**Sustainable Smart City Assistant Using IBM Granite LLM**

Project Documentation

# 1. Introduction

**Project Title: Sustainable Smart City Assistant Using IBM granite LLM**

• Team Member: VETRISELVAN S

• Team Member: ELUMALAI K

• Team Member: EXAROWAN M

• Team Member: KEERTHIKA R

• Team Member: MADHUMITHA S

# 2. Project Overview

• Purpose:

The purpose of the Eco Assistant & Policy Analyzer is to empower individuals, communities, and policymakers with tools for sustainable living and informed decision-making. The assistant generates eco-friendly tips based on user-provided keywords and simplifies complex policy documents into concise summaries. It combines AI-driven natural language processing with a user-friendly interface to promote environmental awareness and actionable insights.

• Features:

* Eco Tips Generator: Provides personalized eco-friendly recommendations.
* Policy Summarization: Simplifies lengthy documents into key provisions and implications.
* PDF Support: Extracts text directly from uploaded policy documents.
* Interactive UI: Intuitive interface built with Gradio.
* AI Integration: Uses IBM Granite LLM via Hugging Face Transformers.

# 3. Architecture

Frontend (Gradio):

Provides a tabbed interface with input fields, buttons, and output textboxes for both eco tips and policy summaries.

Backend (PyTorch + Transformers):

Handles text processing, PDF parsing, and interaction with the IBM Granite model.

LLM Integration (IBM Granite):

Granite-3.2-2b-instruct model from Hugging Face provides language understanding and generation.

# 4. Setup Instructions

**Prerequisites:**

* Python 3.9 or later
* pip and virtual environment tools
* Internet access to download Hugging Face models

**Installation Process:**

* Install dependencies: pip install transformers torch gradio PyPDF2
* Run the script: python maja.txt
* The Gradio app will launch locally with a shareable link.

# 5. Folder Structure

maja.txt – Main application file containing model loading, functions, and Gradio interface.

functions/ – (Optional future extension) Separate utility functions.

uploads/ – (Optional) Store uploaded PDF files.

# 6. Running the Application

* Run python code
* Open the link in your browser provided by Gradio.
* Navigate to Eco Tips Generator tab for sustainability suggestions.
* Navigate to Policy Summarization tab to analyze PDFs or pasted text.

# 7. User Flows

Eco Tips Generator: Input keywords → AI generates actionable tips.

Policy Summarization: Upload PDF or paste text → AI produces key points.

# 8. Authentication

The current version runs in an open environment for demonstration. Future enhancements may include user authentication and access control.

# 9. User Interface

* Tabbed layout with two main sections (Eco Tips Generator, Policy Summarization).
* File upload support for policy analysis.
* Large output textboxes for readability.
* Shareable Gradio link for easy collaboration.

# 10. Testing

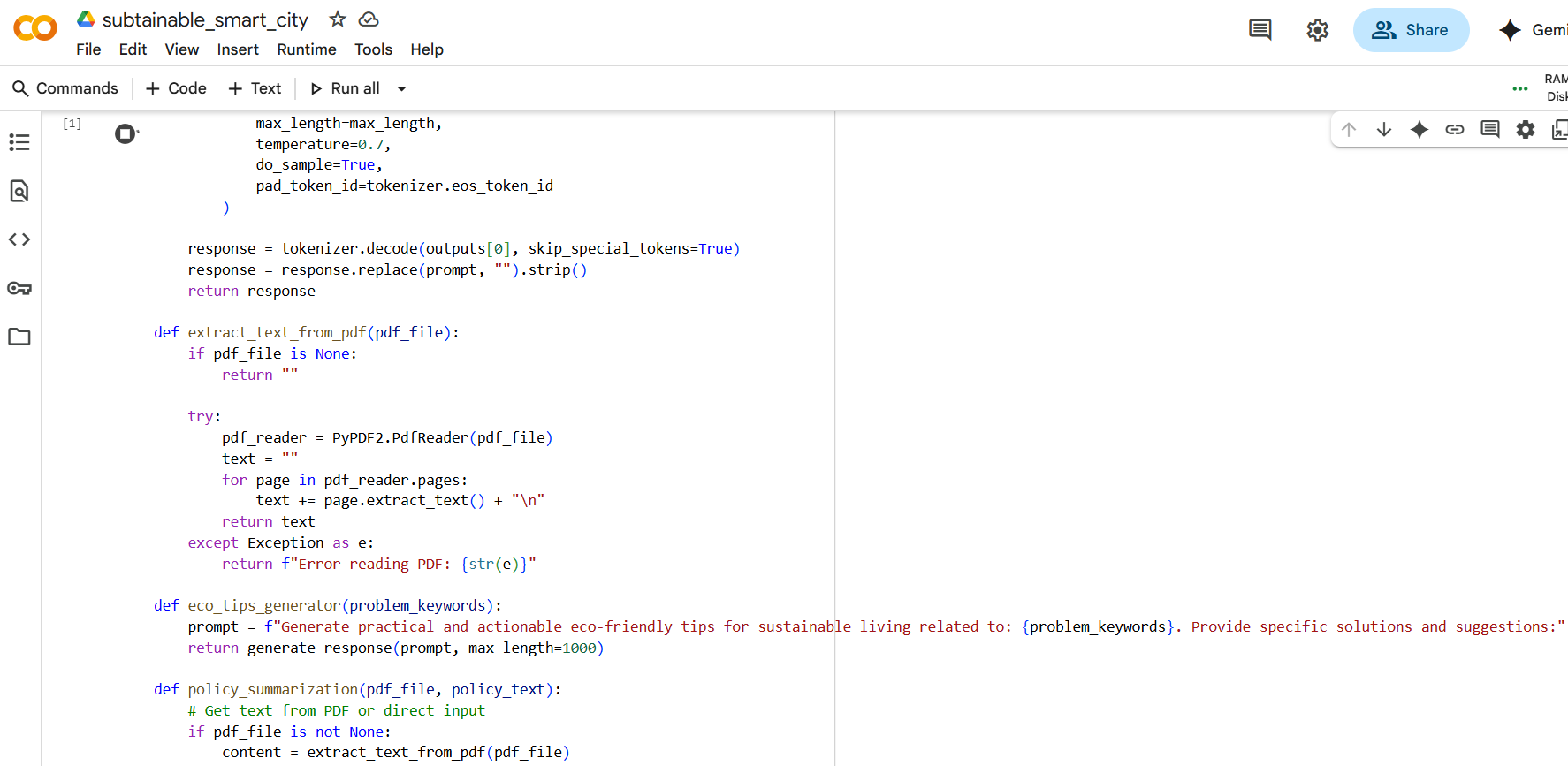
Unit Testing: Functions like eco\_tips\_generator and policy\_summarization were tested individually.

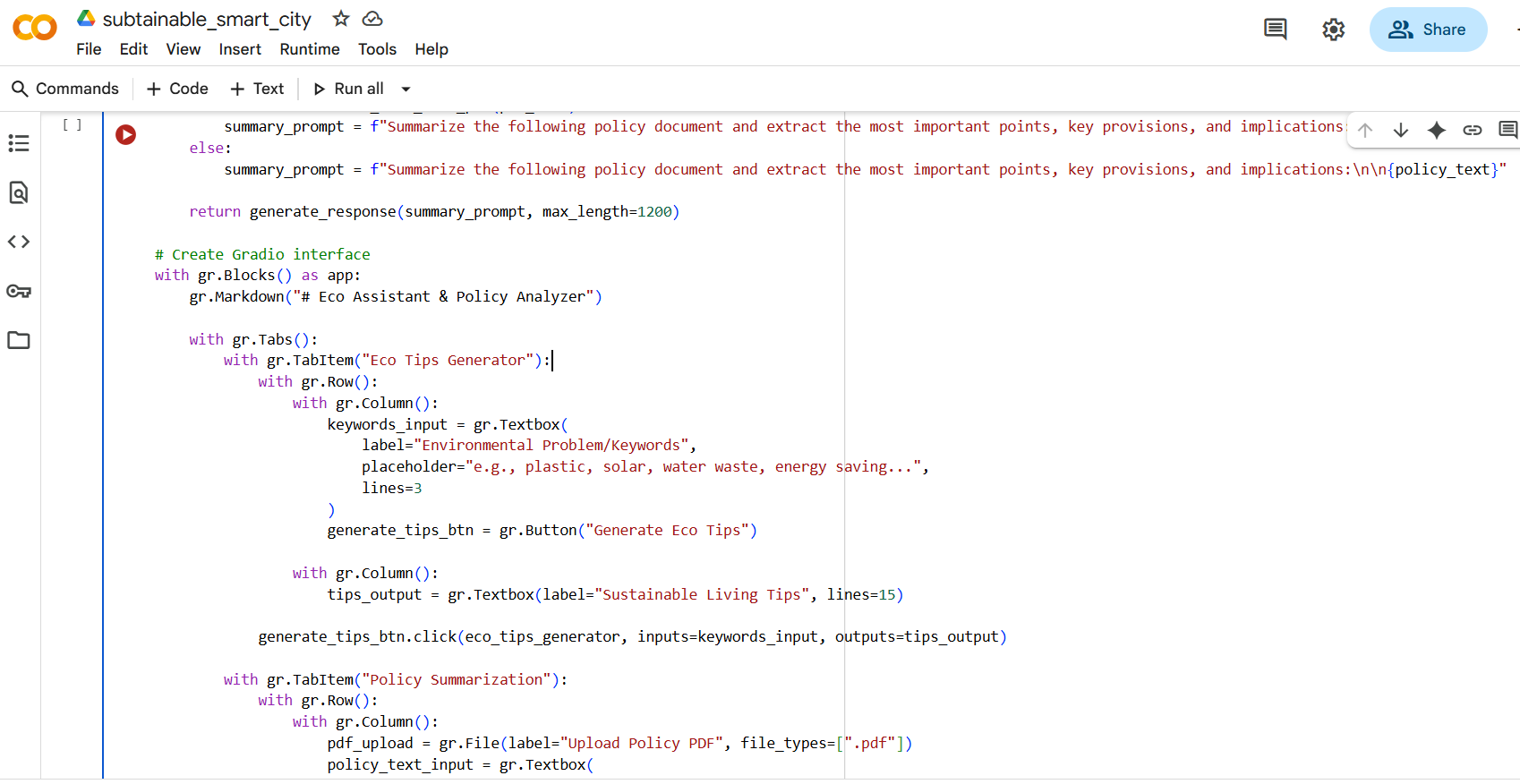
Manual Testing: Gradio interface tested with sample inputs and PDFs.

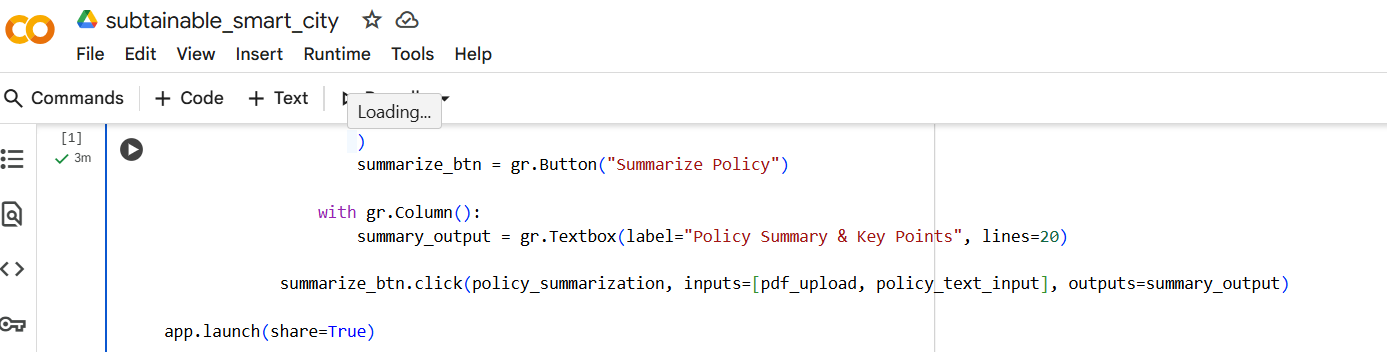
Edge Case Handling: Empty input, corrupted PDFs handled with error messages.

# 11. Screenshots











# 12. Known Issues

• Large PDFs may slow down summarization.

• Requires stable internet connection for model usage.

• Limited to English text at present.

# 13. Future Enhancements

• Add multi-language support.

• Provide visual insights such as charts or graphs.

• Detailed policy recommendations beyond summarization.

• Extend API endpoints for integration with other applications.

# 14. Git Hub Link:

**Code link:**

https://github.com/Exarowan/sustainable\_smart\_city.git