Sustainable smart city AI using IBM Granite

Project Documentaion

1.Introduction

Team Leader: KAVITHA V

Team Member: YASHVINI M

Team Member: DEVI PRIYA N

Team Member: JOICE CLARA T

Team Member: UTTARA C

The Sustainable Smart City Assistant is an AI-powered application designed to promote eco-friendly living, support policy summarization, enable citizen engagement, and provide data-driven insights for urban management. By leveraging IBM Watsonx Granite LLM, integrated machine learning modules, and a user-friendly interface, it helps solve sustainability challenges and greener cities.

2.Project overview

The Sustainable Smart City AI project brings together language models and user friendly web interface to support eco-awareness and citizen engagement.Its current capabilities include:

Conversational Interface

The system uses the IBM Granite language model with a conversational function (generate_response). Although not exposed directly in the UI as a chatbot, this engine powers eco-tip generation and policy summarization, making interactions natural and human-like.

Policy Summarization

Users can upload a policy PDF or paste raw text. The system extracts and summarizes content, highlighting key provisions, points, and implications for easy understanding of complex documents.

Eco-Tip Generator

A dedicated feature that generates specific, actionable eco-friendly living

suggestions based on user-inputted keywords (e.g., "plastic," "water waste," or "energy saving").

About us

Provides information about the apps mission to promote eco-friendly living and policy awareness through AI-based tools.

Contact us

Offers contact details including email and website for user support,inquiries,engagement with the project team.

User Authentication(Login & Sign Up)

Supports user login and signup via modal dialogs,enabling personalized and secure access to the application features

Multimodal Input Support

Supports text and PDF inputs (and in future, sensor data and images) for analysis.

Streamlit or Gradio UI

The system uses *Gradio's Blocks API* to build a modular and modern web interface. It includes tabs for eco-tip generation, policy summarization, about information, and contact details. Additionally, login and signup modals enhance user management (though authentication is placeholder).

3.Architecture

Frontend (Streamlit/Gradio): Provides conversational interface, PDF upload, and eco-tips generation.

User input components (text boxes, file uploaders, buttons).

CSS-customized login & signup modals.

Backend (FastAPI): Handles authentication and authorization.Manages API endpoints for conversation, summarization, feedback collection, anomaly detection, and forecasting.Communicates with ML modules and vector search services.

LLM Integration (IBM Watsonx Granite)

Uses Granite models for: Text generation, Summarization**,** Eco-friendly suggestions**,** Conversational intent handling**,**

Vector Search (Pinecone)provides semantic search for sustainability policies and documents.

Stores embeddings for efficient retrieval and context-aware responses.

ML Modules (Forecasting and Anomaly Detection)Forecasting module predicts resource usage.Anomaly detection module identifies irregular consumption or eco-patterns.

Models integrate with backend and serve data-driven insights.

4. Setup Instructions

Prerequisites

- Python 3.9+
- CUDA-enabled GPU (for faster inference, optional)
- API keys for:
  - IBM Watsonx Granite
  - Pinecone (if using vector search)

Installation process

It can be installed using:

```
# Install dependencies

pip install -r requirements.txt
```

5.Folder Structure

The project mainly focuses on maintainability and scalability.

app.py serves as the primary entry point for the Gradio prototype.

The **backend/** folder contains FastAPI routes and core business logic.

The **models/** directory is dedicated to AI-related components, including model loading, prompt engineering, and forecasting algorithms.

The **utils/** folder provides helper utilities, such as PDF extraction, error handling, and response post-processing scripts.

The **data/** folder stores sample and test policy documents.

The **docs/** folder contains formal documentation, architecture diagrams, and project screenshots.

The **tests/** folder ensures code reliability through unit tests and integration tests.

Supporting files like **requirements.txt, README.md,** and configuration scripts simplify environment setup and make onboarding seamless for new developers.

6.Running the Application

Frontend (Gradio):

Run the main script (app.py) directly:

python app.py should be used.It access the UI through the local URL provided or via the public share link.

7.API Documentation

Currently, no public API is available as the application runs as a standalone Gradio interface. Future versions will introduce FastAPI for REST endpoints including:

/generate_tips — Generate eco-friendly tips based on keywords.

/summarize_policy — Summarize text or PDF policy documents.

/user_auth — Login and signup services.

8. Authentication

Basic modal-based login and signup UI buttons exist but backend authentication logic is to be implemented.

9. User Interface

Tabbed UI with functionalities:

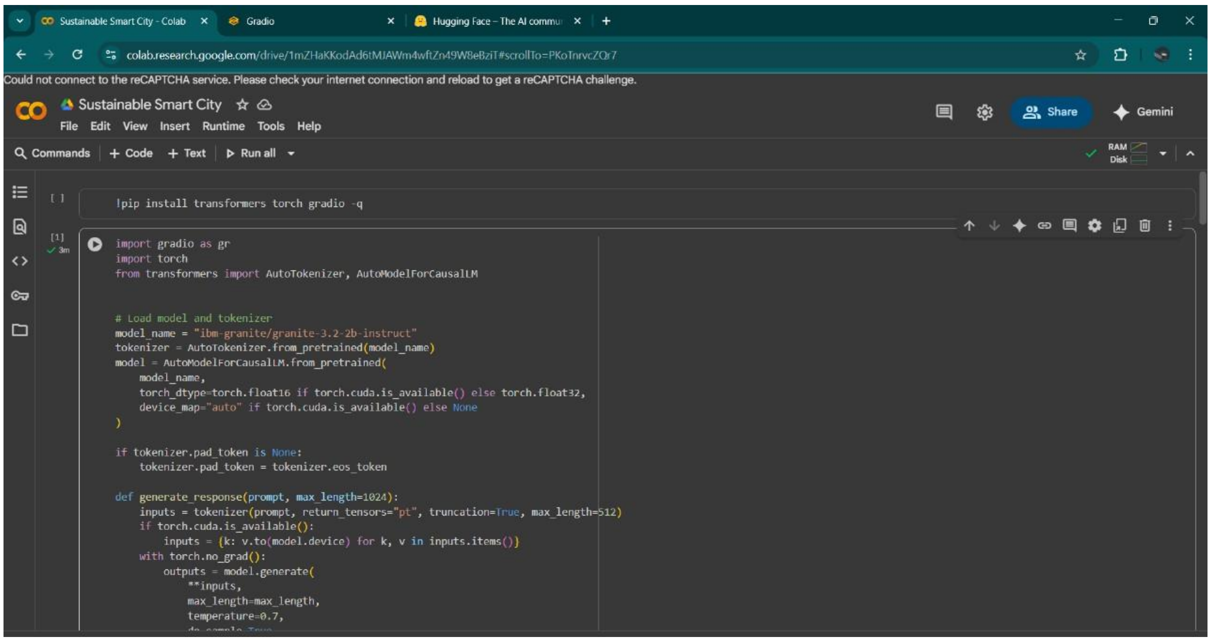
- Eco Tips Generator
- Policy Summarization
- About Us
- Contact Us
- Modal dialogs for user login and signup.
- PDF upload and text input for policy summaries.

10.Testing

Manual testing includes input of various keywords and document uploads to verify correct summary and tip generation. Automated test scripts are under development. Performance testing will involve simulating hundreds of concurrent users to evaluate the systems stability, responsiveness, and resource utilization under load. Security testing will verify that authentication and authorization mechanisms function as intended once they are implemented, ensuring proper access control and data protection.

11.Screenshots

1.Running the code in Google Colab


Sustainable Smart City - Colab
Gradio
Hugging Face – The AI commu
colab.research.google.com/drive/1mZHaKKodAd6tMJAWm4wftZn49W8eBziT#scrollTo=PKoTnrvcZQr7
Sustainable Smart City
File Edit View Insert Runtime Tools Help
Share
Gemini
Commands + Code + Text Run all
RAM
Disk
[1]
3m

Could not connect to the reCAPTCHA service. Please check your internet connection and reload to get a reCAPTCHA challenge.

```
!pip install transformers torch gradio -q
```

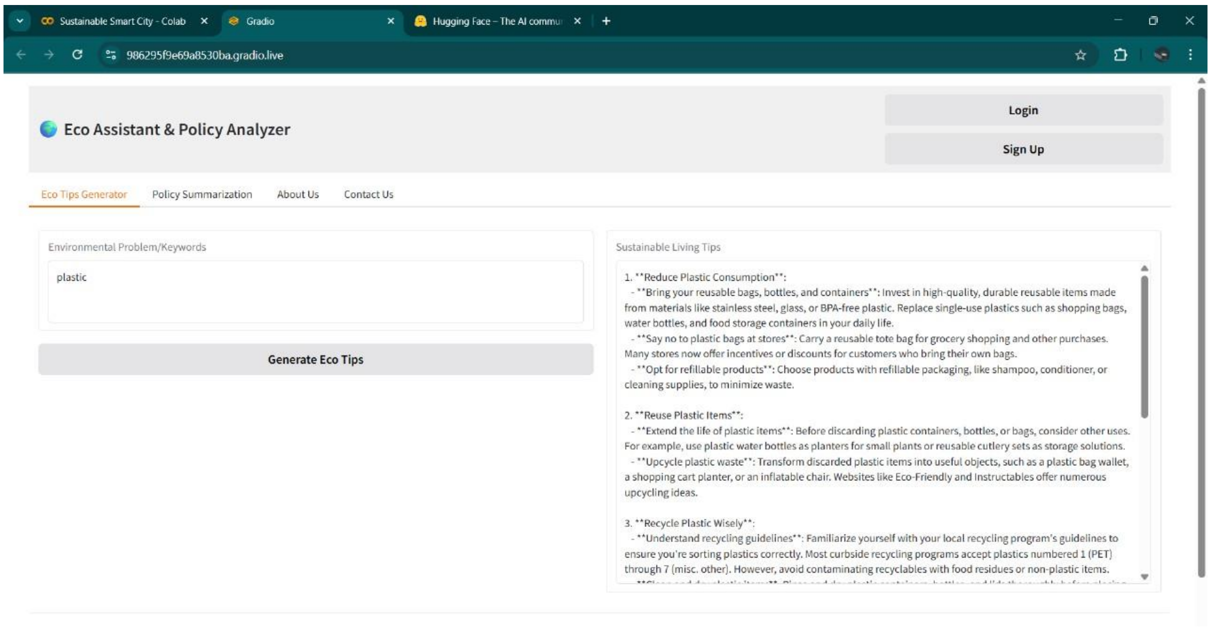
```
import gradio as gr
import torch
from transformers import AutoTokenizer, AutoModelForCausalLM


# Load model and tokenizer
model_name = "ibm-granite/granite-3.2-2b-instruct"
tokenizer = AutoTokenizer.from_pretrained(model_name)
model = AutoModelForCausalLM.from_pretrained(
    model_name,
    torch_dtype=torch.float16 if torch.cuda.is_available() else torch.float32,
    device_map="auto" if torch.cuda.is_available() else None
)

if tokenizer.pad_token is None:
    tokenizer.pad_token = tokenizer.eos_token

def generate_response(prompt, max_length=1024):
    inputs = tokenizer(prompt, return_tensors="pt", truncation=True, max_length=512)
    if torch.cuda.is_available():
        inputs = {k: v.to(model.device) for k, v in inputs.items()}
    with torch.no_grad():
        outputs = model.generate(
            **inputs,
            max_length=max_length,
            temperature=0.7,
```

2.Output showing AI Eco tips


Sustainable Smart City - Colab
Gradio
Hugging Face – The AI commu
986295f9e69a8530ba.gradio.live
Eco Assistant & Policy Analyzer
Login
Sign Up
Eco Tips Generator
Policy Summarization
About Us
Contact Us
Environmental Problem/Keywords
Generate Eco Tips
Sustainable Living Tips

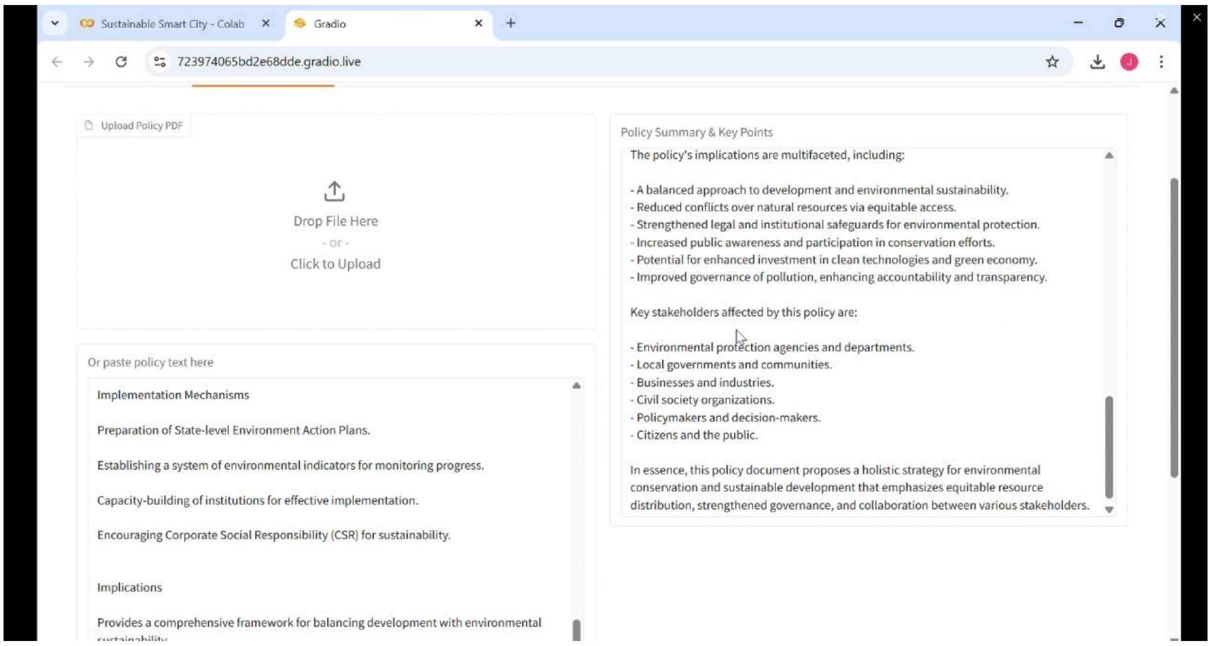
```
plastic
```

```
1. **Reduce Plastic Consumption**:
   - **Bring your reusable bags, bottles, and containers**: Invest in high-quality, durable reusable items made
from materials like stainless steel, glass, or BPA-free plastic. Replace single-use plastics such as shopping bags,
water bottles, and food storage containers in your daily life.
   - **Say no to plastic bags at stores**: Carry a reusable tote bag for grocery shopping and other purchases.
Many stores now offer incentives or discounts for customers who bring their own bags.
   - **Opt for refillable products**: Choose products with refillable packaging, like shampoo, conditioner, or
cleaning supplies, to minimize waste.

2. **Reuse Plastic Items**:
   - **Extend the life of plastic items**: Before discarding plastic containers, bottles, or bags, consider other uses.
For example, use plastic water bottles as planters for small plants or reusable cutlery sets as storage solutions.
   - **Upcycle plastic waste**: Transform discarded plastic items into useful objects, such as a plastic bag wallet,
a shopping cart planter, or an inflatable chair. Websites like Eco-Friendly and Instructables offer numerous
upcycling ideas.

3. **Recycle Plastic Wisely**:
   - **Understand recycling guidelines**: Familiarize yourself with your local recycling program's guidelines to
ensure you're sorting plastics correctly. Most curbside recycling programs accept plastics numbered 1 (PET)
through 7 (misc. other). However, avoid contaminating recyclables with food residues or non-plastic items.
```

3.Output for policy summarization


Sustainable Smart City - Colab
Gradio
723974065bd2e68dde.gradio.live
Upload Policy PDF
Drop File Here
- or -
Click to Upload
Or paste policy text here
Policy Summary & Key Points

```
Implementation Mechanisms

Preparation of State-level Environment Action Plans.

Establishing a system of environmental indicators for monitoring progress.

Capacity-building of institutions for effective implementation.

Encouraging Corporate Social Responsibility (CSR) for sustainability.


Implications

Provides a comprehensive framework for balancing development with environmental
```

```
The policy's implications are multifaceted, including:

- A balanced approach to development and environmental sustainability.
- Reduced conflicts over natural resources via equitable access.
- Strengthened legal and institutional safeguards for environmental protection.
- Increased public awareness and participation in conservation efforts.
- Potential for enhanced investment in clean technologies and green economy.
- Improved governance of pollution, enhancing accountability and transparency.

Key stakeholders affected by this policy are:

- Environmental protection agencies and departments.
- Local governments and communities.
- Businesses and industries.
- Civil society organizations.
- Policymakers and decision-makers.
- Citizens and the public.

In essence, this policy document proposes a holistic strategy for environmental
conservation and sustainable development that emphasizes equitable resource
distribution, strengthened governance, and collaboration between various stakeholders.
```

12. Known Issues

- Large PDFs may lead to incomplete text extraction or GPU memory errors.

- Authentication backend not implemented yet.
- Some complex policy texts might produce incomplete or generic summaries.

13. Future Enhancement

- Implement backend REST API with FastAPI.
- Integrate vector-based document search via Pinecone.
- Add resource forecasting, KPI forecasting, anomaly detection ML modules.
- Extend multimodal input support beyond PDF and text (e.g., images).
- Complete authentication and user management system.
- Add citizen feedback collection and reporting interface.
- Improve UI with streamlit alternative for wider deployment options.