Sustainable Smart City Assistant using IBM Granite

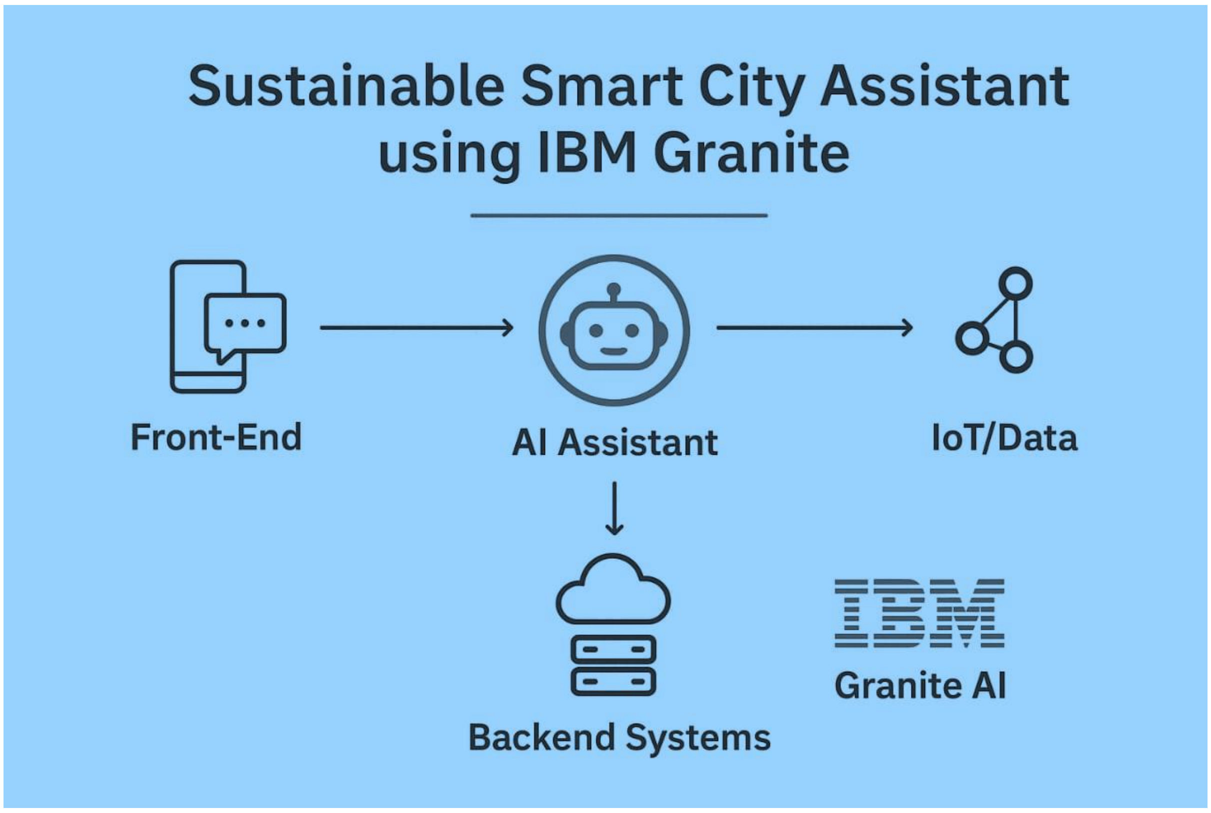
TEAM MEMBERS:

Krishnameera.M

Madhuvathani.I

Niranjana.T

Pooja.P



1. Introduction

Sustainable smart cities aim to combine technology, data, and intelligent systems to improve the quality of life, reduce environmental impact, and ensure efficient resource management. A Smart City Assistant powered by IBM Granite (Large Language Model) can act as a conversational AI that helps citizens, administrators, and planners make informed, sustainable choices.

2. Objectives

- To integrate AI-driven assistance for sustainable urban living.
- To leverage IBM Granite for natural language understanding and interaction.
- To promote eco-friendly practices and efficient city resource management.
- To provide real-time support for citizens and city authorities.

3. Role of IBM Granite

- ➢ IBM Granite is a family of foundation models optimized for enterprise use cases. Its strengths:
- ➢ Scalable AI: Handles large-scale city data and conversations.
- ➢ Natural Language Interaction: Citizens can interact in multiple languages.
- ➢ Customizable & Secure: Tailored for urban governance with enterprise-grade security.

- Sustainability Insights: Supports predictive analysis for energy, transport, and waste.

4. Features of the Smart City Assistant

- Energy Management: Suggests energy-saving measures, monitors renewable energy usage.
- Sustainable Transport: Guides citizens to use public transport, EV charging stations, and bike-sharing.
- Waste Management: Provides schedules for recycling, waste segregation tips, and alerts.
- Water Conservation: Educates on water-saving practices, monitors leakages.
- Citizen Engagement: Responds to queries about city services, emergency alerts, and green initiatives.
- Data-Driven Decisions: Assists city planners with AI powered recommendations for sustainable growth.

5. Architecture

- Front-End: Chatbot interface (mobile app, web, voice assistant).
- Core AI Engine: IBM Granite model handling natural language processing and reasoning.
- Data Integration: Real-time IoT sensors (traffic, energy, pollution, weather).
- Backend Systems: Cloud infrastructure with IBM watsonx, databases, and APIs for smart city services.

6. Benefits

- Reduces carbon footprint by promoting eco-friendly practices.
- Enhances citizen satisfaction with personalized assistance.
- Improves efficiency in energy, transport, and wasten management.
- Enables data-driven policies for city administration.
- Builds a resilient and sustainable urban ecosystem.

7. Use Case Scenarios

A citizen asks: “Where is the nearest EV charging station?” → Assistant provides real-time directions.

A planner asks: “How can we reduce peak energy consumption?” → Assistant suggests demand-response strategies.

A resident asks: “What are today’s recycling pickup schedules?” → Assistant responds instantly.

8. Future Scope

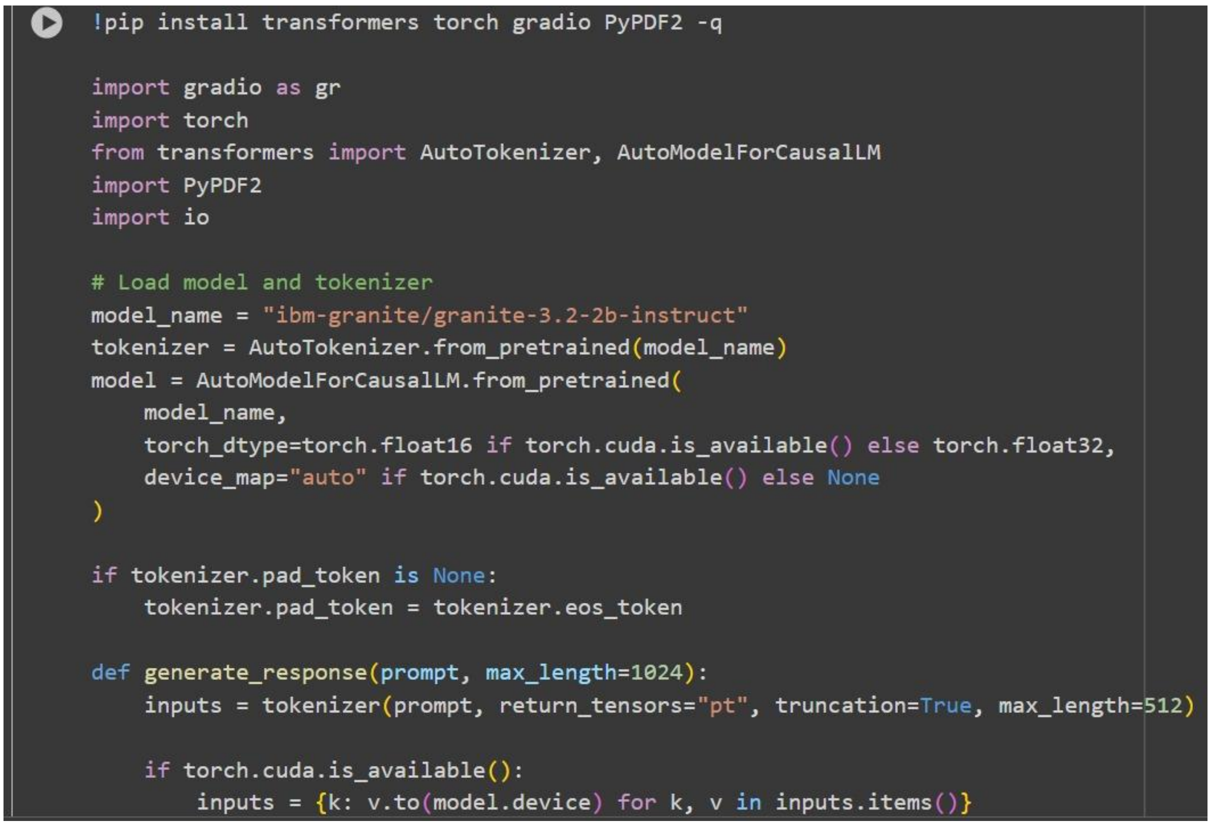
- ➢ Integration with digital twins for city planning.
- ➢ Multilingual support for inclusivity.
- ➢ AI-driven sustainability reports for city councils.
- ➢ Expansion to climate change adaptation strategies
- ➢ Information and communication technology (ICT)

Information and communication technology includes an array of data-related technologies.The capture, storage, retrieval, processing, display, representation, presentation, organization, management, security, transfer and interchange of data and information.

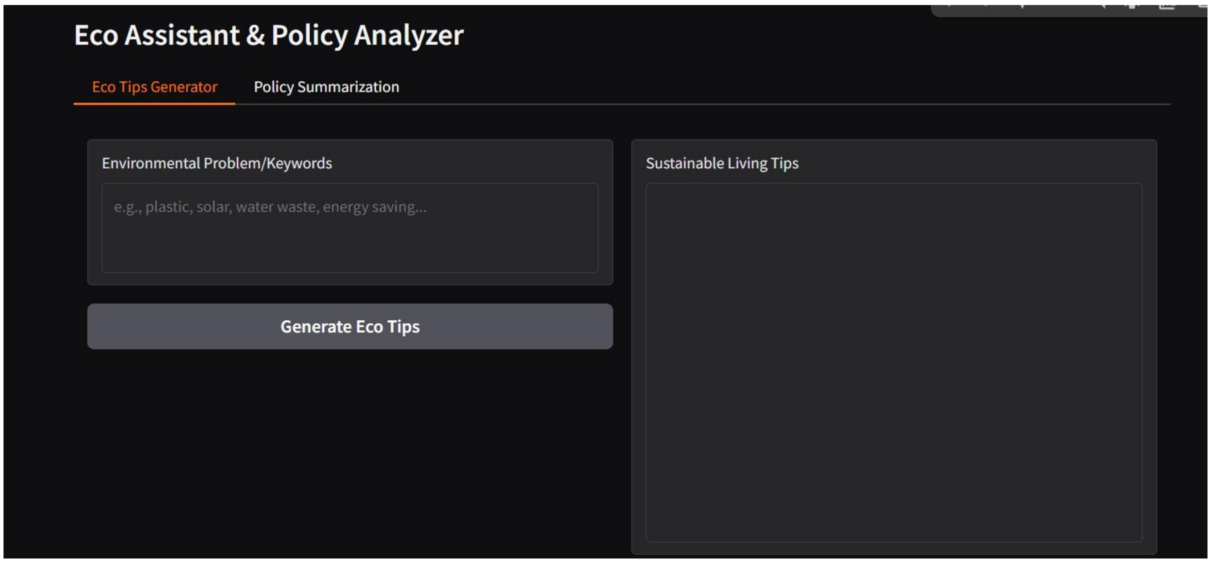
- ➢ Internet of Things (IoT)

The Internet of Things (IoT) refers to a network of physical devices, vehicles, appliances and other physical objects that are embedded with sensors, software and network connectivity .

9.Project images

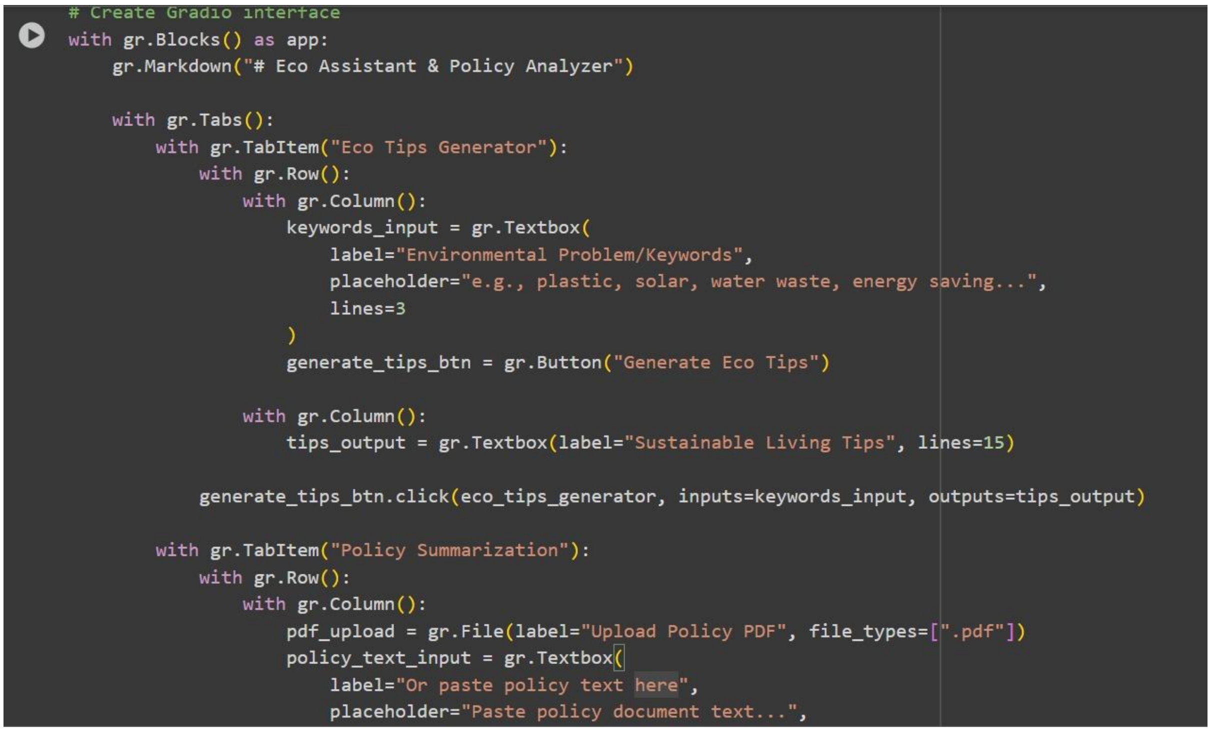


```
!pip install transformers torch gradio PyPDF2 -q

import gradio as gr
import torch
from transformers import AutoTokenizer, AutoModelForCausalLM
import PyPDF2
import io

# 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()}
```

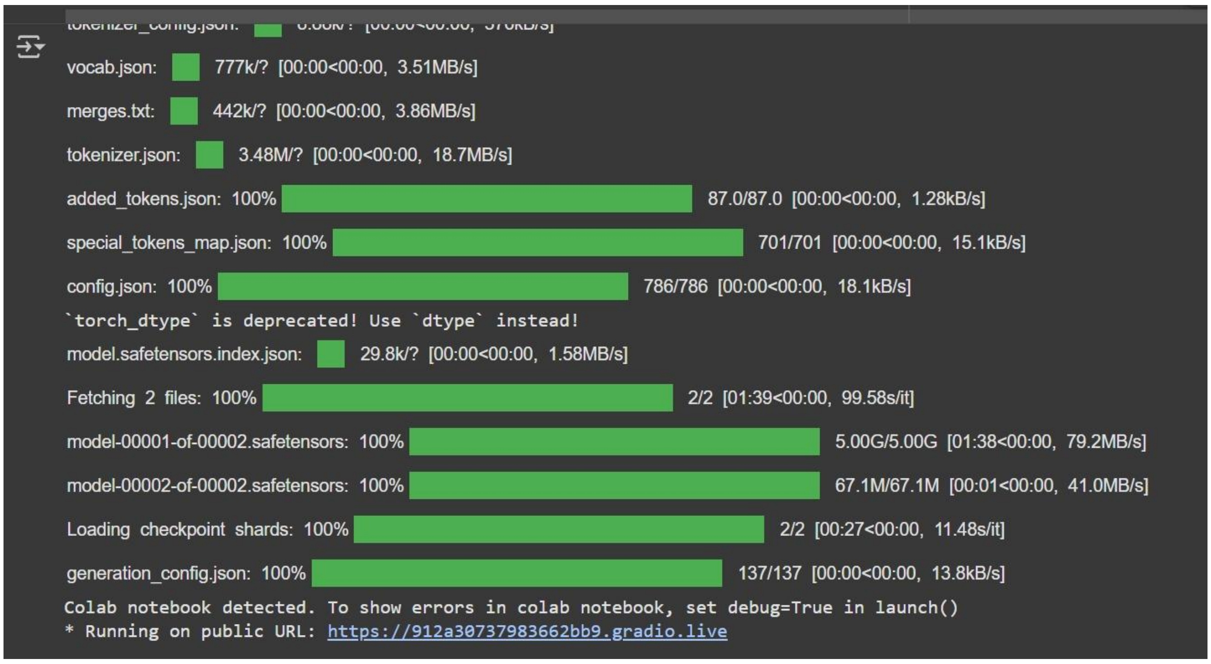


Eco Assistant & Policy Analyzer

Eco Tips Generator
Policy Summarization
Environmental Problem/Keywords
e.g., plastic, solar, water waste, energy saving...
Generate Eco Tips
Sustainable Living Tips



```python
# Create Gradio interface
with gr.Blocks() as app:
    gr.Markdown("# Eco Assistant & Policy Analyzer")

    with gr.Tabs():
        with gr.TabItem("Eco Tips Generator"):
            with gr.Row():
                with gr.Column():
                    keywords_input = gr.Textbox(
                        label="Environmental Problem/Keywords",
                        placeholder="e.g., plastic, solar, water waste, energy saving...",
                        lines=3
                    )
                    generate_tips_btn = gr.Button("Generate Eco Tips")

                with gr.Column():
                    tips_output = gr.Textbox(label="Sustainable Living Tips", lines=15)

            generate_tips_btn.click(eco_tips_generator, inputs=keywords_input, outputs=tips_output)

        with gr.TabItem("Policy Summarization"):
            with gr.Row():
                with gr.Column():
                    pdf_upload = gr.File(label="Upload Policy PDF", file_types=[".pdf"])
                    policy_text_input = gr.Textbox(
                        label="Or paste policy text here",
                        placeholder="Paste policy document text...",
```



```
tokenizer_config.json: [illegible]
vocab.json: 777k/? [00:00<00:00, 3.51MB/s]
merges.txt: 442k/? [00:00<00:00, 3.86MB/s]
tokenizer.json: 3.48M/? [00:00<00:00, 18.7MB/s]
added_tokens.json: 100% 87.0/87.0 [00:00<00:00, 1.28kB/s]
special_tokens_map.json: 100% 701/701 [00:00<00:00, 15.1kB/s]
config.json: 100% 786/786 [00:00<00:00, 18.1kB/s]
`torch_dtype` is deprecated! Use `dtype` instead!
model.safetensors.index.json: 29.8k/? [00:00<00:00, 1.58MB/s]
Fetching 2 files: 100% 2/2 [01:39<00:00, 99.58s/it]
model-00001-of-00002.safetensors: 100% 5.00G/5.00G [01:38<00:00, 79.2MB/s]
model-00002-of-00002.safetensors: 100% 67.1M/67.1M [00:01<00:00, 41.0MB/s]
Loading checkpoint shards: 100% 2/2 [00:27<00:00, 11.48s/it]
generation_config.json: 100% 137/137 [00:00<00:00, 13.8kB/s]
Colab notebook detected. To show errors in colab notebook, set debug=True in launch()
* Running on public URL: https://912a30737983662bb9.gradio.live
```

10. Conclusion

A Sustainable Smart City Assistant powered by IBM Granite combines the power of AI, IoT, and sustainable practices to create a smarter, greener, and more livable urban environment. By empowering citizens and city authorities, it contributes to achieving SDG goals and ensures a future-ready urban ecosystem.

THANK YOU