**Task: Automated Essay Generation from CSV Data and User Prompts via Flask API**

**Objective**: Develop a Python program that generates an essay based on the content of a provided CSV file and user prompts, with data and prompts being delivered via a Flask-based API. The program should utilize a language model, such as LLM (e.g., Illama 2 or Vicuna), to create coherent and contextually relevant essays.

**Requirements**:

**Flask API:**

* Develop a Flask-based API that exposes endpoints for receiving user prompts and the CSV file.
* Define API routes for sending prompts and uploading the CSV file.

**CSV Data Handling via API:**

* Implement an API endpoint to accept and process the uploaded CSV file. The CSV data should be stored in a suitable data structure (e.g., a dictionary) for easy access.

**User Input via API:**

* Implement an API endpoint for accepting user prompts or questions as input.
* Language Model Integration:
* Integrate a pre-trained LLM model (e.g., Illama 2 or Vicuna) using a suitable Python library like Hugging Face Transformers.

**Generating Essay:**

* Use the language model to generate essay content based on the user prompts and the data from the CSV file.
* Ensure that the generated essay incorporates the words from the CSV file with their specified occurrences.

**Essay Structure:**

* Organize the essay into coherent paragraphs and sections.
* Provide an introduction, body, and conclusion as appropriate for the essay format.

**API Output:**

* Implement an API endpoint to return the generated essay as output. The essay should be sent back to the web API for retrieval by the user.

**Testing and Validation:**

* Test the Flask-based API with various user prompts and CSV data uploads to verify that the generated essays are contextually relevant and contain the specified words and occurrences.

**Documentation:**

* Provide clear documentation on how to use the Flask-based API, including instructions for invoking the API, sending prompts, and retrieving the generated essay.

**Submission:**

The candidate should submit the following:

* The Python code for the Flask-based API and the program that generates essays.
* Any necessary data files, including the CSV file used for word occurrences.
* Detailed documentation explaining how to use the Flask-based API, including API endpoints, input formats, and retrieval of generated essays.

**Note:** The use of a Flask-based API allows for more convenient interaction with the program by providing endpoints for data input and essay retrieval. The candidate should ensure that the API is robust, handles errors gracefully, and provides clear responses.