Sustainable Smart City Assistant using IBM Granite LLM

1.introduction:

- PROJECT TITLE: SUSTAINABLE SMART CITY ASSISTANT USING IBM GRANITE LLM.
- TEAM MEMBER: SADHANA S
- TEAM MEMBER:SAKTHI B
- TEAM MEMBER:SAMEERA BEGUM M
- TEAM MEMBER:SANDHIYA A

2.PROJECT OVERVIEW: The Sustainable Smart City Assistant is an AI-powered system designed to improve urban living by enabling intelligent, data-driven decision-making. It leverages IBM Granite Large Language Models (LLMs) to provide context-aware insights, natural language interactions, and sustainable solutions for city governance and citizen engagement.

3. Role of IBM Granite: LLM IBM Granite LLM provides advanced natural language understanding and generation capabilities. It powers conversational interfaces, decision support systems, and real-time policy simulations. Its enterprise-grade design ensures trust, transparency, and scalability for smart city applications.

4.AUTHENTICATION: Authentication

All endpoints require authentication via API key or OAuth 2.0. Use HTTPS for all requests.

Header (API Key):

Authorization: Bearer <API_KEY>

OAuth 2.0: Bearer tokens obtained via the token endpoint with client credentialgrant.

Rate limiting & Throttling

Default rate limits: • 60 requests per minute per API key (standard)

• 600 requests per minute (enterprise tier).

When limits are exceeded, API returns HTTP 429 with Retry-After header.

5. Use Cases • :

• **Mobility:** AI-driven traffic optimization and public transport planning .

- **Energy**: Smart grid demand prediction and renewable integration.
- **Waste:** Route optimization for collection and recycling .
- **Water**: Leak detection and efficient distribution management Governance: Policy simulation and automated reporting.
- **Citizen Services:** 24/7 multilingual digital assistant .

6. Sustainability Impact:

The assistant supports the United Nations Sustainable Development Goals (SDGs) by reducing carbon emissions, improving resource efficiency, and fostering inclusive governance. Predictive analytics enable cities to minimize energy waste, optimize water usage, and create greener mobility solutions.

7. Challenges & Future Scope: Key challenges include ensuring data privacy, achieving interoperability among diverse city systems, and managing scalability. Future advancements will involve deeper AI integration with edge computing, autonomous city operations, and continuous learning from citizen feedback.

8. Roles & Scopes:

Access control is managed via roles and scopes assigned to tokens.

Citizen Role:

- tips:read
- chat:read

Planner Role:

- policy:read
- analytics:read

Admin Role:

- * (all endpoints)

IoT Device Role:

- telemetry:write

6. Security Best Practices

* Always use HTTPS (TLS 1.2+)

* Rotate API keys every 90 days

* Use short-lived OAuth tokens in production

* Store secrets in vaults (e.g., AWS Secrets Manager, HashiCorp Vault)

* Enable audit logging for all authentication events

* Apply role-based access control (RBAC)

* Monitor failed login attempts and anomalies

SOURCE CODE:

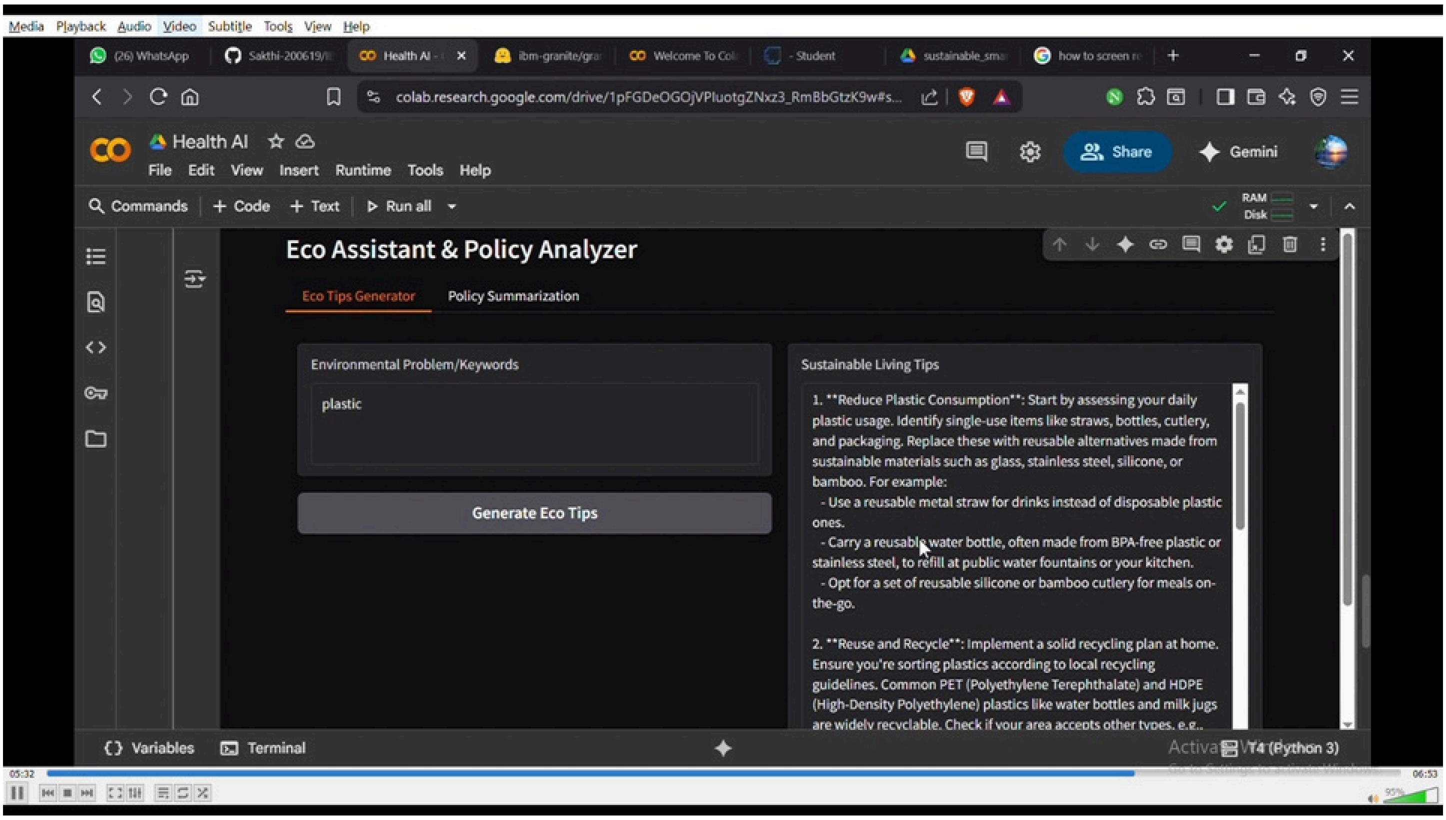
```
   import gradio as gr
import torch
from transformers import AutoTokenizer, AutoModelForCausalLM
import PyPDF2
import io
#Load model and tokenizer
model_name = "ibm-granite/granite-3.2-2b-instruct"
tokenizer = AutoTokenizer.from_pretrained(model_name)
model = AutoModelForCausalLM.from_pretrained(
    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.t(model.device) for k, v in inputs.items()}
with torch.no_grad():
  outputs = model.generate(
       **inputs,
       max_length=max_length,
       temperature=0.7,
       do_sample=True,
       pad_token_id=tokenizer.eos_token_id
  )
response=tokenizer.decode(outputs[0], skip_special_tokens=True)
response=response.replace(prompt, "").strip()
return response
def extract_text_from_pdf(pdf_file):
    if pdf_file is None:
      return ""
    try:
      pdf_reader=PyPDF2.PdfReader(pdf_file)
      text=""
      for page in pdf_reader.pages:
          text+=page.extract_text() + "\n"
      return text
    except Exception as e:
return f"Error reading POF: {str(e)}"
def eco_tips_generator (problem_keywords):
    prompt=f"Generate practical and actionable eco-friendly tips for sustainable living related to:{problem_keywords}.provide specific solutions and suggestions:"
    return generate_response(prompt, max_length=1000)
def policy_summarization(pdf_file, policy_text):
    #Get text from PDF or direct input
    if pdf_file is not None:
      content=extract_text_from_pdf(pdf_file)
      summary_prompt=f"Summarize the following policy document and extract the most important points, key provisions, and implications:\n\n{content}"
    else:
      summary_prompt=f"Summarize the following policy document and extract the most important points, key provisions, and implications:\n\n{policy_text}"
    return generate_response(summary_prompt, max_length=1200)
#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
           )
           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...",
              lines=5
            )
            summarize_btn=gr.Button("Summarize Policy")
          with gr.column():
            summary_output=gr.Textbox (label="Policy Summary & Key Points", lines=20)

        summarize_btn.click(policy_summarization, inputs=[pdf_upload, policy_text_input], outputs=summary_output)
app.launch(share=True)
```

OUTPUT:


Media Playback Audio Video Subtitle Tools View Help
colab.research.google.com/drive/1pFGDeOGOjVPluotgZNxz3_RmBbGtzK9w#s...
Health AI
File Edit View Insert Runtime Tools Help
Share
Gemini
Commands + Code + Text Run all
RAM
Disk
Eco Assistant & Policy Analyzer
Eco Tips Generator
Policy Summarization
Environmental Problem/Keywords
Generate Eco Tips
Sustainable Living Tips
Variables
Terminal
T4 (Python 3)
05:32
06:53

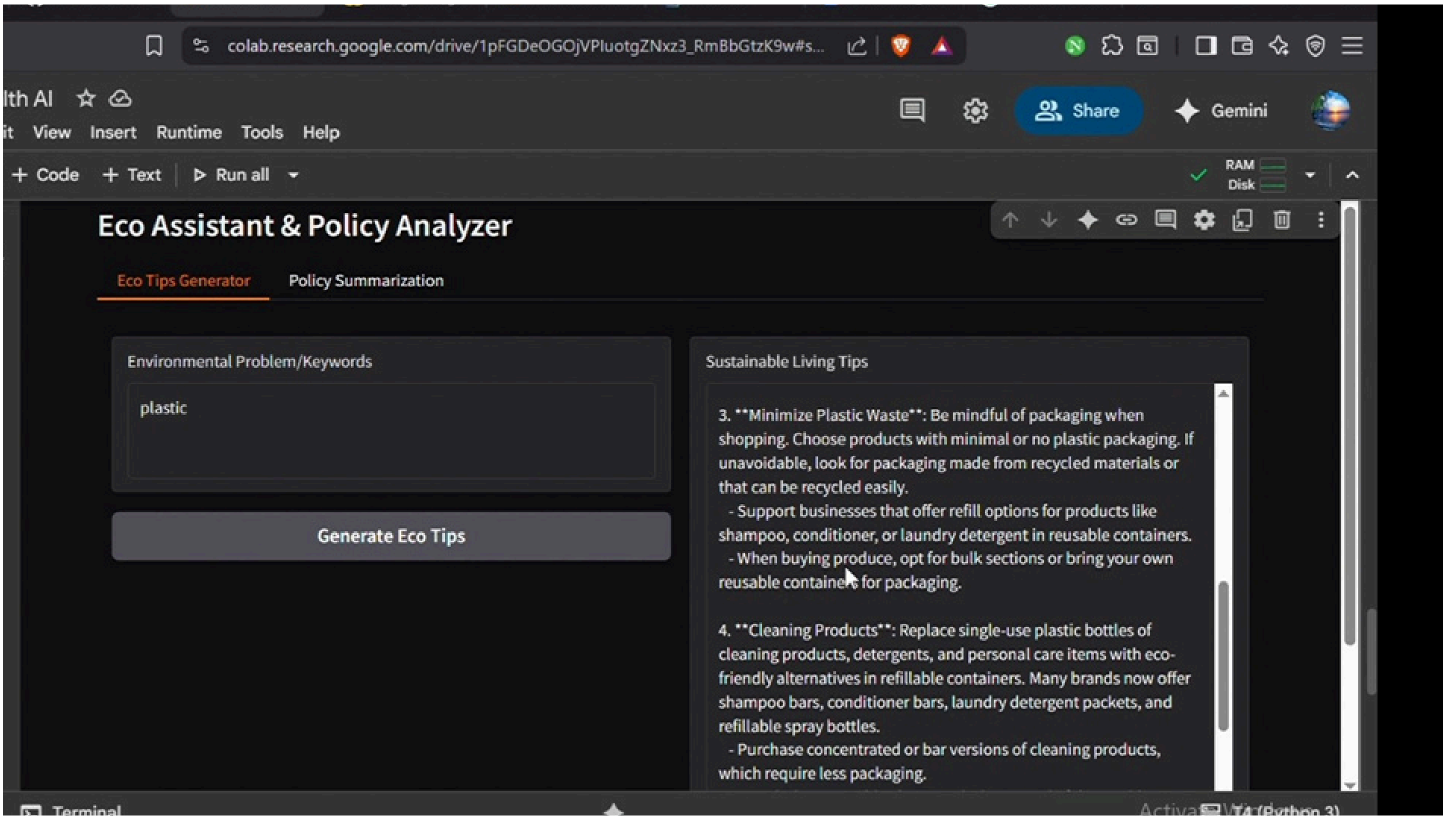
Environmental Problem/Keywords

plastic

Sustainable Living Tips

```
1. **Reduce Plastic Consumption**: Start by assessing your daily
plastic usage. Identify single-use items like straws, bottles, cutlery,
and packaging. Replace these with reusable alternatives made from
sustainable materials such as glass, stainless steel, silicone, or
bamboo. For example:
  - Use a reusable metal straw for drinks instead of disposable plastic
ones.
  - Carry a reusable water bottle, often made from BPA-free plastic or
stainless steel, to refill at public water fountains or your kitchen.
  - Opt for a set of reusable silicone or bamboo cutlery for meals on-
the-go.

2. **Reuse and Recycle**: Implement a solid recycling plan at home.
Ensure you're sorting plastics according to local recycling
guidelines. Common PET (Polyethylene Terephthalate) and HDPE
(High-Density Polyethylene) plastics like water bottles and milk jugs
are widely recyclable. Check if your area accepts other types, e.g.,
```



colab.research.google.com/drive/1pFGDeOGOjVPluotgZNxz3_RmBbGtzK9w#s...
lth AI
it View Insert Runtime Tools Help
Share
Gemini
+ Code + Text Run all
RAM
Disk

Eco Assistant & Policy Analyzer

Eco Tips Generator
Policy Summarization

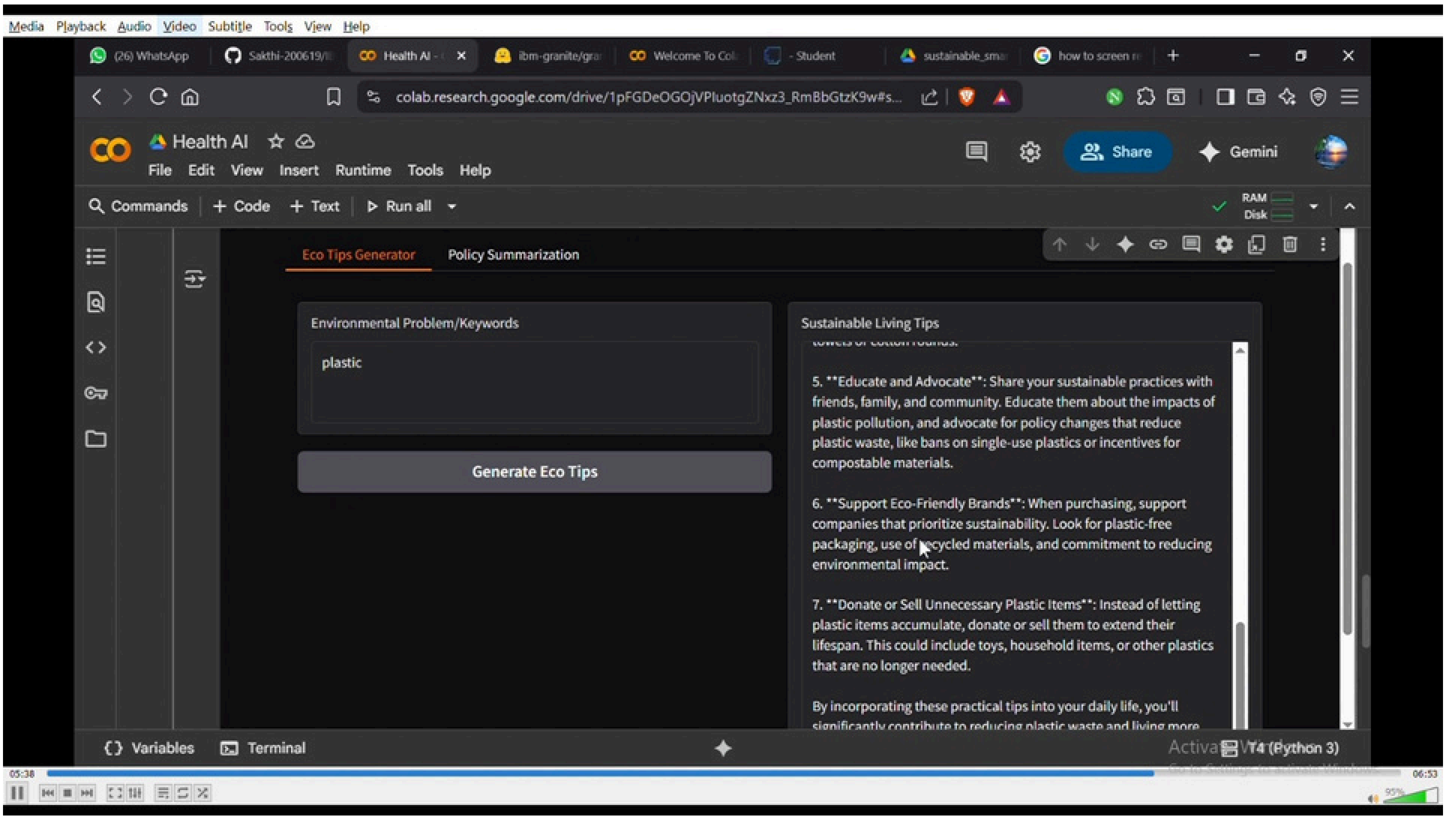
Environmental Problem/Keywords

plastic

Sustainable Living Tips

```
3. **Minimize Plastic Waste**: Be mindful of packaging when shopping. Choose products with minimal or no plastic packaging. If unavoidable, look for packaging made from recycled materials or that can be recycled easily.
   - Support businesses that offer refill options for products like shampoo, conditioner, or laundry detergent in reusable containers.
   - When buying produce, opt for bulk sections or bring your own reusable container for packaging.

4. **Cleaning Products**: Replace single-use plastic bottles of cleaning products, detergents, and personal care items with eco-friendly alternatives in refillable containers. Many brands now offer shampoo bars, conditioner bars, laundry detergent packets, and refillable spray bottles.
   - Purchase concentrated or bar versions of cleaning products, which require less packaging.
```



Media Playback Audio Video Subtitle Tools View Help
(26) WhatsApp
Sakthi-200619/
Health AI
ibm-granite/gra
Welcome To Col
- Student
sustainable_sma
how to screen
colab.research.google.com/drive/1pFGDeOGOjVPluotgZNxz3_RmBbGtzK9w#s...
Health AI
File Edit View Insert Runtime Tools Help
Share
Gemini
Commands + Code + Text Run all
RAM
Disk
Eco Tips Generator
Policy Summarization

Environmental Problem/Keywords

plastic

Sustainable Living Tips

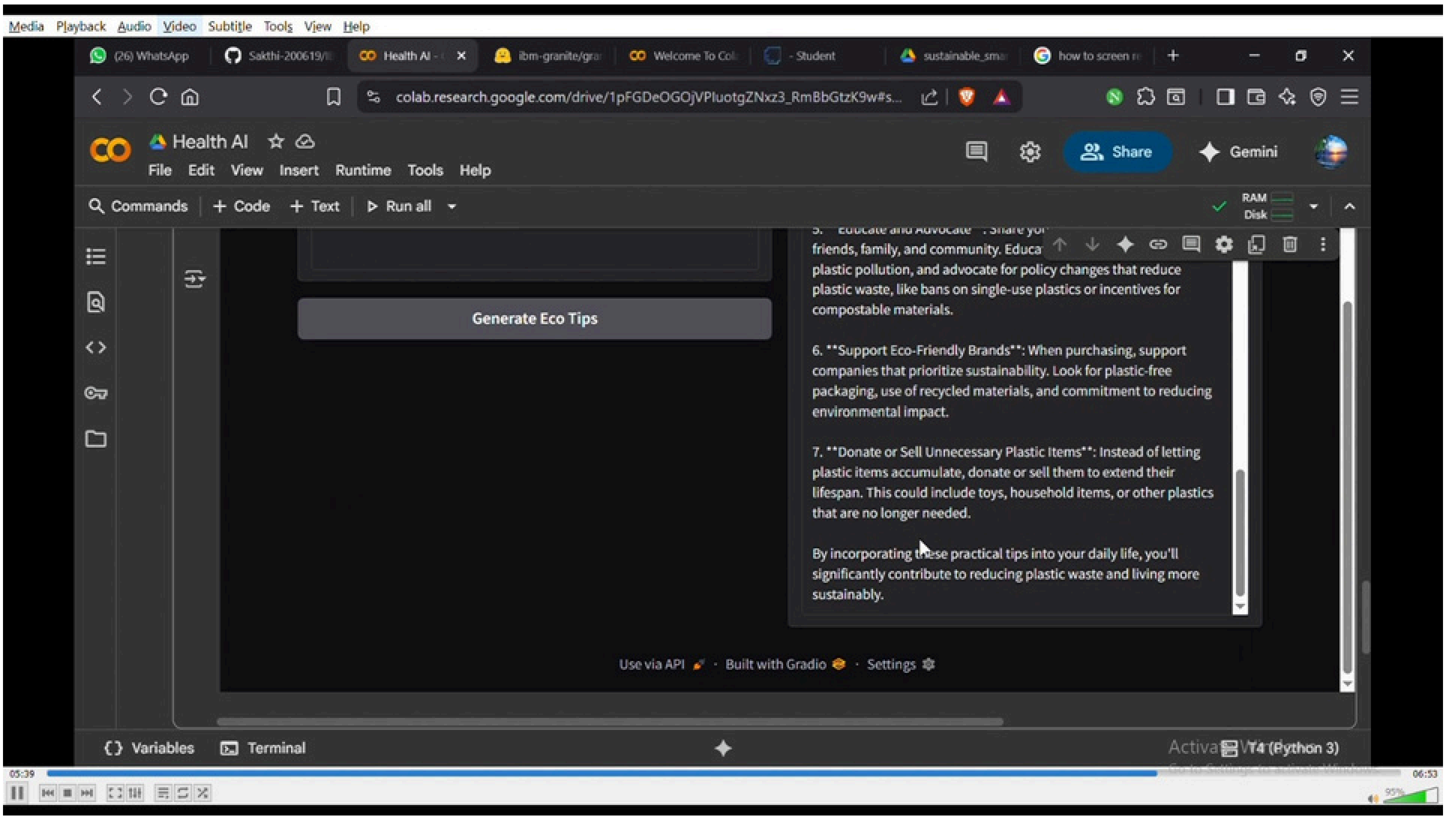
```
towels or cotton rounds.

5. **Educate and Advocate**: Share your sustainable practices with friends, family, and community. Educate them about the impacts of plastic pollution, and advocate for policy changes that reduce plastic waste, like bans on single-use plastics or incentives for compostable materials.

6. **Support Eco-Friendly Brands**: When purchasing, support companies that prioritize sustainability. Look for plastic-free packaging, use of recycled materials, and commitment to reducing environmental impact.

7. **Donate or Sell Unnecessary Plastic Items**: Instead of letting plastic items accumulate, donate or sell them to extend their lifespan. This could include toys, household items, or other plastics that are no longer needed.

By incorporating these practical tips into your daily life, you'll significantly contribute to reducing plastic waste and living more
```


Media Playback Audio Video Subtitle Tools View Help
colab.research.google.com/drive/1pFGDeOGOjVPluotgZNxz3_RmBbGtzK9w#s...
Health AI
File Edit View Insert Runtime Tools Help
Share
Gemini
Commands + Code + Text Run all
RAM
Disk
Generate Eco Tips
Use via API · Built with Gradio · Settings
Variables
Terminal
T4 (Python 3)
05:39
06:53

5. **Educate and Advocate**: Share yo
friends, family, and community. Educa
plastic pollution, and advocate for policy changes that reduce plastic waste, like bans on single-use plastics or incentives for compostable materials.

6. **Support Eco-Friendly Brands**: When purchasing, support companies that prioritize sustainability. Look for plastic-free packaging, use of recycled materials, and commitment to reducing environmental impact.

7. **Donate or Sell Unnecessary Plastic Items**: Instead of letting plastic items accumulate, donate or sell them to extend their lifespan. This could include toys, household items, or other plastics that are no longer needed.

By incorporating these practical tips into your daily life, you'll significantly contribute to reducing plastic waste and living more sustainably.

10. Conclusion:

By combining IBM Granite LLM with IoT and smart city platforms, the Sustainable Smart City Assistant provides a scalable, trustworthy, and citizen-focused solution to build more sustainable and livable urban environments.