



Team Members

This project was proudly developed by:

- **Dhanush**
- **Prithiviraj**
- **Nivesh**
- **Gokul Sakthi**

Their collaborative effort brought together AI, sustainability, and smart city innovation in one powerful tool.



Table of Contents

1. Team Members
 2. Project Overview
 3. Technologies Used
 4. Installation
 5. Model Details
 6. Functional Modules
 7. Gradio Interface
 8. Use Cases
 9. Launch Instructions
 10. Appendix (Optional Enhancements)
-

Project Overview

This project is a dual-purpose AI assistant designed to:

- Generate eco-friendly living tips based on environmental keywords.
- Summarize policy documents from uploaded PDFs or pasted text.

It uses a transformer-based language model and a Gradio-powered interface to make sustainability and policy analysis accessible to everyone.

3. Technologies Used

Component Purpose

Python Core programming language

Gradio Web-based user interface

Transformers Model loading and inference

Torch Deep learning backend

PyPDF2 PDF text extraction

4. Installation

Install all required packages using:

```
pip install transformers torch gradio PyPDF2
```

5. Model Details

- **Model Name:** ibm-granite/granite-3.2-2b-instruct
 - **Framework:** Hugging Face Transformers
 - **Device Support:** Automatically uses GPU if available
 - **Token Handling:** Uses EOS token as pad token if none is defined
-

6. Functional Modules

`generate_response(prompt, max_length=1024)`

Generates a response from the model using temperature sampling and truncation.

extract_text_from_pdf(pdf_file)

Extracts text from each page of a PDF using PyPDF2.

eco_tips_generator(problem_keywords)

Creates actionable sustainability tips based on user-provided keywords.

policy_summarization(pdf_file, policy_text)

Summarizes policy documents from either uploaded PDFs or pasted text.

7. Gradio Interface

Tab 1: Eco Tips Generator

- **Input:** Environmental keywords
- **Output:** Sustainable living suggestions
- **Button:** "Generate Eco Tips"

Tab 2: Policy Summarization

- **Input:** PDF upload or text input
 - **Output:** Summary with key points
 - **Button:** "Summarize Policy"
-

8. Use Cases

- Educating citizens on eco-conscious practices
 - Assisting policymakers with quick document analysis
 - Promoting sustainability in smart city initiatives
-

9. Launch Instructions

To start the app:

```
app.launch(share=True)
```

This enables public access to the Gradio interface.

10. Appendix: Optional Enhancements

- Add multilingual support for global accessibility
 - Include citation extraction for policy documents
 - Integrate with smart city dashboards or APIs
-