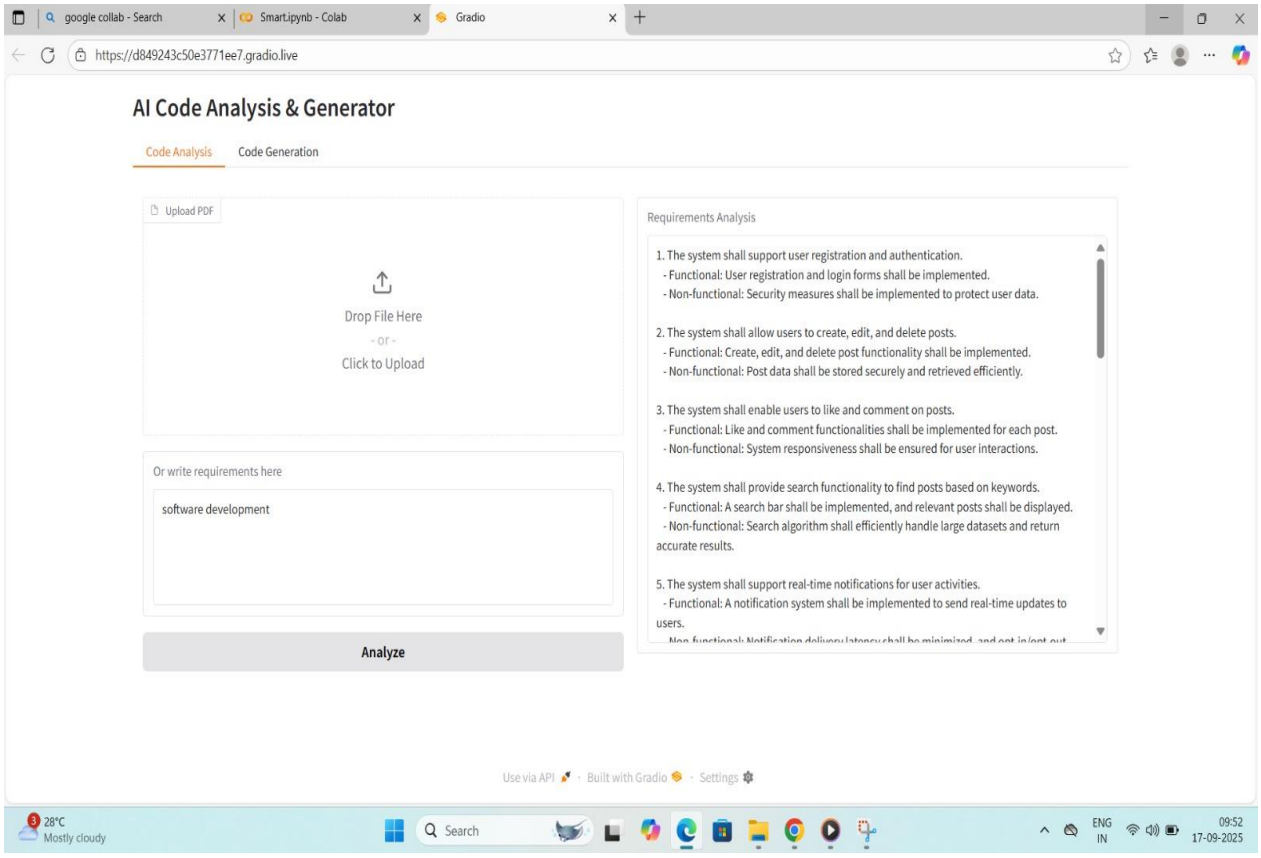
12. Known Issues

- Dependency on API keys (Watsonx, Pinecone)
- Limited offline functionality

13. Future Enhancements

- Automated test case generation
- Intelligent bug triaging
- Integration with CI/CD tools
- Support for A


google collab - Search
Smart.ipynb - Colab
Gradio
https://d849243c50e3771ee7.gradio.live
AI Code Analysis & Generator
Code Analysis
Code Generation
Upload PDF
Drop File Here
- or -
Click to Upload
Requirements Analysis
Or write requirements here
Describe your software requirements...
Analyze
Use via API
Built with Gradio
Settings
Breaking news
Nijjar killing: Kha...
Search
ENG
IN
09:49
17-09-2025



gile sprint planning