Project Documentation

## Smart City Project - Virtual Assistant & Analyzer

# 1. Introduction

Project Title : Smart City - Virtual Assistant & Analyzer

Team Member : ASWIN B

Team Member : MUKESH C

Team Member : NIJAY ADHITYA P K

Team Member : MADHURAI VEERAN S

# 2. Project Overview

Purpose :  
  
The purpose of this Smart City project is to provide a digital assistant that empowers users with eco-friendly tips, analyzes policy documents, answers policy-related queries, and generates actionable plans for sustainable living. By leveraging IBM Granite LLM and Gradio interface, it demonstrates how AI can support both citizens and city officials in promoting smart and sustainable decisions.

Features:  
  
• Conversational Interface – Natural language interaction to ask questions and get updates.  
• Policy Summarization – Summarizes lengthy policy documents into concise points.  
• Eco-Tip Generator – Provides personalized sustainability advice.  
• Policy Q&A – Answers questions based on uploaded policy documents.  
• Action Plan Generator – Creates detailed step-by-step action plans for sustainable living.  
• PDF Text Extraction – Reads and extracts text from uploaded PDF documents.  
• Gradio UI – User-friendly interface with multiple tabs for easy interaction.

# 3. Architecture

Frontend (Gradio): Provides an interactive web UI with tabs for Eco Tips, Policy Summarization, Policy Q&A, and Action Plan generation.  
  
Backend (Python & IBM Granite LLM): Powers natural language understanding and response generation.  
  
Libraries Used: Transformers, Torch, Gradio, PyPDF2.

# 4. Setup Instructions

Prerequisites:  
• Python 3.9 or later  
• pip and virtual environment tools  
• Internet access to install dependencies and run the app  
  
Installation Process:  
• Install dependencies using pip  
• Load the IBM Granite model and tokenizer  
• Run the Gradio application

# 5. Folder Structure

SmartCity.ipynb – Jupyter notebook containing full implementation.  
  
Functions included: generate\_response, extract\_text\_from\_pdf, eco\_tips\_generator, policy\_summarization, policy\_qa, generate\_action\_plan.  
  
Gradio interface – Includes tabs for Eco Tips, Policy Summarization, Policy Q&A, and Action Plan Generator.

# 6. Running the Application

• Install the required libraries.  
• Load the model and tokenizer.  
• Run the Gradio Blocks interface.  
• Use Eco Tips Generator to get sustainability tips.  
• Use Policy Summarization to upload PDFs or input text for summarization.  
• Use Policy Q&A to ask questions about uploaded policy documents.  
• Use Action Plan Generator to create step-by-step plans for sustainable living.

# 7. API Documentation

Functions:  
  
• generate\_response(prompt) – Returns AI-generated text response.

• extract\_text\_from\_pdf(pdf\_file) – Extracts text content from uploaded PDF.

• eco\_tips\_generator(problem\_keywords) – Provides eco-friendly tips based on keywords.

• policy\_summarization(pdf\_file, policy\_text) – Summarizes uploaded or pasted policy documents.

• policy\_qa(pdf\_file, question) – Answers specific questions based on uploaded policy document.

• generate\_action\_plan(problem\_keywords) – Generates a detailed action plan with steps, stakeholders, and challenges.

# 8. Authentication

This demo version runs in an open environment without authentication. For secure deployment, token-based authentication or OAuth can be integrated.

# 9. User Interface

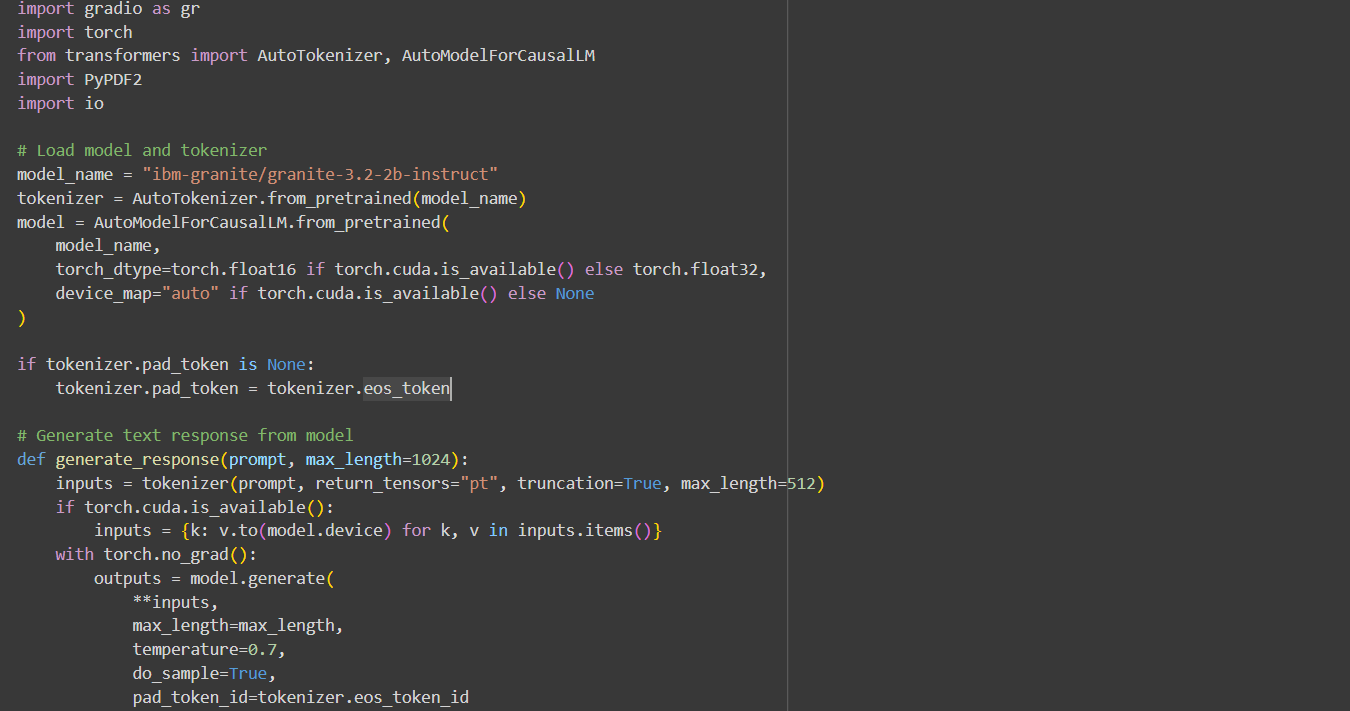
The interface is built using Gradio with:  
• Sidebar navigation  
• Tabbed layouts for Eco Tips, Policy Summarization, Policy Q&A, and Action Plan Generator  
• Real-time interaction with AI model  
• Output display for tips, policy summaries, Q&A answers, and action plans

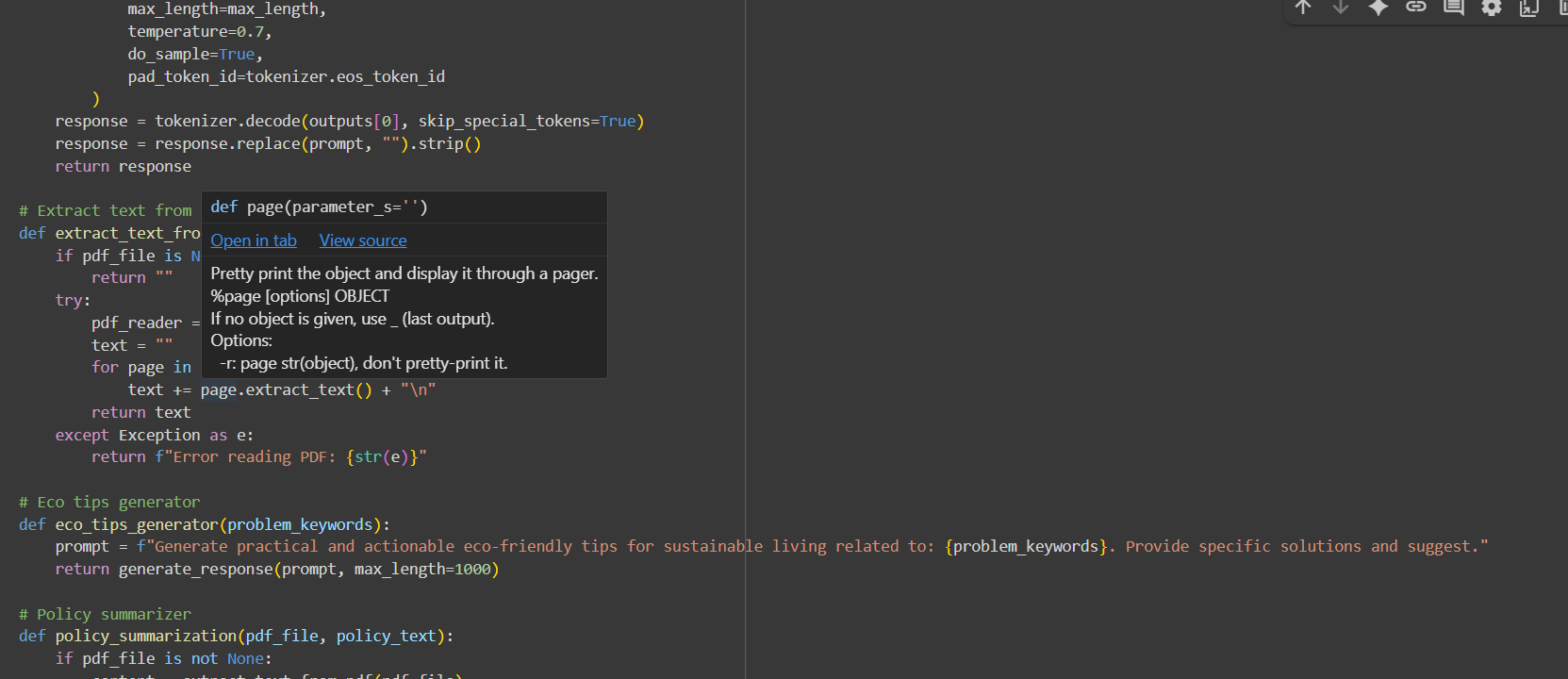
# 10. Testing

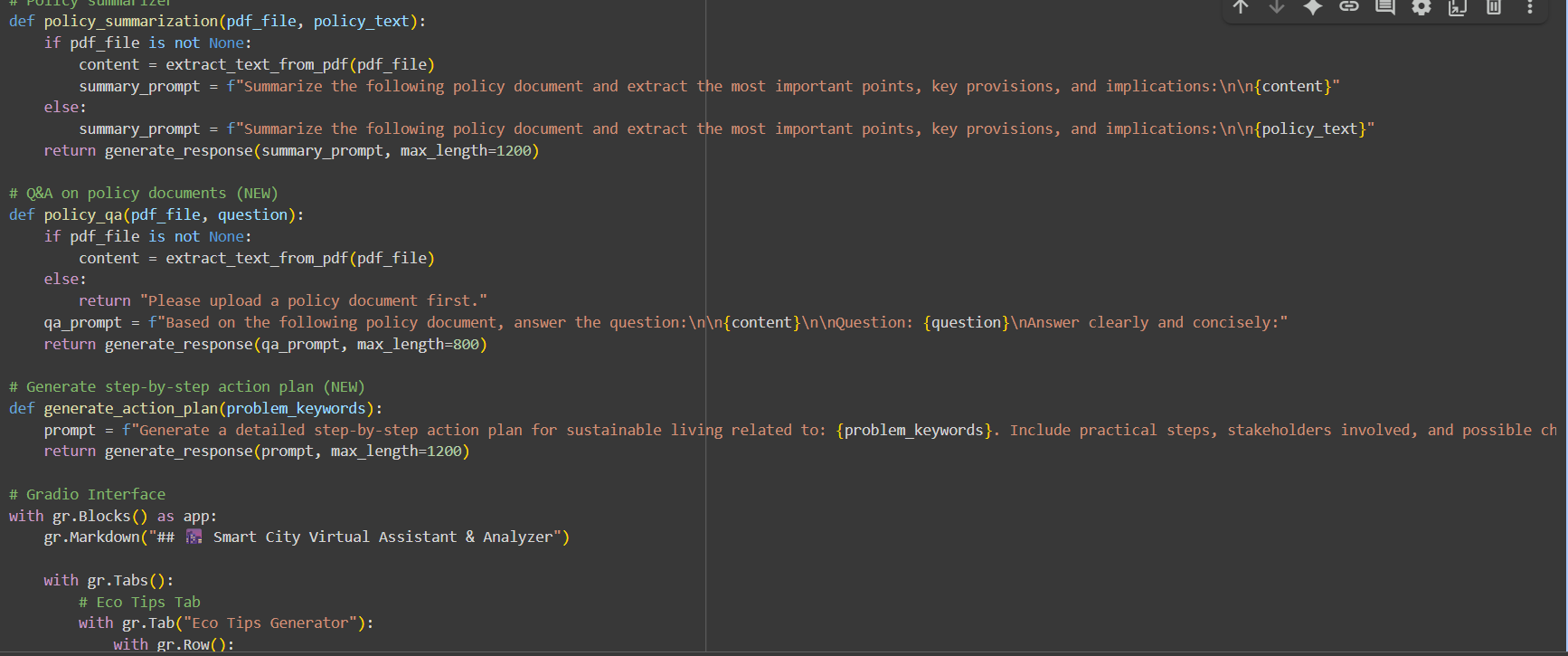
Testing Phases:  
  
• Unit Testing – For model functions and PDF extraction.  
• Manual Testing – For Gradio UI, eco tips generation, policy summarization, policy Q&A, and action plan generation.  
• Edge Case Handling – For empty inputs, invalid PDFs, and large text.

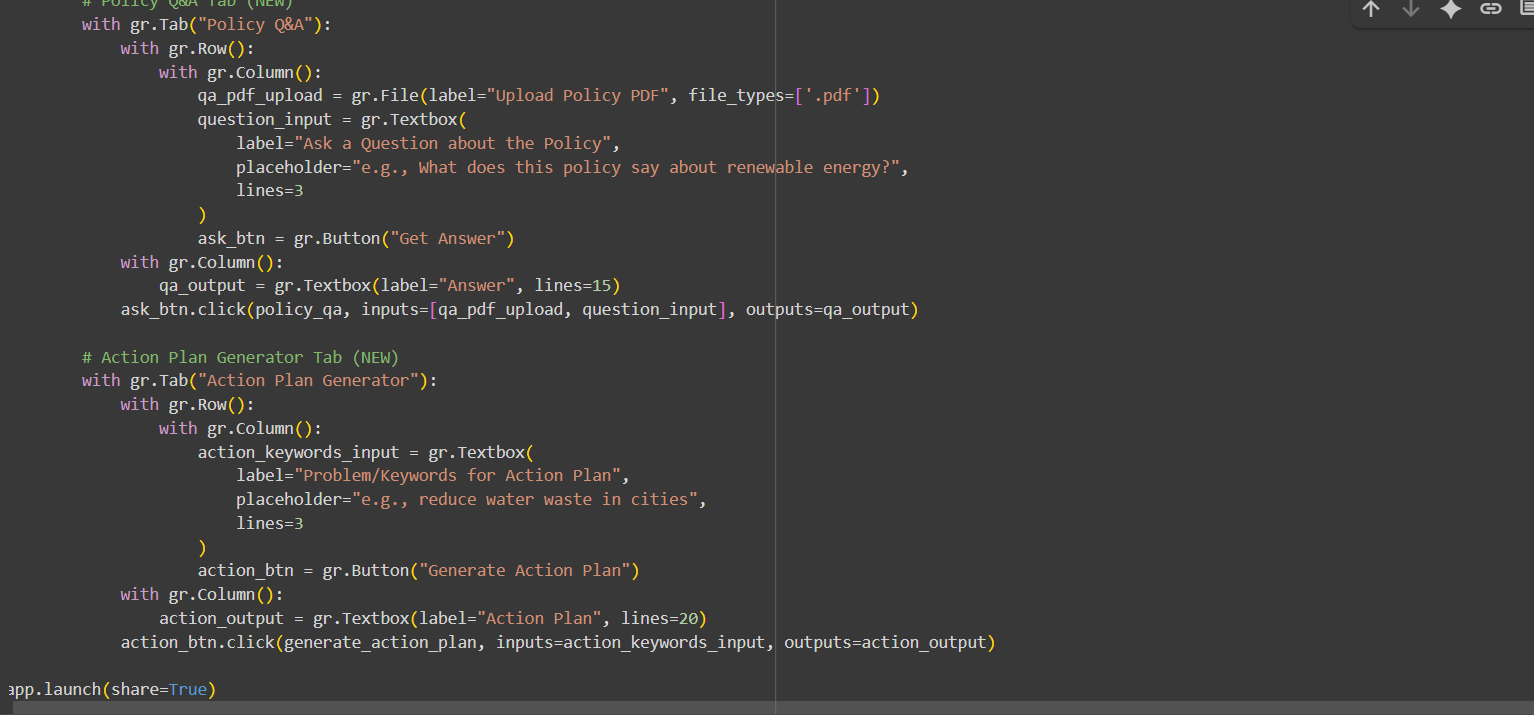
# 11. Screenshots

Screenshots of the Gradio interface should be added here.

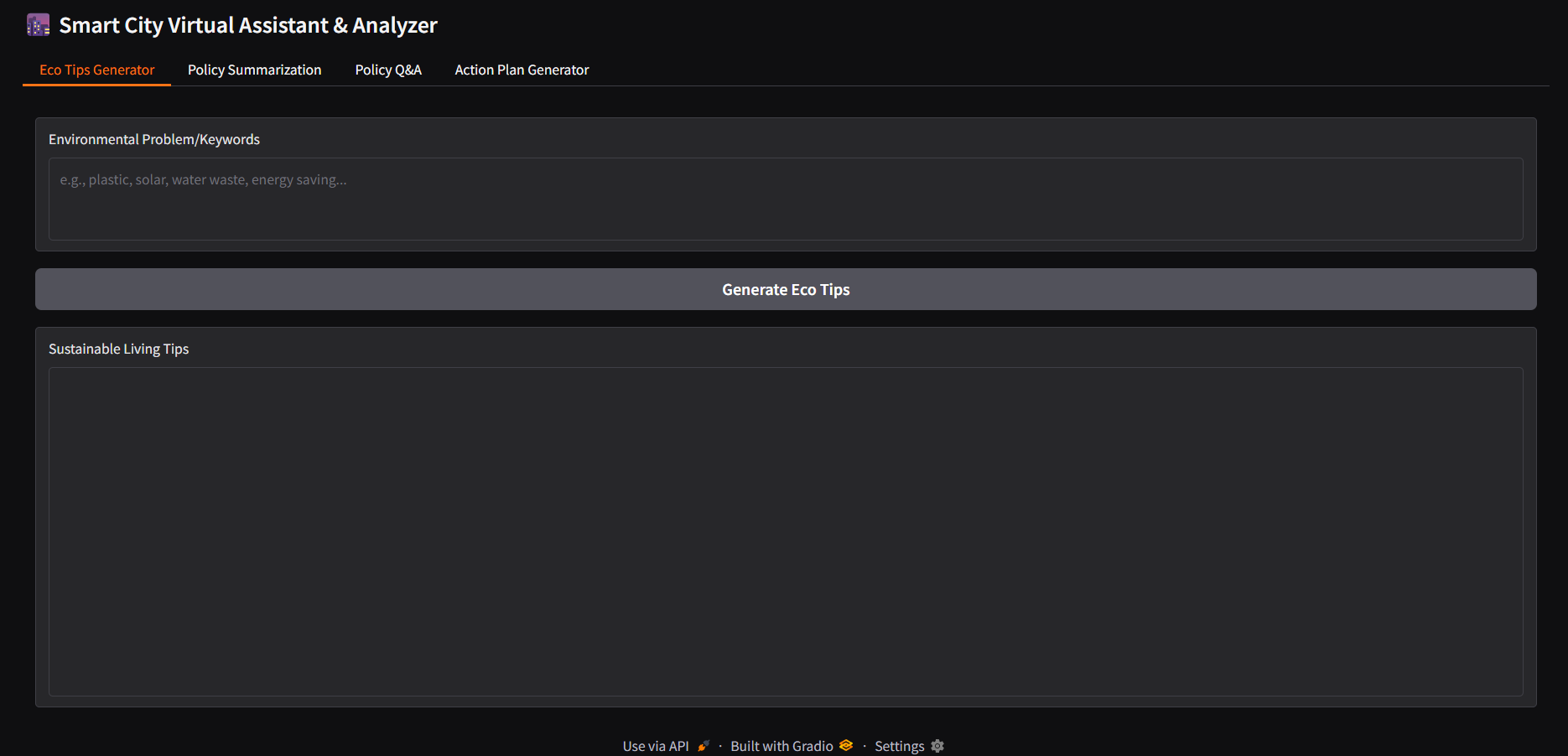


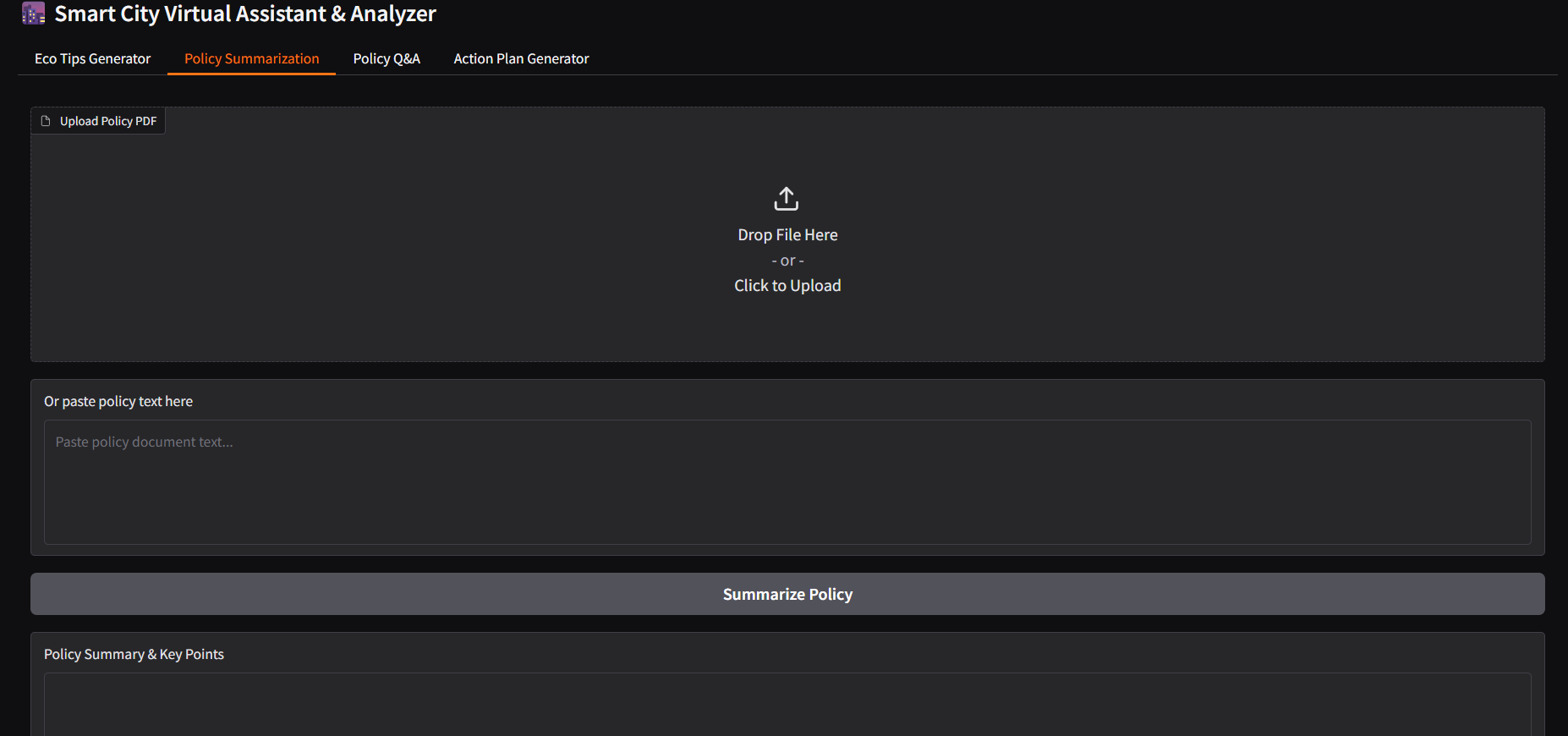


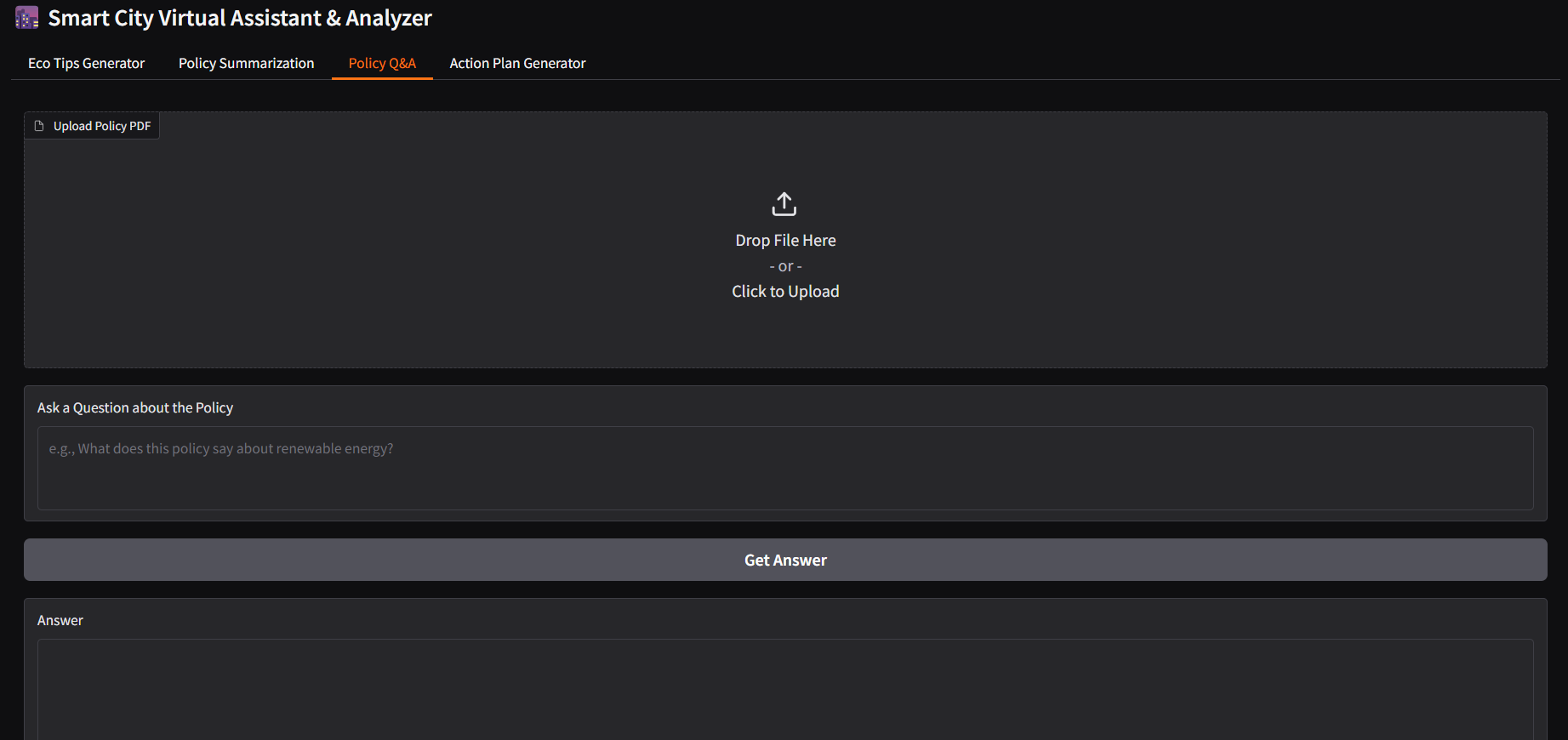


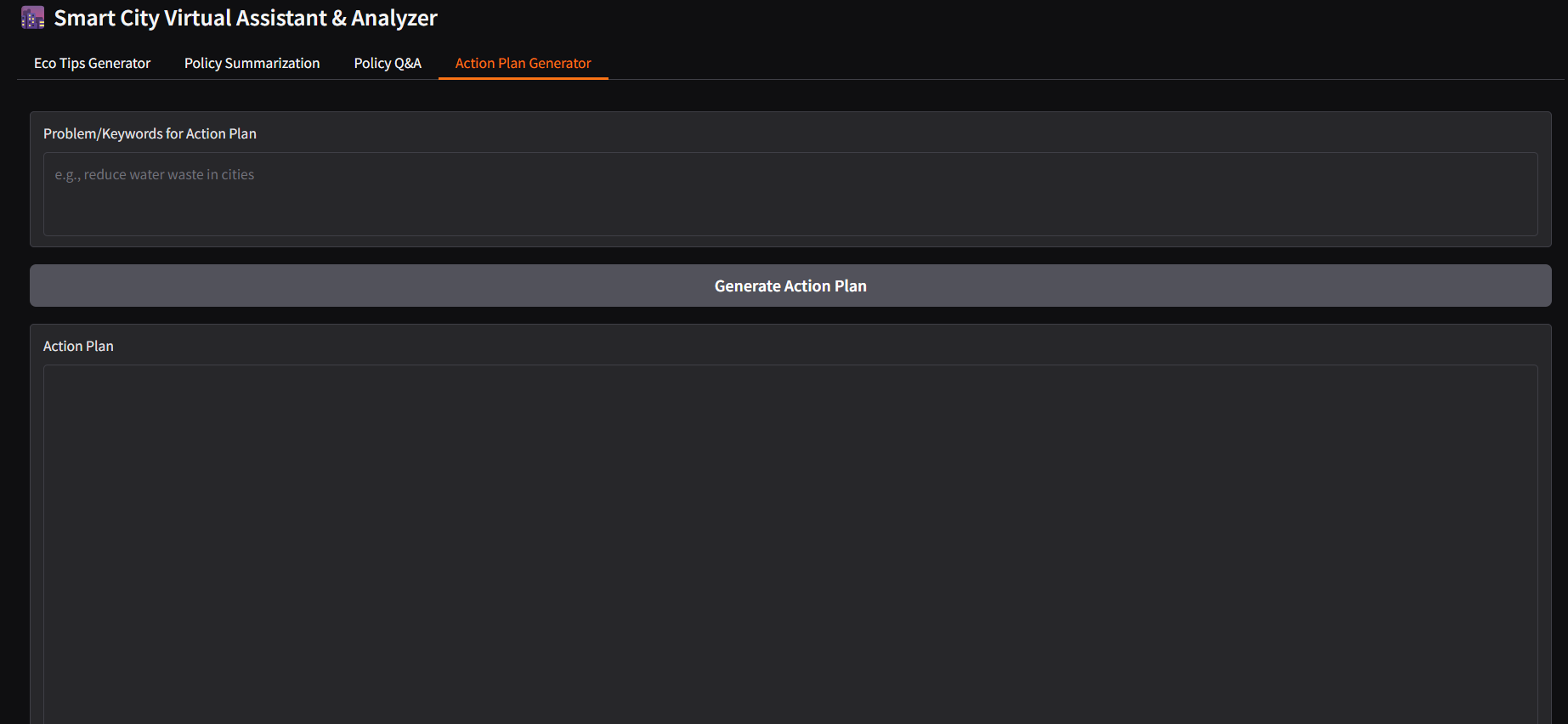


# Output:









# 12. Known Issues

• Large PDF documents may take time to process.  
• Summarization and Q&A depend on the model’s performance.  
• Limited customization of Gradio UI.

# 13. Future Enhancements

• Add secure authentication with user roles.  
• Extend functionality with forecasting and anomaly detection.  
• Improve UI with additional visualization features.  
• Enhance summarization and Q&A accuracy with fine-tuned models.  
• Add multilingual support for wider accessibility.