Smart Report Generator

The **Smart Report Generator** is a Python-based web application that automates the creation of structured reports from raw text input. Developed using Flask for the backend and HTML/CSS for the frontend, it provides a user-friendly interface where users can input text and receive a formatted report as output. The application focuses on reducing manual effort in report writing by processing and organizing the content automatically. It's ideal for generating summaries, academic reports, or professional documents with minimal user intervention.

🔧 Project Structure

The repository is organized into several key components:

- **`app.py`**: The main Flask application script that defines routes and handles HTTP requests.
- **`utils.py`**: Contains helper functions for processing and formatting input text.
- **`templates/`**: Holds HTML templates used to render the web pages.
- **`static/`**: Includes static assets like CSS files for styling the frontend.
- **`sample_input.txt`**: Provides example input data to demonstrate the application's functionality.
- **`report_data.txt`**: Stores the generated report output.
- **`requirements.txt`**: Lists the Python dependencies required to run the application.

⚙️ Functionality

While the repository lacks detailed documentation, the application's workflow can be inferred:

1. **Input**: Users provide raw text data, either by uploading a file or entering text directly into the web interface.
2. **Processing**: The application processes the input text using functions defined in `utils.py`, which may include parsing, summarization, or formatting operations.
3. **Report Generation**: A structured report is generated and saved to `report_data.txt`.
4. **Output**: The generated report is displayed to the user through the web interface.

🚀 Getting Started

To set up and run the application locally:

1. **Clone the Repository**:

   ```
   git clone https://github.com/VinitaChowkekar/Smart-Report-Generator.git
   cd Smart-Report-Generator
   ```

2. **Install Dependencies**:

   ```
   pip install -r requirements.txt
   ```

3. **Run the Application**:

   ```
   python app.py
   ```

4. **Access the Web Interface**:

   Open your web browser and navigate to `http://localhost:5000` to use the application.

📌 Use Cases

The Smart Report Generator can be utilized in various scenarios, including:

- **Academic Summaries**: Transforming lecture notes or research findings into concise reports.
- **Business Reports**: Converting meeting notes or raw data into structured business reports.
- **Data Analysis**: Summarizing analytical findings into readable formats for stakeholders.