SmartSDLC – AI-Enhanced Software Development Lifecycle

Team

- **Team Leader:** M SUJAY KUMAR
- **Team Member:** ANANTHA NARAYANAN.S
- **Team Member:** M. Balachandar
- **Team Member:** JEYARUKSHAN.S

Project Documentation

1. Project Overview

The core purpose of this project, as detailed in the provided document, is to build a **Sustainable Smart City Assistant**. This AI-powered tool is designed to help cities and residents create a more eco-conscious urban environment. It achieves this by leveraging AI and real-time data to optimize key resources like energy, water, and waste. The assistant also serves as a valuable decision-making tool for city officials, offering clear insights, forecasting capabilities, and summaries of complex policies.

Features

The assistant includes a wide range of features to support its purpose:

- **Conversational Interface:** Allows for natural language interaction.
- **Policy Summarization:** Converts lengthy government documents into concise, actionable summaries.
- **Resource Forecasting:** Uses predictive analytics to estimate future resource usage.
- **Eco-Tip Generator:** Recommends personalized actions to reduce environmental impact.
- **Citizen Feedback Loop:** Collects and analyzes public input to inform city planning.
- **KPI Forecasting:** Helps officials track progress and plan ahead.
- **Anomaly Detection:** Identifies unusual patterns in data to flag potential issues early.
- **Multimodal Input Support:** Accepts text, PDFs, and CSVs for analysis.
- **User-friendly UI:** Provides an intuitive dashboard for all users.

2. Architecture

The project's architecture is composed of several integrated components:

- **Frontend (Streamlit):** An interactive web UI built with the Streamlit framework.
- **Backend (FastAPI):** A REST framework that provides API endpoints for all functionalities.
- **LLM Integration (IBM Watsonx Granite):** Utilizes Granite LLM models for natural language understanding and generation.

- **Vector Search (Pinecone):** Stores document embeddings for semantic search.
- **ML Modules:** Lightweight ML models built with Scikit-learn handle forecasting and anomaly detection.

3. Setup Instructions

To get the project running, follow these steps:

1. **Prerequisites:** Ensure you have Python 3.9+, pip, and API keys for IBM Watsonx and Pinecone.
2. **Installation:** Clone the repository, install dependencies, and create a .env file for credentials.
3. **Running:** Launch the FastAPI server for the backend and then the Streamlit dashboard for the frontend.

4. Other Details

- **Folder Structure:** The project is organized into app/ (backend), ui/ (frontend), and several key Python scripts for different functionalities.
- **API Documentation:** The backend APIs are documented in Swagger UI, with endpoints for chat, document uploads, and more.
- **User Interface:** The UI is designed to be minimalist, functional, and accessible, featuring a sidebar, KPI visualizations, and various layouts.
- **Testing:** The project underwent a multi-phase testing process, including unit, API, manual, and edge case handling.