Sentiment Analysis Project Overview

1. Introduction

This project is designed to analyze the sentiment of user reviews by accepting CSV or XLSX files, processing them, and sending the reviews to an external sentiment analysis API (Groq API).
The results are presented as a summary of positive, negative, and neutral sentiments.

2. Technologies Used

- Python: The programming language used for the backend.
- Django: The web framework used to build the application.
- Pandas: For handling and processing the CSV/XLSX files.
- Groq API: For performing sentiment analysis on the reviews.
- HTML/CSS/JS: For the frontend form and user interaction.

3. Project Flow

1. **User Uploads File**:
   - The user uploads a CSV or XLSX file through the form on the webpage.
   - Supported file types: CSV, XLSX.

2. **File Processing**:
   - The backend reads the uploaded file using the Pandas library.
   - It checks for the presence of a 'Review' column in the file.

3. **Sentiment Analysis**:

   - Each review is sent to the Groq API for sentiment analysis.

   - The API returns a score for positive, negative, and neutral sentiments.

4. **Displaying Results**:

   - The results are calculated and sent back to the user in JSON format, showing the total sentiment breakdown.

4. Key Features

- Supports both CSV and XLSX file formats.
- Provides real-time feedback on sentiment for uploaded reviews.
- Includes user-friendly form with drag-and-drop file upload.
- Uses background tasks for long-running processes (via Django Celery).

5. How to Run the Project

1. Clone the repository.
2. Install the required dependencies using pip:

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

3. Set the Groq API Key as an environment variable:

   ```
   export GROQ_API_KEY='your-api-key'
   ```

4. Run the Django development server:

```
python manage.py runserver
```

5. Access the web app via `http://127.0.0.1:8000/` and upload your review file for sentiment analysis.

6. Additional Notes

- Ensure that the Groq API Key is valid and correctly set in the environment variables.
- This project supports further scaling using asynchronous tasks for file processing and sentiment analysis.
- Frontend includes a loading spinner for better UX during file processing.