**SUSTAINABLE SMART CITY**

**DOCUMENTATION**

**1.Introduction**

**Team Leader: Mahalakshmi R**

**Team Member: Asra H**

**Team Member: Pooja sri V**

**Team Member: Gopika S**

**Team Member: Harini R**

**2. Project overview**

The presented program integrates **artificial intelligence (AI)**, **natural language processing (NLP)**, and a **graphical web interface** to create an application that can generate eco-friendly living tips and summarize policy documents. The solution leverages open-source libraries such as **PyTorch**, **Transformers**, **PyPDF2**, and **Gradio**.

The main objective of this project is twofold:

1. **Eco Tips Generator** – helps individuals and communities adopt sustainable living practices by generating context-based suggestions.
2. **Policy Summarization** – assists users in understanding lengthy policy documents by providing concise summaries with key points and implications.

By combining these two functionalities, the program demonstrates how modern AI models can support sustainability, environmental awareness, and informed decision-making.

**3. Technologies Used**

**a) Python**

The program is written in Python, a versatile language popular for machine learning, NLP, and automation.

**b) PyTorch**

PyTorch is used as the underlying deep learning framework. It enables the AI model to perform text generation tasks efficiently on both CPU and GPU.

**c) Transformers Library**

This library, developed by Hugging Face, provides easy access to pre-trained large language models (LLMs). In this program, the **IBM Granite 3.2 Instruct Model** (ibm-granite/granite-3.2-2b-instruct) is used. This model specializes in generating human-like text responses based on prompts.

**d) PyPDF2**

PyPDF2 allows the program to extract raw text from PDF documents. This is especially useful for policy summarization, where documents are often lengthy and available only in PDF format.

**e) Gradio**

Gradio is used to create an **interactive web interface**. With just a few lines of code, users can interact with the AI model, upload PDFs, input keywords, and view results in a user-friendly manner.

**4. Program Workflow**

The workflow of the program can be divided into the following stages:

**Step 1: Model Initialization**

* The tokenizer and model are loaded from Hugging Face’s model hub.
* The program automatically detects whether a **GPU** is available. If yes, it loads the model in half-precision (float16) for efficiency. Otherwise, it runs on CPU.

**Step 2: Input Handling**

* For Eco Tips: The user enters **keywords** related to an environmental issue (e.g., “plastic waste”, “solar energy”).
* For Policy Summarization: The user either uploads a **PDF file** or pastes policy text directly into the interface.

**Step 3: Processing**

* If a PDF is uploaded, PyPDF2 extracts the text from each page.
* A prompt is then constructed and passed to the AI model through the **generate\_response()** function.
* The model generates a relevant response, either eco tips or a policy summary.

**Step 4: Output Display**

* The result is displayed inside a **Gradio textbox**.
* Users can directly copy or save the generated output for further use.

**5. Core Functionalities**

**a) Eco Tips Generator**

* Purpose: To encourage sustainable living by providing practical suggestions.
* Input: Keywords such as “plastic pollution”, “energy saving”, “water conservation”.
* Output: A list of **actionable tips**, like reducing single-use plastics, installing solar panels, or conserving water.

**b) Policy Summarization**

* Purpose: To simplify lengthy policy documents into short, understandable notes.
* Input: Either a **PDF upload** or raw policy text.
* Output: Summarized key points, major provisions, and implications.
* Benefit: Saves time and helps non-experts understand complex legal or governmental documents.

**6. User Interface Design**

The interface is designed with **two main tabs**:

1. **Eco Tips Generator Tab**
   * Input box for environmental keywords.
   * Button to generate tips.
   * Output textbox to display results.
2. **Policy Summarization Tab**
   * Option to upload a PDF file.
   * Alternative option to paste raw policy text.
   * Button to summarize policy.
   * Output textbox showing the summary.

This design ensures **simplicity, usability, and clarity**. Even non-technical users can interact with the tool easily.

**7. Advantages of the Program**

1. **Dual Functionality** – Combines sustainability guidance with policy analysis in a single application.
2. **AI-Powered** – Leverages the IBM Granite model for intelligent and context-aware responses.
3. **User-Friendly** – Gradio provides a web-based interface without requiring technical expertise.
4. **Time-Saving** – Reduces hours of reading policies into minutes of summarization.
5. **Practical Impact** – Encourages eco-friendly lifestyle changes while simplifying governance policies.

**8. Limitations**

1. **Model Dependency** – The quality of responses depends heavily on the pre-trained AI model.
2. **PDF Text Extraction Issues** – Some scanned or image-based PDFs may not extract text properly.
3. **Generalized Suggestions** – Tips and summaries may not always be region-specific or fully accurate.
4. **Hardware Requirements** – Running on CPU may be slow for long documents.

**9. Future Enhancements**

* **Integration with OCR** to handle scanned PDFs.
* **Multi-language support** for global accessibility.
* **Database storage** to save past summaries and tips.
* **Interactive graphs/charts** for visual representation of eco solutions.
* **Voice-enabled interface** for accessibility.

**10. Conclusion**

The **Eco Assistant & Policy Analyzer** program demonstrates the potential of combining **AI-driven text generation** with practical applications in sustainability and governance. By enabling users to generate eco-friendly tips and summarize policy documents, it highlights how artificial intelligence can empower individuals and communities to make informed and responsible choices.

This project bridges the gap between **complex information** and **practical action**, making AI not only a tool for research but also a **companion for everyday problem-solving**.