Sustainable Smart City – Project Documentation

Project Title: Sustainable Smart City

Team ID: NM2025TMID12529
Team Size: 5
Team Leader: Faleel H
Team Members:

1. Rajendrabharathi S
2. Mohamed Ajeem
3. Wasim Akram S
4. PUGAZHARASAN A

1. Introduction

The Sustainable Smart City Assistant is an innovative AI-driven project aimed at creating greener and smarter urban environments. By combining Artificial Intelligence (AI), real-time data, and user-friendly interfaces, the system assists city officials and citizens to adopt more sustainable practices. It is designed to help manage resources such as water, energy, and waste while simplifying complex policy documents into actionable insights. Through this project, cities can become more efficient, inclusive, and resilient.

2. Project Objectives

The main objectives of this project are: - To provide personalized eco-friendly tips for citizens to adopt sustainable lifestyles. - To summarize lengthy government policies into simple and clear points. - To support city officials with predictive analytics for energy, water, and waste management. - To improve citizen engagement through real-time feedback mechanisms. - To act as a decision-making partner for better governance.

3. Problem Statement

Urbanization has created several challenges such as energy shortages, waste mismanagement, and high levels of pollution. Citizens are often unaware of how they can contribute to sustainability, while policymakers struggle to communicate important regulations effectively. Traditional systems are fragmented and fail to provide real-time insights. This project addresses these gaps by delivering a single platform that simplifies policies, generates actionable tips, and forecasts future trends.

4. Scope of the Project

The project focuses on eco-friendly city management using AI. It covers two primary functionalities: eco-tip generation and policy summarization. The assistant is designed for use by both citizens and government officials. It does not currently manage large-scale IoT networks or enforce compliance but serves as a supportive digital assistant to improve awareness and planning.

5. Literature Review / Background

Many existing smart city projects emphasize infrastructure, IoT, and automation but lack citizen-centered tools. Previous studies show that sustainable practices are more effective when citizens are engaged directly. AI-based summarization and eco guidance are still emerging fields, and this project builds upon them by combining both functionalities into a single interactive platform.

6. Features of the System

Key features include: - Conversational Interface for natural interactions. - Eco-Tip Generator for practical sustainability advice. - Policy Summarization for better policy awareness. - Citizen Feedback Loop for inclusive governance. - KPI Forecasting and anomaly detection for city planners. - Multi-format support for analyzing documents like PDFs.

7. System Requirements

To run the project successfully, the following requirements are necessary: - Google Colab or a local Python environment with GPU support. - Python 3.9 or later. - Installed libraries: Transformers, Torch, Gradio, PyPDF2. - Stable internet connection for model downloads. - Optional: T4 GPU runtime in Colab for faster execution.

8. Technology Stack

The project is built using the following technologies: - **Programming Language:** Python - **Libraries:** Hugging Face Transformers, Torch, Gradio, PyPDF2 - **Environment:** Google Colab with GPU runtime - **LLM:** IBM Granite model for natural language processing - **Interface:** Gradio-based web app for user interaction.

9. Project Architecture

The architecture consists of: - Frontend: Gradio UI with tabs for Eco Tips and Policy Summarization. - Backend: AI model integration using Hugging Face Transformers and Torch. - PDF Processor: PyPDF2 library for extracting text. - LLM Integration: IBM Granite for generating intelligent responses.

10. Data Flow Diagram

Data flows through the system as follows: 1. User inputs keywords or uploads a policy PDF. 2. Input is processed by the tokenizer and AI model. 3. Model generates eco tips or policy summaries. 4. Output is displayed on the Gradio interface in real-time.

11. Module Description

The system is divided into modules: - Eco Tips Generator: Provides actionable tips based on keywords. - Policy Summarizer: Extracts and simplifies PDF or text policies. - PDF Reader: Extracts raw text from PDF documents. - AI Response Generator: Uses LLM to produce intelligent outputs.

12. Implementation Steps

Implementation is done in Google Colab: 1. Open a new Colab notebook. 2. Change runtime to T4 GPU. 3. Install dependencies. 4. Write and run the complete project code. 5. Launch Gradio and access the app via shareable link.

13. User Interface

The user interface is built with Gradio. It provides two main tabs: - Eco Tips Generator: Accepts keywords like 'plastic' or 'solar'. - Policy Summarization: Accepts PDF uploads or text input. Outputs are displayed in text boxes with clear formatting.

14. Code Description

Key functions: - generate_response(): Generates intelligent text using the model. - extract_text_from_pdf(): Extracts text from PDF documents. - eco_tips_generator(): Produces sustainability advice. - policy_summarization(): Summarizes lengthy documents.

15. Testing and Validation

Testing was done in multiple stages: - Unit testing for text extraction and response generation. - Functional testing of Gradio interface. - Validation with sample policies and environmental keywords. - Edge case handling for empty or invalid inputs.

16. Output Screenshots

The Gradio app provides clear outputs: - Eco tips displayed in a large text box. - Policy summaries shown in a scrollable text area. Screenshots can be added to demonstrate the outputs visually.

17. Results and Discussions

The project successfully delivered an AI-based assistant capable of generating eco tips and simplifying policies. Users found the outputs practical and easy to understand. The system proved efficient in handling large documents and producing concise summaries.

18. Advantages of the Project

Advantages include: - Encourages sustainable living among citizens. - Saves time by simplifying lengthy policies. - Provides decision-making support to city officials. - Easy-to-use interface accessible via web.

19. Limitations of the Project

Limitations include: - Dependence on GPU for faster processing. - Limited to English language in the current version. - Accuracy depends on the LLM model quality. - No offline functionality.

20. Future Enhancements

Planned improvements: - Multilingual support for wider accessibility. - Mobile app version for broader reach. - Integration with IoT data from smart sensors. - Stronger security with user authentication.

Conclusion

The Sustainable Smart City Assistant demonstrates how AI can transform urban management into a more eco-conscious, efficient, and citizen-friendly process. By bridging the gap between citizens, policymakers, and technology, it creates a foundation for sustainable living. With further enhancements, this project can evolve into a critical tool for smart governance and environmental responsibility.