Project Overview

The **Customer Feedback Analyzer** is a **full-stack web application** designed to help software enterprise companies analyze and derive actionable insights from customer feedback. The application takes a **CSV file** containing raw customer feedback as input, processes it using **Natural Language Processing (NLP)** techniques, and generates a **comprehensive report** with the following key components:

1. **Summarized Reviews**:
   * Positive, negative, and neutral reviews with concise summaries.
2. **Analysis Areas**:
   * Categorization of feedback into areas like **bugs**, **UI**, **performance**, **errors**, etc.
3. **Overall Customer Satisfaction**:
   * A summary of overall customer satisfaction based on feedback sentiment.
4. **Areas to Work On**:
   * Highlighted areas that need improvement based on recurring issues in feedback.

The application is designed to be **user-friendly**, **scalable**, and **efficient**, making it a valuable tool for businesses to improve their products and services.

**Key Features:**

1. **CSV File Upload**:
   * Users can upload a CSV file containing customer feedback.
   * The file should have a column named "Feedback" for analysis.
2. **Sentiment Analysis**:
   * Classifies feedback into **positive**, **negative**, and **neutral** categories.
   * Uses state-of-the-art NLP models like **BERT** or **RoBERTa**.
3. **Summarization**:
   * Generates concise summaries for each sentiment category using models like **Mystral AI**.
4. **Topic Modeling**:
   * Categorizes feedback into areas using models like **Mystral AI**.
5. **Overall Satisfaction Report**:
   * Provides an overall satisfaction report that is generated based on the reviews using models like **Mystral AI**.
6. **Areas to Work On**:
   * Identifies key areas that need improvement based on recurring issues in negative feedback.
7. **Export Options**:
   * Allows users to export the report as a **PDF** or **Excel** file for sharing.

**Technical Stack:**

**Frontend:**

* **Chart.js**: For creating visualizations like bar charts, pie charts, and word clouds.
* **HTML/CSS**: For structuring and styling the web pages.

**Backend:**

* **Python**: Core programming language for backend and NLP tasks.
* **Flask/FastAPI**: For building REST APIs to handle file uploads and process feedback.
* **Pandas**: For reading and processing CSV files.

**NLP:**

* **Hugging Face Transformers**: For sentiment analysis and summarization using pre-trained models like BERT and GPT.
* **SpaCy**: For text preprocessing and topic modeling.
* **NLTK**: For basic NLP tasks like tokenization and stopword removal.

**How It Works:**

1. **User Uploads CSV File**:
   * The user uploads a CSV file containing customer feedback through the web interface.
2. **Backend Processing**:
   * The backend reads the CSV file using Pandas.
   * Performs **sentiment analysis** to classify feedback into positive, negative, and neutral.
   * Generates **summaries** for each sentiment category.
   * Uses **topic modeling** to categorize feedback into areas like bugs, UI, performance, etc.
   * Calculates an **overall satisfaction score** and identifies **areas to work on**.
3. **Frontend Display**:
   * The frontend displays the analyzed data in a user-friendly dashboard.
   * Includes visualizations like bar charts, pie charts, and word clouds.
4. **Export Report**:
   * The user can export the report as a PDF or Excel file for further use.