

****

**1.Introduction:**

Sentilytics is an AI-powered sentiment analysis tool designed for real-time analysis of user-generated comments. The system categorizes comments into positive, negative, or neutral sentiments. It supports both single comment analysis and bulk analysis via CSV/Excel uploads. Users can visualize sentiment distribution using bar graphs and word clouds. The project comprises a React.js frontend and a Django backend with a custom sentiment analysis model.

**2.Project Profile:**

**2.1 Project Description:**

Our system simplifies sentiment analysis by allowing users to analyze comments individually or in bulk. Users can either input a single comment or upload a CSV file containing multiple comments, and the system will classify them as **positive, negative, or neutral**.

For guest users, a single comment analysis is available without registration. Registered users, on the other hand, can perform batch analyses, visualize results through **bar graphs and word clouds**, and even correct misclassified sentiments to improve model accuracy.

Admins have access to user activity, and comment classification results. The Sentilytics platform ensures a **user-friendly experience**, helping businesses and individuals understand customer opinions in a structured and efficient manner.

**2.2 Project Modules:**

1. Single Comment Analysis:

* Users can analyze individual comments.

1. Multiple Comments Analysis:

* Supports CSV/Excel file uploads for bulk analysis.

1. YouTube Comments Analysis:

* Fetches and analyzes YouTube comments using the Google API.

1. Export Results :

* Users can download results in Excel format.

1. Graphical Representations:

* Bar graphs for sentiment distribution.
* Word clouds for frequent terms in positive reviews.

1. Manual Sentiment Editing :

* Users can correct misclassified sentiments to improve model accuracy.

1. User Management :

* Guest users can analyze single comments, while registered users have access to batch analysis and sentiment corrections.

1. Database Storage :

* Stores all analyses with timestamps, comment sources, and sentiment results.

**2.3 Technology Stack:**

* Frontend: React.js
* Backend: Django (Python)
* Database: PostgreSQL
* Machine Learning Model: Custom-built using Logistic Regression
* Datasets: Sentiment140, Amazon Product Review

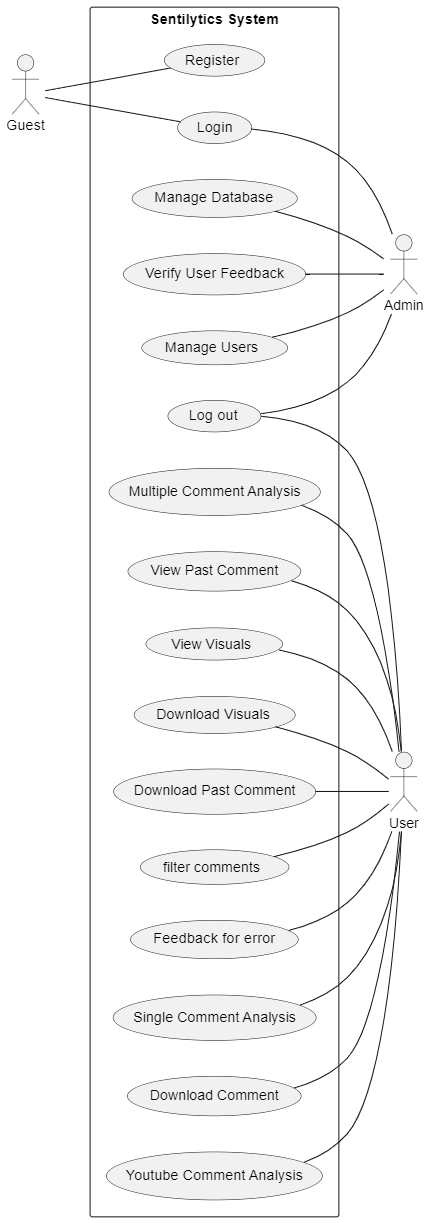
**2.4 Implementation Details:**

Sentilytics follows a structured workflow for sentiment analysis:

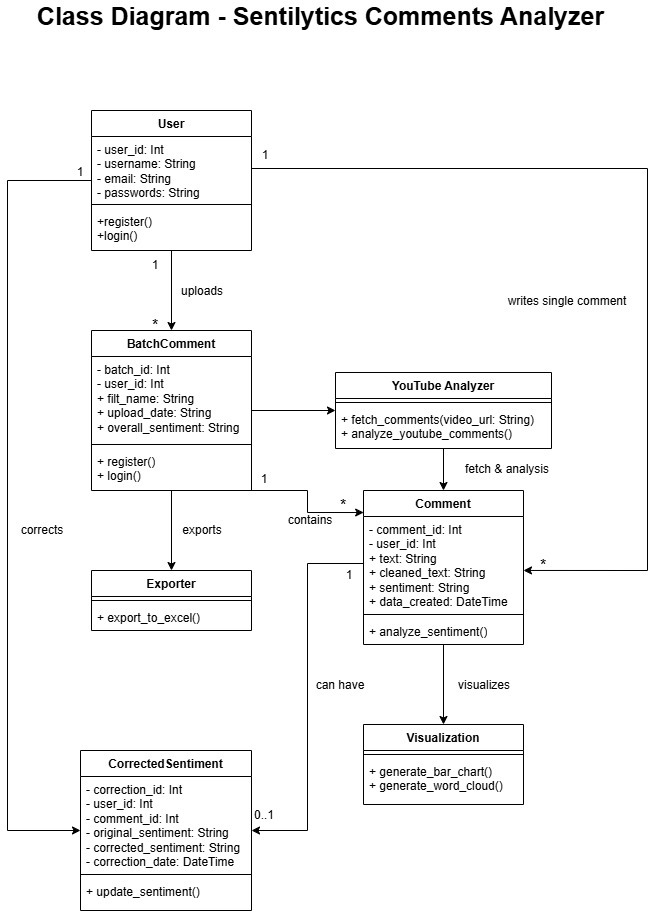
1. Preprocessing: Text is cleaned by removing stopwords, special characters, and applying tokenization.
2. Feature Extraction: TF-IDF vectorization converts text into numerical format for analysis.
3. Model Prediction: The Logistic Regression model predicts sentiment polarity.
4. Result Storage: Analysis results are stored in the database for retrieval and visualization.
5. User Feedback Loop: Users can correct sentiments, which are stored to refine future model training.

**3.UML Diagrams:**

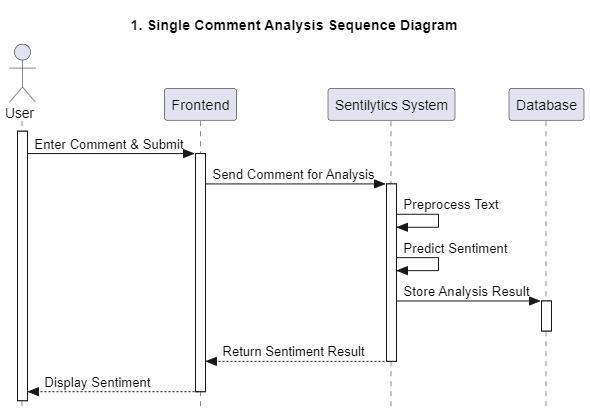
**3.1 Use-case Diagram:**

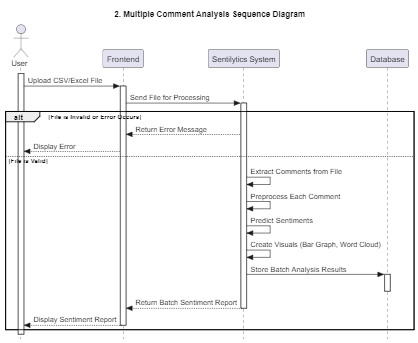
****

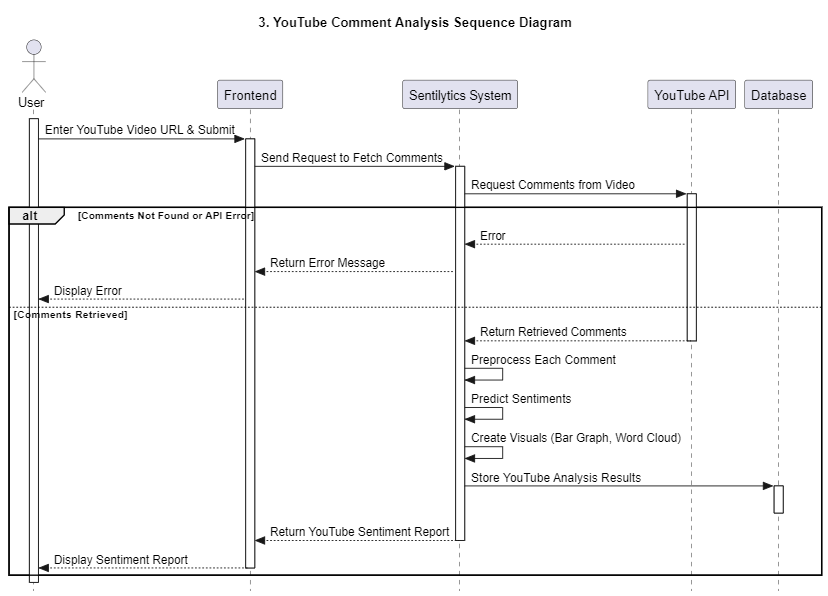
**3.2 Class Diagram:**

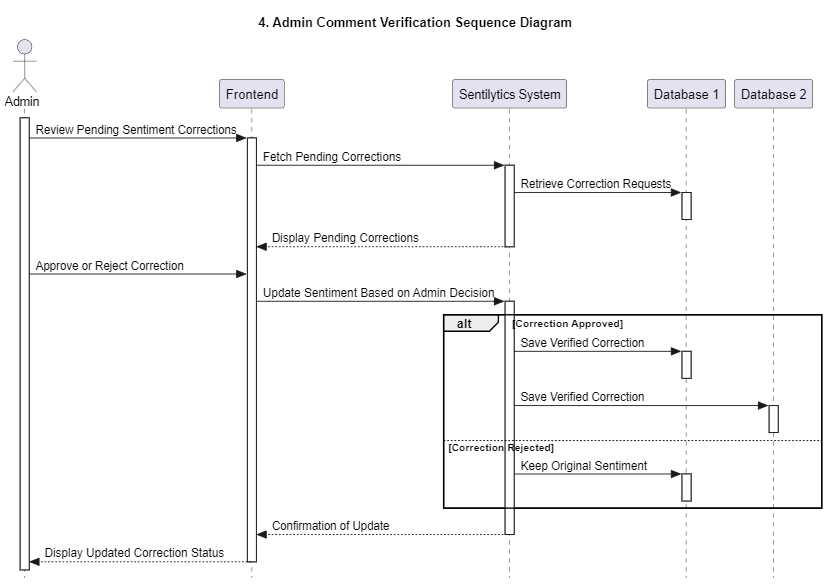


**3.3 Sequence Diagram:**

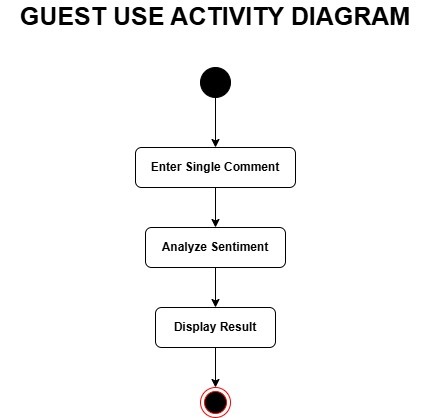


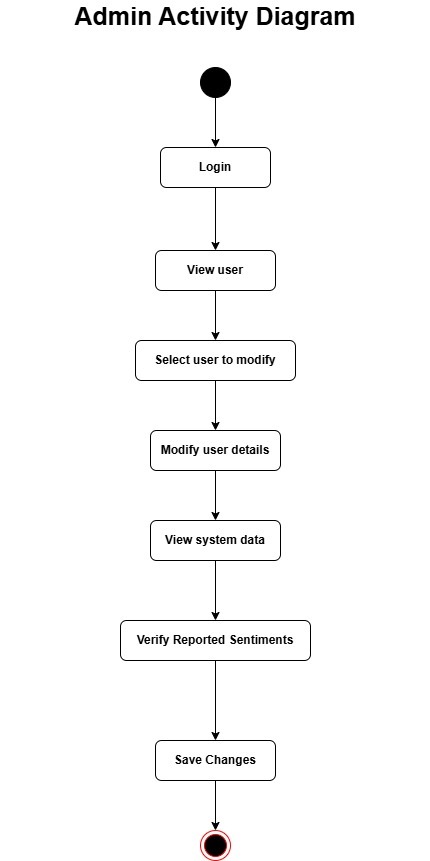
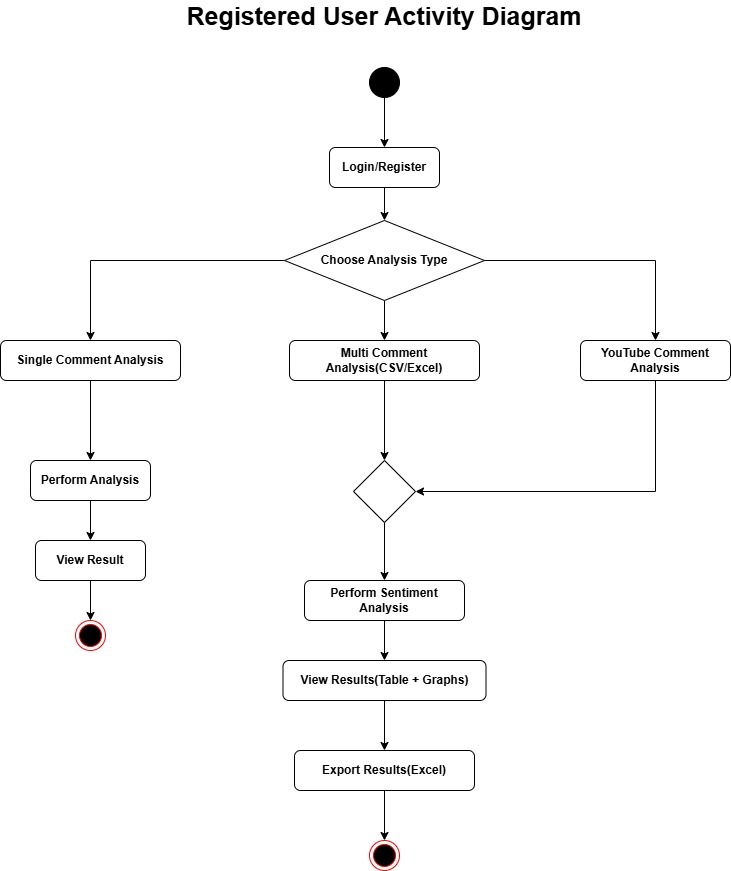






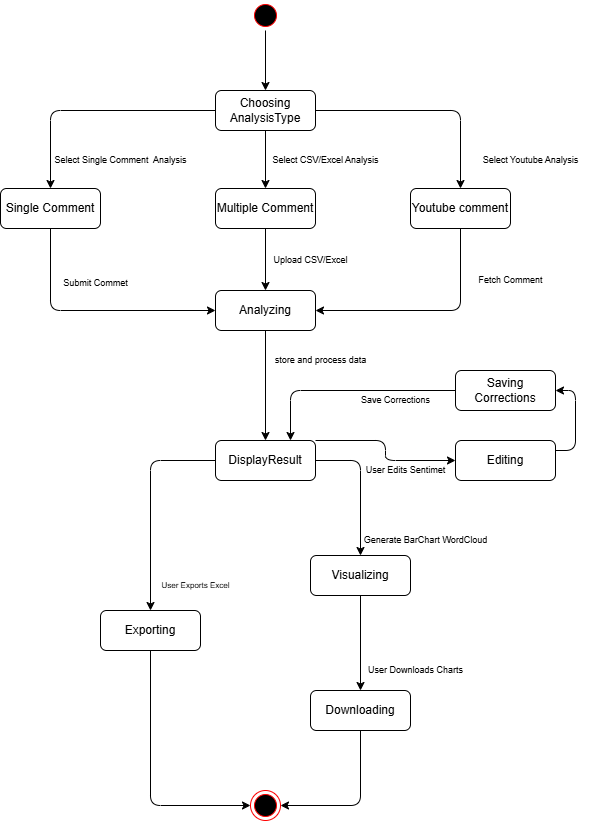
**3.4 Activity Diagram:**





**3.5 State Chart Diagram**

**Comments State**



|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Constraints** | **Description** |
| id | Integer (Auto) | Primary Key | Unique user ID |
| username | Varchar | Unique, Required | Username for authentication |
| email | Varchar | Unique, Required | User's email |
| password | Varchar | Required, Non-Nullable | Encrypted user password |
| is\_staff | Boolean | Default=False, Non-Nullable | Determines if user has admin access |
| is\_superuser | Boolean | Default=False, Non-Nullable | Determines if user has full access |
| Date\_joined | DateTime | Auto Timestamp, Non-Nullable | Date when the user registered |

**4.DATA DICTIONARY**

**4.1 USER TABLE:**

**4.2 TOKEN TABLE:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Constraints** | **Description** |
| key | Char | Primary Key | Authentication token for API access |
| user\_id | Integer | Foreign Key (auth\_user.id), Non-Nullable | Links token to a user |
| created | DateTime | Auto Timestamp, Non-Nullable | Timestamp when the token was created |

**4.3 COMMENT TABLE:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Constraints** | **Description** |
| id | Integer (Auto) | Primary Key, Non-Nullable | Unique comment ID |
| user\_id | Integer | Foreign Key (auth\_user.id), Non-Nullable | User who analyzed the comment |
| batch\_id | Integer | Foreign Key (analysis\_batchcomment.id), Default=None,Nullable | Links to batch analysis (if applicable) |
| comment | Text | Required, Non-Nullable | Original comment text |
| cleaned\_text | Text | Auto-generated, Non-Nullable | Processed text after cleaning |
| sentiment | Varchar | Required, Non-Nullable | Predicted sentiment |
| score | Float | Auto-generated, Non-Nullable | Sentiment score |
| date\_created | DateTime | Auto Timestamp, Non-Nullable | Timestamp when the comment was analyzed |
| updated\_at | DateTime | Auto Timestamp | Timestamp when the comment was last updated |
| is\_updated | Boolean | Default=False, Non-Nullable | Indicates if sentiment was manually corrected |
| comment\_type | Varchar | Required,Default='single',Choices=(single,batch) | Shows type of comment (Single,Batch) |

**4.4 BATCH COMMENTS**

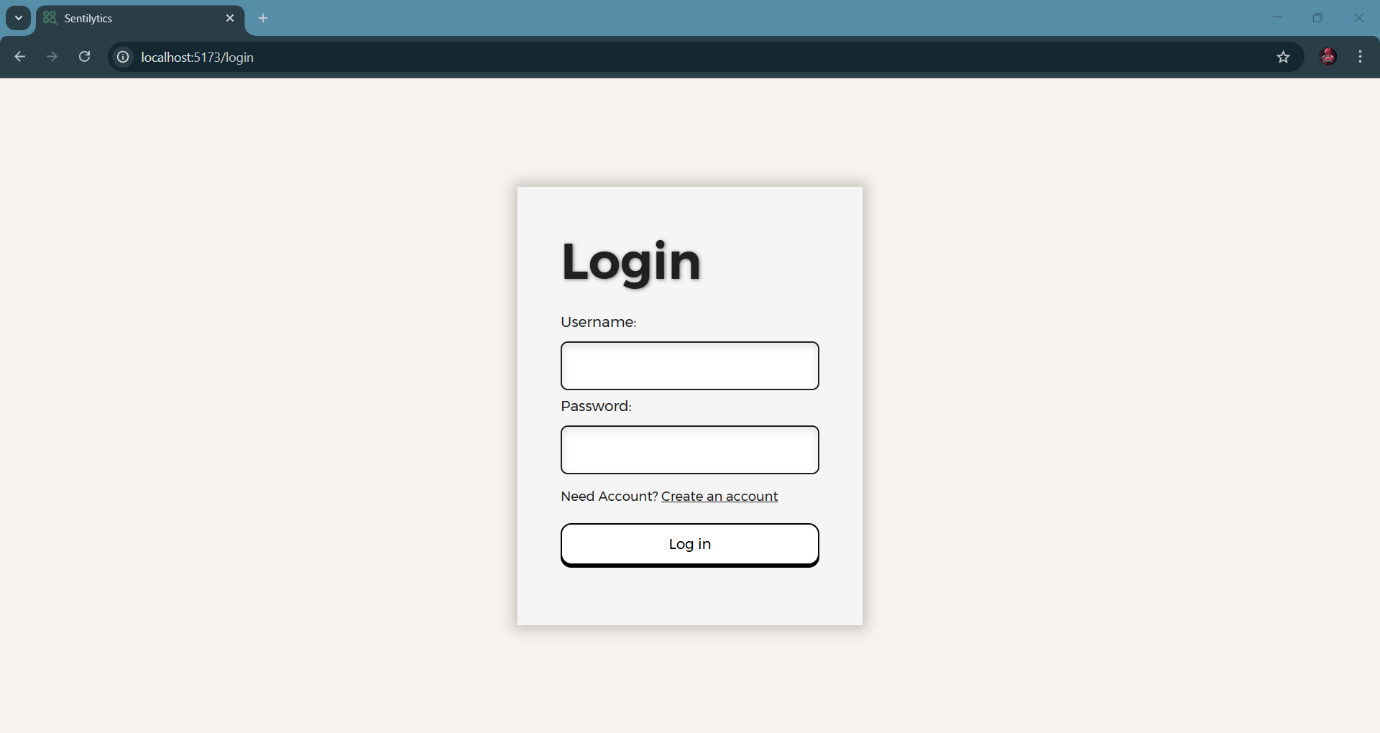
|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Constraints** | **Description** |
| id | Integer (Auto) | Primary Key, Non-Nullable | Unique batch ID |
| user\_id | Integer | Foreign Key (auth\_user.id), Non-Nullable | User who performed batch analysis |
| comment\_type | Varchar | Required, Non-Nullable | Source type (CSV File, Excel File, YouTube) |
| overall\_sentiment | Varchar | Auto-generated, Non-Nullable,Choices(positive,negative,neutral) | Aggregated sentiment for batch |
| date\_created | DateTime | Auto Timestamp, Non-Nullable | Timestamp when batch analysis was performed |

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Constraints** | **Description** |
| id | Integer (Auto) | Primary Key, Non-Nullable | Unique correction ID |
| comment\_id | Integer | Foreign Key (analysis\_comment.id), Non-Nullable | Links to the corrected comment |
| comment\_text | Varchar | Required, Non-Nullable | Original comment text |
| user\_id | Integer | Foreign Key (auth\_user.id), Non-Nullable | User who corrected the sentiment |
| predicted\_sentiment | Varchar | Required, Non-Nullable | model predicted sentiment value |
| corrected\_sentiment | Varchar | Required, Non-Nullable | Corrected sentiment value |
| feedback\_verified | Boolean | Default=False, Nullable | Indicates if admin verified correction(True=valid correct,False=Invalid,Null =pending) |
| date\_corrected | DateTime | Auto Timestamp, Non-Nullable | Timestamp when sentiment was corrected |

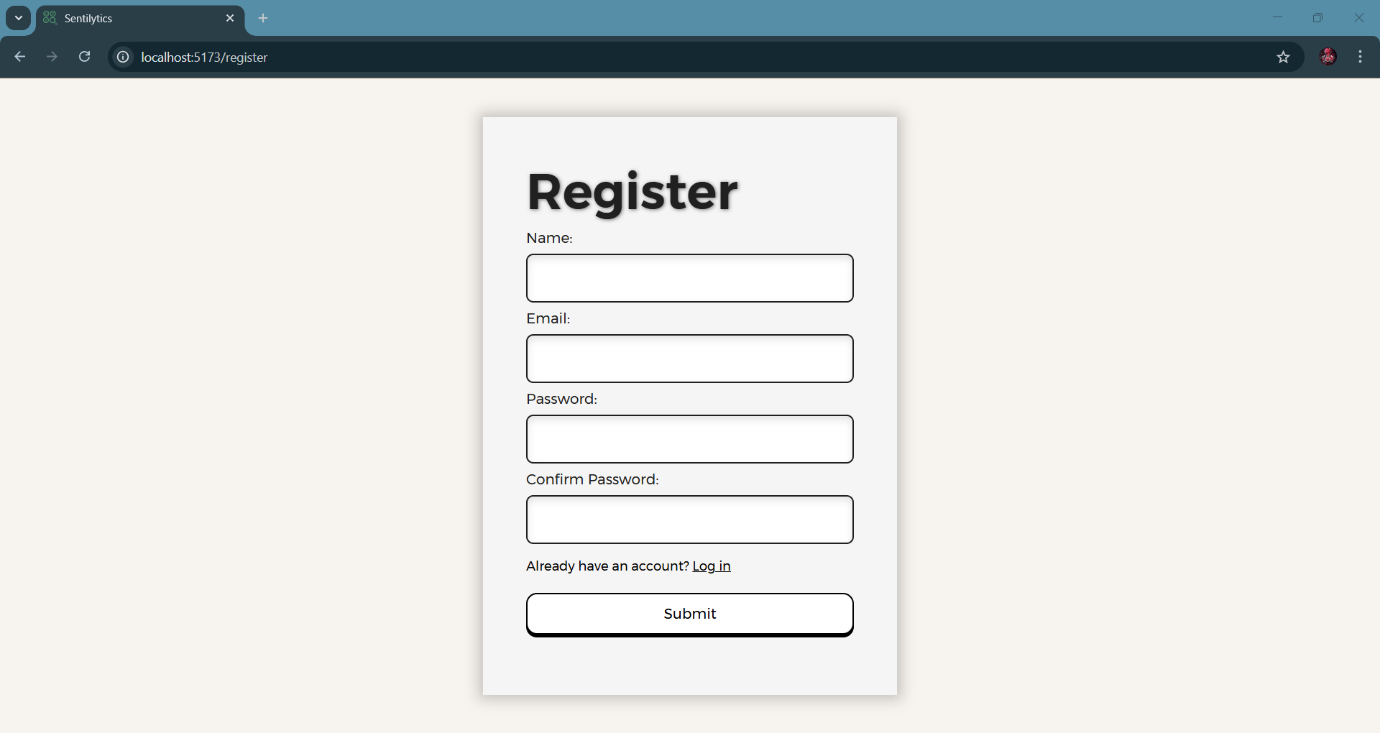
* 1. **CORRECTED SENTIMENT:**

**5.Screen Layouts:**

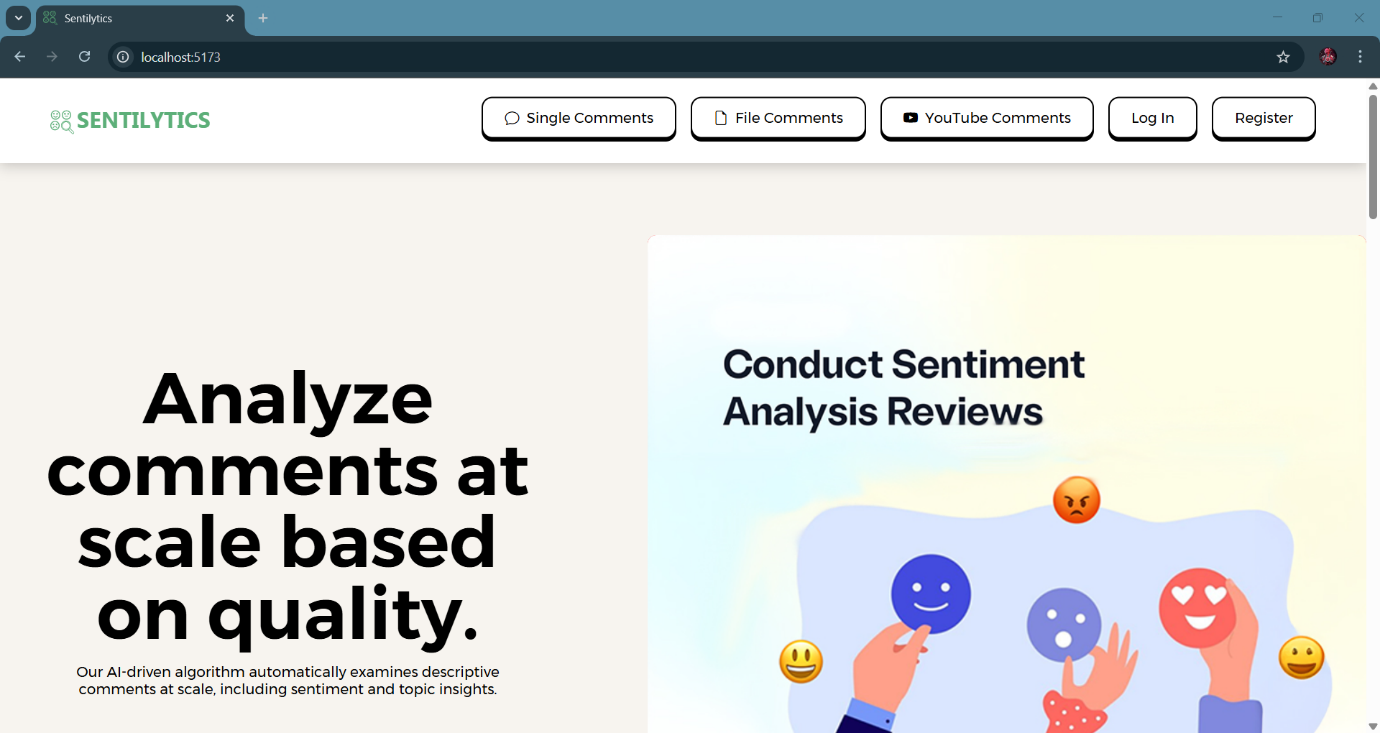
**Login Page:**

****

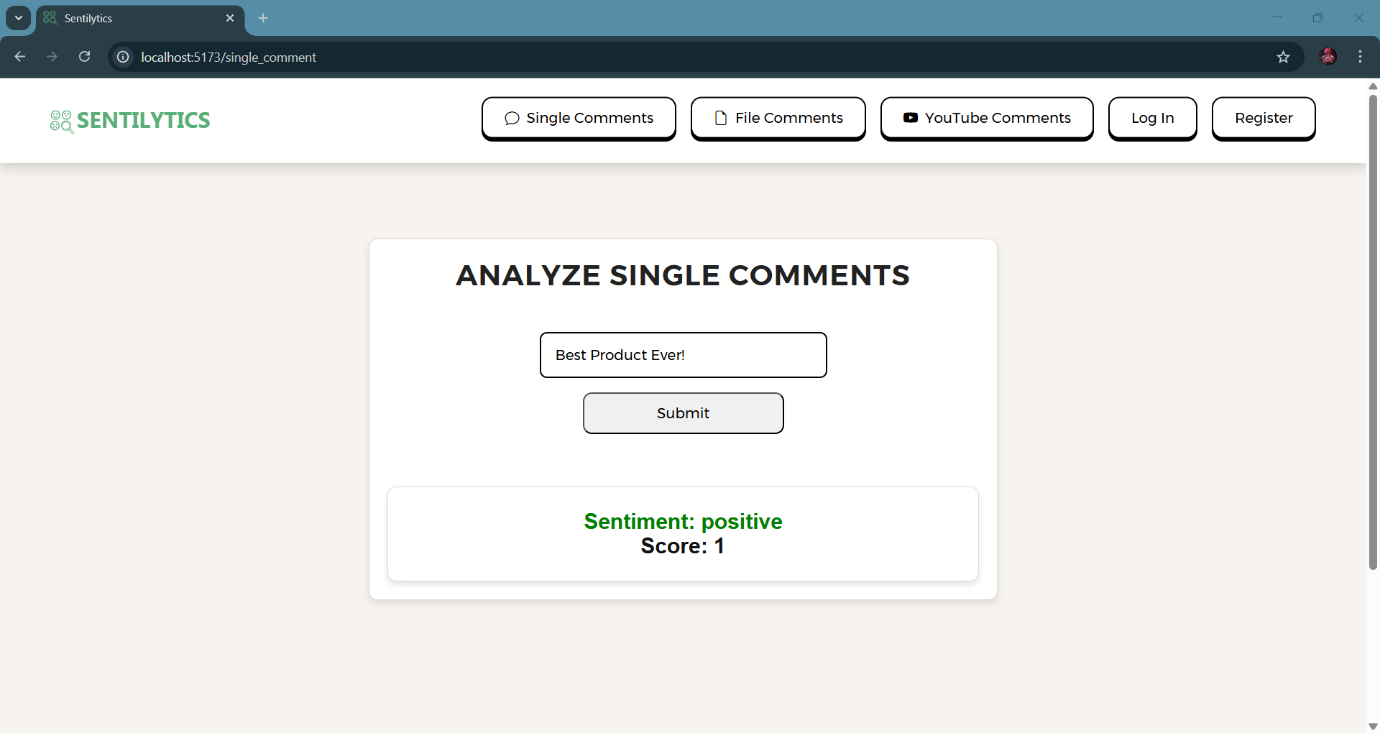
**Register Page:**

****

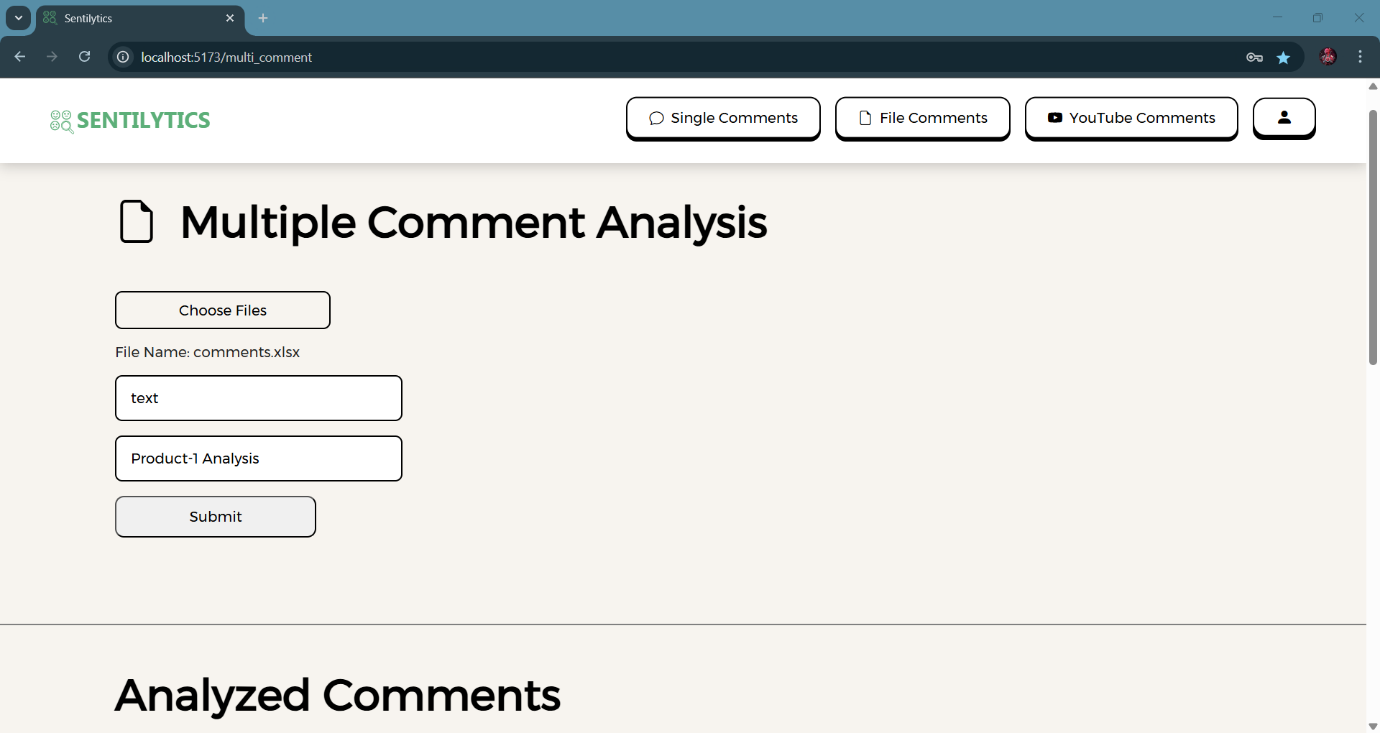
**Home Page: Single Comment Analysis**

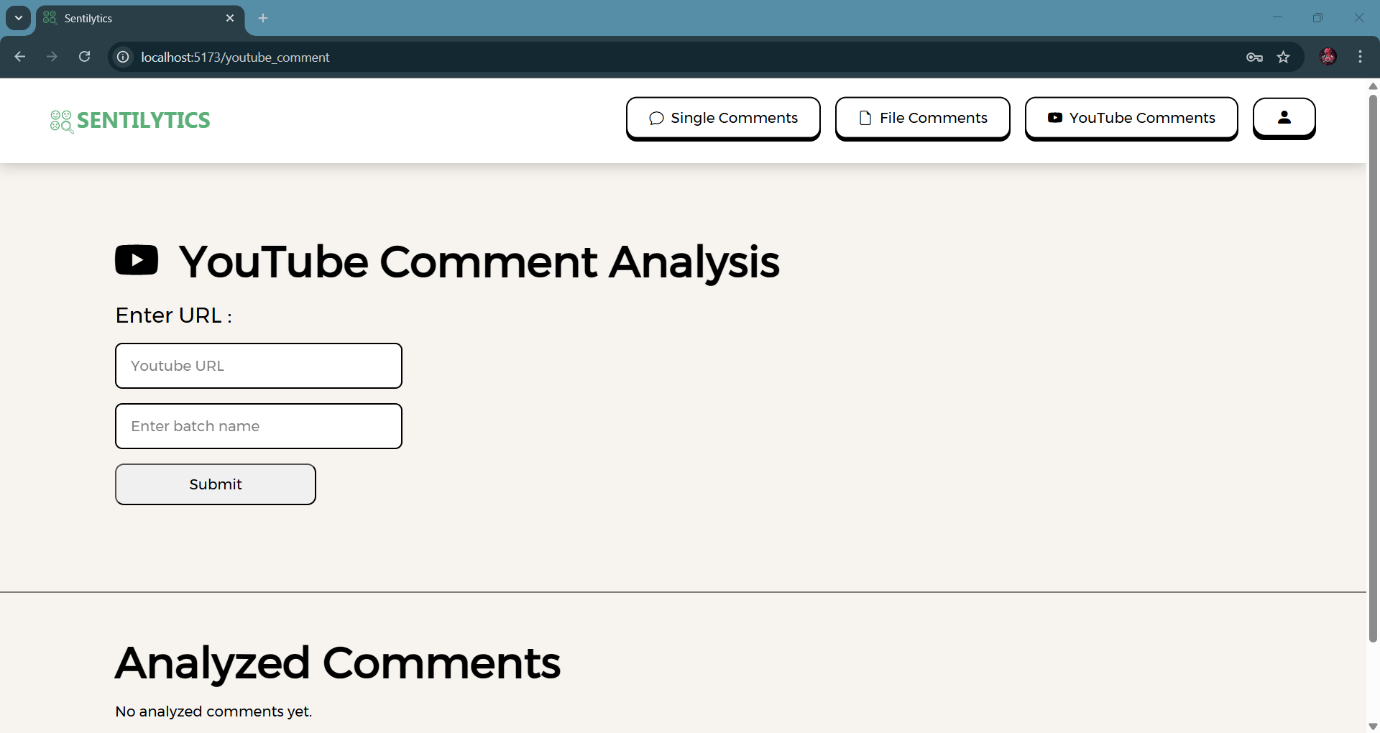
****

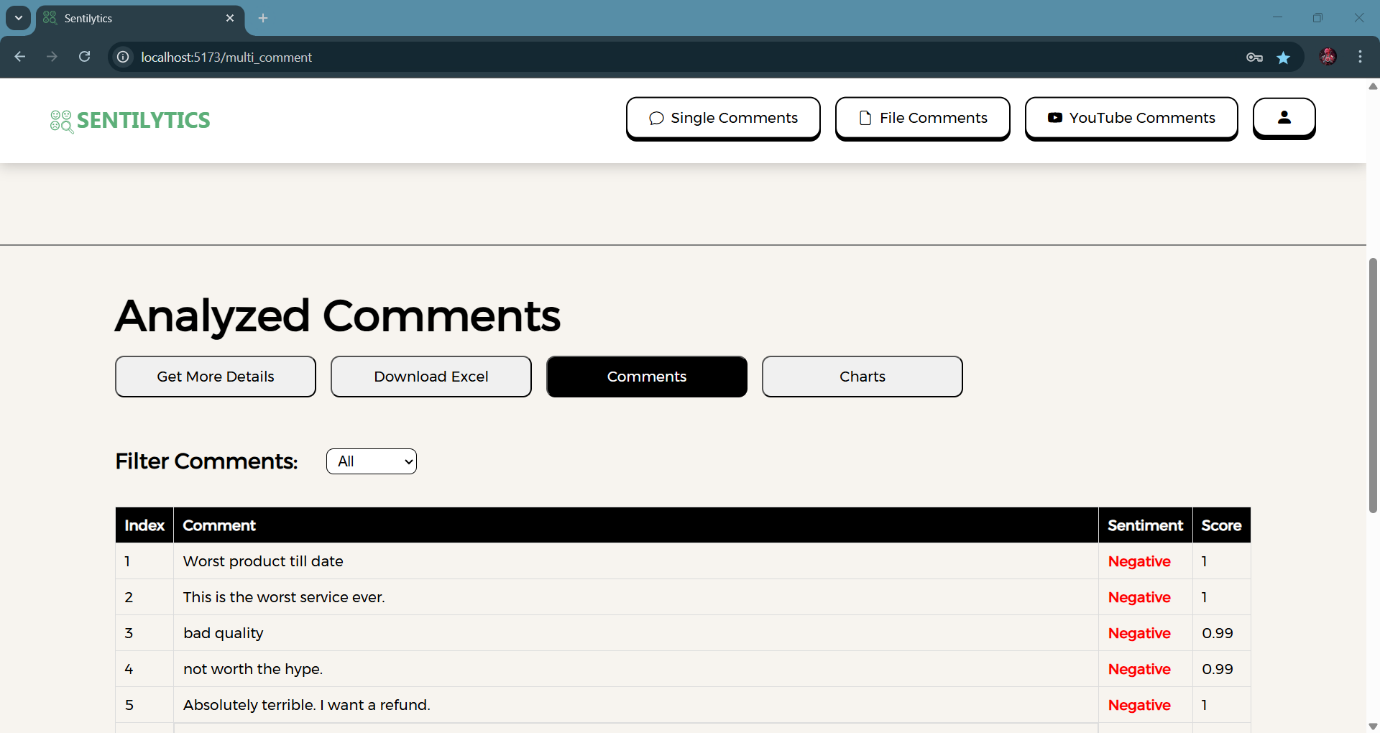
**Single Comment Analysis:**

****

**Multiple/Youtube Comment Analysis:**

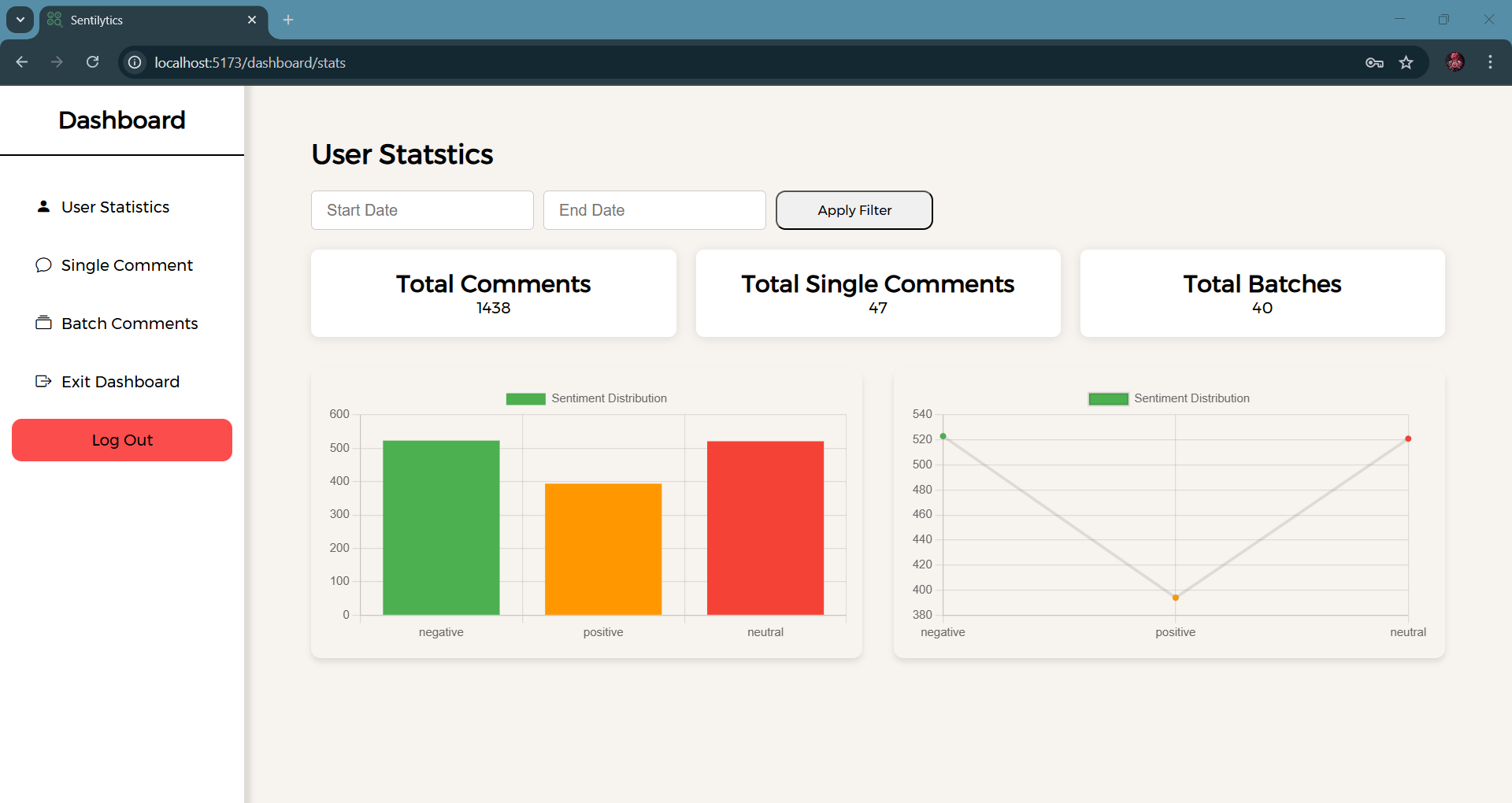
****

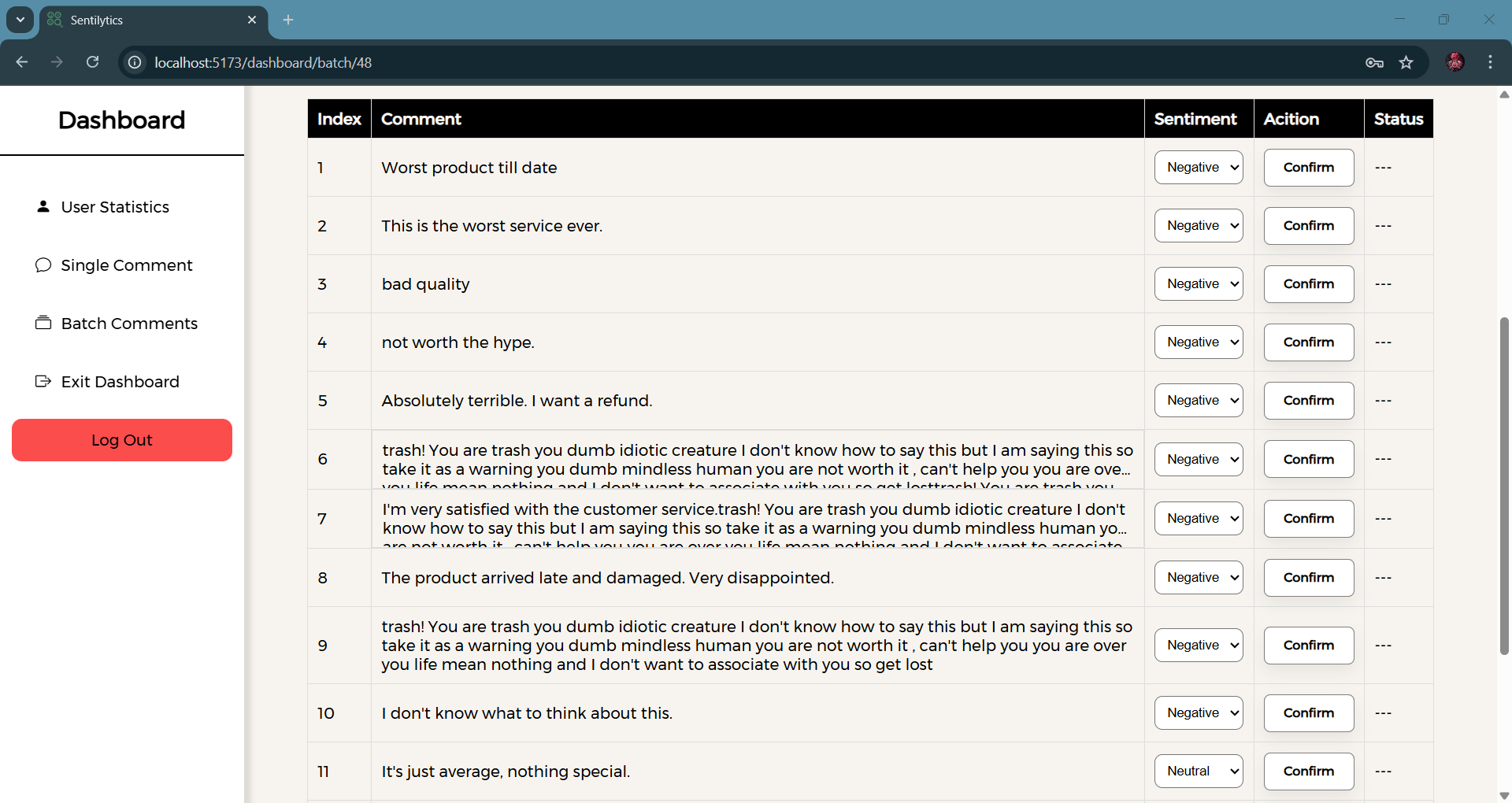
****

****

****

**Dashboard:**

****

****

**6.Conclusion:**

Sentilytics successfully leverages Logistic Regression with TF-IDF vectorization to provide a fast and accurate AI-powered sentiment analysis tool. The project integrates machine learning with an interactive UI to ensure efficient sentiment classification. Logistic Regression was chosen due to its interpretability, efficiency, and robustness for sentiment analysis tasks.

1. **References:**
   * + - [**https://docs.djangoproject.com/en/**](https://docs.djangoproject.com/en/)
       - [**https://youtu.be/j6szNSzw4BU?si=zlcct3gm2H39PEXt**](https://youtu.be/j6szNSzw4BU?si=zlcct3gm2H39PEXt)
       - [**https://youtu.be/\_nvQKN8L1ZE?si=xJRRWVg-ECPf6j79**](https://youtu.be/_nvQKN8L1ZE?si=xJRRWVg-ECPf6j79)
       - [**https://www.kaggle.com/datasets/kazanova/sentiment140**](https://www.kaggle.com/datasets/kazanova/sentiment140)
       - [**https://scikit-learn.org/stable/modules/generated/sklearn.linear\_model.LogisticRegression.html**](https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html)
       - [**https://react.dev/learn**](https://react.dev/learn)
       - [**https://youtu.be/4z9bvgTlxKw?si=LAr87T-412QDhVE-**](https://youtu.be/4z9bvgTlxKw?si=LAr87T-412QDhVE-)