YOUTUBE COMMENTS ANALYZER

PROJECT-ID-BSCS-F21-E-12

Project Advisor:	**Dr. Muhammad Akram Mujahid**
Program:	**BS Computer Science**
Shift:	**Evening**

Submitted By

SAQLAIN AZAM	BSF2100814
MUHAMMAD ZAIN	BSF2100809
ALI SHAN	BSF2101124

University of Education

YouTube Comments Analyzer

BS in Computer Science 2021-2025

A project submitted in partial fulfillment of the
requirement for the award of the degree of
BS in Computer Science

Department of Information Sciences,
Division of Science and Technology,
University of Education, Township, Lahore
May 2025

"We hereby declare that we have read this project documentation and in our opinion this project is sufficient in terms of scope and quality for the award of the degree of BS in Computer Science."

Project Primary Supervisor
Dr. Muhammad Akram Mujahid
Assistant Professor
Department of Information Science
Division of Science and Technology
University Of Education

Project Examiner
Name:
Designation:

DECLARATION

We declare that this project title entitled **"YouTube Comments Analyzer"** is the result of our own research and development except as cited in the references. This project has not been accepted for any degree and is not concurrently submitted in candidate for any other degree. At any time if our statement is found to be incorrect even afterwards of BS in Computer Science, the university has the right to withdraw our BS in Computer Science degree.

Signature: _________________

Name: SAQLAIN AZAM (BSF2100814)

Date:

Signature: _________________

Name: MUHAMMAD ZAIN (BSF2100809)

Date:

Signature: _________________

Name: ALI SHAN (BSF2101124)

Date:

PLAGIARISM UNDERTAKEN

We solemnly declare that project work presented in this documentation entitles **"YouTube Comments Analyzer"** is solely our work with no significant contribution from any other person. Small contribution/help wherever taken has been acknowledged and that complete project has been written by us.

We understand that zero tolerance policy of the HEC and University of Education, Lahore towards plagiarism. Therefore, we as the author of the above titled project declare that no portion of our project documentation and any material used as reference is properly referred/ cited.

We undertake that of we are found guilty of any formal plagiarism in the above titled project even after award of BS in Computer Science degree, the University reserve the rights to withdraw/revoke our BS in Computer Science degree and that HEC and the University has the right to publish our name on the HEC/University Website on which names of students are place who submitted plagiarized projects.

Signature: _________________

Name: SAQLAIN AZAM (BSF2100814)

Date:

Signature: _________________

Name: MUHAMMAD ZAIN (BSF2100809)

Date:

Signature: _________________

Name: ALI SHAN (BSF2101124)

Date:

ACKNOWLEDGEMENT

We would like to express our deepest gratitude to everyone who contributed to the successful completion of our project, **YouTube Comments Analyzer**. First and foremost, we are grateful to our supervisor **Dr. Muhammad Akram Mujahid, Assistant Professor of University of Education** for their invaluable guidance, encouragement, and insights throughout the development of this project. Their expertise helped us overcome challenges and refine the system to achieve its goals. We extend our sincere thanks to University of Education which provided the resources and platform necessary to carry out this work. We are also thankful to our family, friends, and peers for their unwavering support, constructive feedback, and motivation during the course of this project. Finally, we would like to acknowledge the online community and open-source contributors whose tools, libraries, and research served as a foundation for this project. This journey has been a significant learning experience, and we are truly grateful to all who have been a part of it.

1. SAQLAIN AZAM
2. MUHAMMAD ZAIN
3. ALI SHAN

Date:

May 14, 2025

ABSTRACT

The large volume of user-generated comments on YouTube presents the both easiness and challenges in understanding viewer's feedback and insights. This project, "YouTube Comments Analyzer", aims to streamline the analysis of YouTube video comments to generate understandable report and concise summary. This is a web-based platform where users can input YouTube video URL, then fetch comments from YouTube API and analyze comments using AI model. The website categorizes comments into positive, negative, suggestions, questions, and abusive category. It provides sentiment analysis, highlight abusive content, evaluates engagement metrics, and summarizes viewer sentiments or generate summary. The feedback are visually represented with the help of graphs and charts, making data understandable for user. Additionally, the system allows users to download the analysis report as PDF or Excel files for offline view. This project uses machine learning models to achieve accuracy of data and scalability. This website is helpful for the content creators, seeking to understand audience behavior and feedback.

List of Tables

List of Figures

Table of Contents

Chapter 1

Gathering and Analyzing Information

1.1 Introduction

In today's, YouTube stands as one of the largest source for content creation and viewers engagement. Daily millions of videos upload on YouTube, the comments section helps for viewers to express their opinions, share feedback, and engage in discussions. It is difficult for creator to read all comments. The "YouTube Comments Analyzer" project aims, using AI model categorize comments and generate report. By summarizing comments and categorizing them into positive, negative, suggestions and question, the tool will enable users to get report of comments quickly. The analyzer tool also implement filtering techniques to identify abusive language. After categorizing all comments, it generate is graph or chart and generate summary of that report.

1.2 Problem Description

YouTube is one of the greatest video-sharing source globally, and hosting millions of user-generated comments. These comments often replicate user feedback. However, analyzing these comments manually is a task difficult and time-taking process for content creators. Some issues are major: The large number of comments on video, impossible for content creators to read and extract meaningful insights manually. Some viewers write abusive comments, which have negatively impact the community and creators. Identifying and addressing such content is a critical challenge for content creator.

1.3 Aims and Objectives

The primary aim of the project is to automating the analysis of YouTube comments.

- Analyze positive, negative, suggestions, questions and abusive comments.
- Generate report in graphical form.
- Generate concise summary of data.

1.4 Research Questions

- How we can categorize comments in positive, negative, suggestions, questions and abusive comments?
- How we can make graphs and charts for display result in the form of pie chart and bar graph?
- How we can generate the summary of comments?
- How can AI model efficiently analyze YouTube comments for sentiments and abusive content detection?

1.5 Methodology

Data is fetch from YouTube API for training and testing. Preprocessing includes cleaning the text by deleting special characters. The system uses machine learning techniques for categorizing comments into positive, negative, abusive, suggestions, and questions. Results are represented with the help of bar chart and line graph.

1.5.1 Available Methodologies

Various methodologies are used for analyzing user-generated comments. Rule-based approaches depend on predefined keywords or patterns but unable to take the context of comments. Machine learning models like SVM and Naive Bayes Logistic Regression gives higher accuracy but we need well-labeled structured datasets for better performance. Deep learning methods, including LSTMs, RNNs, and transformers like BERT identifying complex language and description. The choice of method depends upon the scalability, accuracy and availability of the resources.

1.5.2 Chosen Methodology

This project will uses deep learning-based approach including transformer models. These models are best to do sentiment analysis and detecting abusive language with high accuracy. BERT do very well at understanding nuanced and descriptive language. When it combining with preprocessing and visualization tools, then system achieves effective and scalable performance. This methodology make sure that system meets its categorization objectives.

1.5.3 Reason for Chosen Methodology

Deep learning models rule-based and traditional machine learning methods are used for analyzing data. Rule-based approaches are limited to only context analyzing, while traditional models can works with large and complex datasets. Transformer models, such as: BERT give high accuracy, scalability, and ability to identify meanings of context. These all capabilities makes the deep learning as appropriate choice for analyzing data and moderating comments effectively.

1.6 Definitions, Acronyms and Abbreviations

1.6.1 Definitions

YouTube Comment Analyzer: A system which analyze, and categorize comments into predefined categories such as positive, negative, abusive, suggestions, and questions also summarizing and visual representation of the data.

Sentiment Analysis: The process of analyzing and categorizing the sentiment which expressed in text, typically as positive, negative, abusive etc.

Abusive Comments: User-generated comments containing offensive, hated speech or inappropriate language.

Visualization: The graphical representation of data with the help of graphs and charts to make understandable report.

1.6.2 Abbreviations

API: Application Programming Interface

ML: Machine Learning

DL: Deep Learning

RNN: Recurrent Neural Network

LSTM: Long Short-Term Memory

BERT: Bidirectional Encoder Representations from Transformers

1.6.3 Acronyms

API: A set of rules which are built for interacting with software applications.

ML: Algorithms and statistical models that enable the computers to perform tasks.

DL: A subset of ML that uses neural networks to analyze human decision-making.

BERT: A transformer-based model which is designed for natural language tasks.

Chapter 2

Software Requirement Specification

2.1 Stakeholders Characteristics

2.1.1 Primary Stakeholders

End Users:

- ***Description***: It includes individuals or organizations interest in analyzing comments of YouTube video. End users may be content creators, marketers, and researchers who wants to know the insights of audience engagement and check for sentiment and abusive contents.
- ***Needs and Expectations****:* Simple, user-friendly interface to enter YouTube video URLs and receive an analysis report with clear results like in the form of graphs. They expect accurate results which includes sentiment trends, vulgar or abusive comment identification, and percentages of likes and dislikes.
- ***Technical Knowledge****:* It doesn't require high profile technical knowledge, rather, general understanding of YouTube and basic analysis tools would be efficient.

Project Team:

- ***Description****:* Developing, testing, and deployment of website are managed by project team. In project team, AI specialists and a project manager will also be very important members.
- ***Needs and Expectations***: For the better development clear requirements and modular project components are very important. In order to meet user need's regular feedback from users will be very important.
- ***Technical Knowledge****:* For natural language processing and sentiment analysis, team has complete knowledge about software development, web development and AI or ML models.

Content Moderators:

- ***Description****:* It includes YouTube channel moderators or third-party moderators interested in viewing or examining abusive or vulgar comments on YouTube channels.
- ***Needs and Expectations****:* Precise and authentic identification of abusive comments with options for send out or dispatch reports to use against moderation and compliant tasks.
- ***Technical Knowledge****:* Some basic technical skills but most importantly focused on usability and interpretation of the analysis.

2.1.2 Secondary Stakeholders

YouTube Platform API Administrators

- ***Description****:* For retrieval of comments data, administrators maintain access to YouTube's public API.
- ***Needs and Expectations****:* Subordination with API usage policies, minimum server load and obedience to data privacy standards.
- ***Technical Knowledge****:* It needs High-level knowledge for implementation of API functionality and access protocols.

Evaluators

- ***Description****:* For academic merit, technical depth and innovation, it has faculty members or project evaluators who will assess the project.
- ***Needs and Expectations***: Detailed documentation is very important, code should be clear, SRS should be clear-cut and obvious and loyalty to project timelines.
- ***Technical Knowledge****:* In AI-based sentiment analysis tools, it has understanding of project management, documentation and the technical components

2.1.3 External Stakeholders

Future Clients

- ***Description****:* For commercial use, businesses, social media monitoring agencies and third-party developers may adopt or expand this project
- ***Needs and Expectations****:* Integrate this tool with other social media (like TikTok, Facebook, Instagram etc) platform for adaptable and scalable infrastructure with vast social media monitoring standards.
- ***Technical Knowledge****:* End-users may have less technical expertise while developers has in-depth knowledge for expanding application's functionalities.

2.2 Domain Requirement

The YouTube Comments Analyzer project works within the domain of sentiment analysis, natural language processing (NLP) and user interaction analysis on YouTube. The primary focus of this tool is to allow users (content creators) to understand audience reactions through automated analysis comments. Important aspects and scope of this project is given below:

2.2.1 YouTube Comments Data

- YouTube comments are handled by this tool which contains textual content.
- To fetch the comments, this tool will use YouTube API. Different limits are applied to the data retrieve based on YouTube's policies. For example, API rate limits and access constraints to different content type.

2.2.2 Sentiment and Abusive Language Analysis

- The analysis of sentiment in the domain includes categorizing comments into positive, negative, suggestive and questioning types. Context understanding, meaning, and emotional tone of language are required by the domain of sentiment analysis.

- Specific keywords, context, and sentence structure are analyzed by abusive language detection which is important sub-domain.

2.2.3 Audience Engagement Metrics

- This tool will also focus on quantifying viewer engagement by calculating percentages of positive, negative, abusive, suggestive and questioning reactions. It

will also give the percentage of likes and dislikes or positive vs. negative comments, because these are important factors for content creators and marketers.
- The domain emphasize accuracy in data interpretation because these metrics influence user perceptions and content decisions.

2.2.4 Report Generation and Visualization

- This tool will transform raw data into visual representation, in the form of graph and chart. These visual reports will help us for quick and clear understanding of audience sentiment and abusive content.
- The domain requires that generated reports can be easily understandable and readable by non-technical users, because it will user-friendly design.

2.2.5 Data Privacy and Compliance Constraints

- There are certain privacy policies and compliance standards to follow during the working with YouTube API domain. Sensitive or personal information should not be stored by the users, and all API interactions must respect YouTube’s terms and conditions.
- The domain emphasizes secure handling of retrieved data and protect user data and maintain privacy standards.

2.3 Functional Requirements

No	Requirement	Description
FR1	Registration	The system must allow only new users to register for an account. The registration page should include fields for: Full Name, Email, Phone number, Password, Confirm Password. The system should validate the inputs such as ensuring the email format is correct, confirming that passwords match, verifying that the email is already registered or not. Upon successful registration, the user should receive a confirmation message “Successfully Registered”. The user's details should be securely stored in the system's database.

FR2	Login	The system allows the users to login to their accounts. It also provide a password recovery option if user forget their credentials like username or password. The login page must include fields for: Email Address, Password. The system should validate the credentials: If credentials are valid, then user granted access to the system. If invalid, then system displays an error message: "Invalid email or password". The login page must include a "Forgot Password" link. When the user clicks on “Forgot Password” button, then system will take registered email address. The system validates the email address, If email is already exists in database, The system sends a password reset link to the user's email. The reset link should allow the user to enter a new password after verifying. The system validates the new password is meets security requirements (e.g., length, special characters). Confirms both password fields match, the password is updated in the database and the system notifies the user: "Password successfully reset." If the email not exist in database, then system displays an error message "No account found". After reset the password, the user move to the login page to login with entering new password. The reset link must expire after a specified time due security purposes.
FR3	YouTube API Integration	The system must connect with the YouTube’s API to retrieve comments from provided YouTube video URL. The format to get data should be in this format: <ul><li>Comment Text: Main comment.</li><li>Like Count: Number of likes on comment. </li><li> Timestamp: Time and date of comment posting.</li></ul>. The system must notify clear error messages to users when invalid URL provide by the user.
FR4	Sentiment Analysis of Comments	The system must analyze each comment then classify all comments as positive, negative, suggestion, and questions. If comments are not retrievable due to any reason then system must notify with proper error message.
FR5	Abusive Language Detection	The system identify comment which contain abusive or offensive language. Detected abusive comments will be highlighted in the report, distinguishing them from other comments. The model must be accurately detect offensive language.
FR6	Engagement Metrics Calculation	The system calculate the percentage of positive, negative, abusive suggestion, and questions. It should determine and display the percentages of positive comments vs negative comments, suggestion, and questions percentages.

FR7	Report Generation with Visualizations	The system should generate a visual report for users, containing: <ul><li>Sentiment Graph: Pie chart and bar graph showing percentages of positive, negative, abusive, suggestion, and questions comments. All matrix show in different way like graphs, chart, and in text form.
FR8	Export Report Functionality	Users should have the option of download the report in the PDF format to keep record for user. Exported files must have same headings and labels after download.

Table 2.1: Functional Requirements

2.4 Non-Functional Requirements

No	**Requirement**	**Description**
NFR1	Performance	The system should be able to analyze comments quickly. Quick response times are ensured and optimized to get comments from API calls.
NFR2	Scalability	The system should be able to handle increasing loads, when a large number of users comes then system should work smoothly, without drop performance.
NFR3	Accuracy	The sentiment analysis and abusive language detection algorithms must be work accurately and maintain the accuracy rate of 85%. To increase the accuracy of system, the system should in periodic testing and updates.
NFR4	Usability	The user interface must be easy to understand for non-technical user. Visualization should be clear, understandable and easy to use for everyone.
NFR5	Security and Privacy	Reports and user data must be safe from unauthorized access. The system should link with YouTube's data term and conditions, confirming no sensitive data stored outside the analysis duration.
NFR6	Reliability and Availability	The system should available 24/7 and provide consistent access to the application and report generation. When error occurs then system gracefully notify to users and, if possible, then automatically retry the request.
NFR7	Maintainability	Development should be in a version control system, if update needed then it must be easy.

NFR8	Compatibility	The application should be compatible for the all major web browsers (e.g. Safari, MicroSoft Edge, Chrome, Firefox). Reports should be exportable in different format like PDF and Word.
NFR9	Language Support	The system should support English languages to accommodate YouTube's global user base.
NFR10	Data Integrity	The system must ensure that the retrieved comments remain accurate throughout the analysis and report generation. Unauthorized access and data loss during analysis must be prevented.

Table 2.2: Non- Functional Requirements

Chapter 3

Analysis [Use Case Description and Use Case Model]

3.1 Use Cases Description

3.1.1 UC Number: 1.1

UC Name: Register

Functional Requirement No: FR1

Primary Actors/Stakeholders: User (YouTube Viewer)

Secondary Actors/Stakeholders: Admin

Description:

New users must register to access the system by providing their credentials, which are stored securely in the system's database.

Preconditions:

1. The user must not have an account before.
2. The system must be fully operational.

Main Success Scenario (MSS):

1. The user navigates to the registration page.
2. The system displays a registration form asking:

    a) Full Name
    b) Email Address
    c) Password
    d) Confirm Password

3. The user enters the details and submits the form.
4. The system evaluate the information and checks if the email is already in use.
5. The system creates a new account and securely, stores the user's information.
6. The system notifies the user of successful registration and redirects them to the login page.

Alternative Scenarios:

1. ***Email already registered:***

    The system informs the user that this email is already registered and suggests logging in instead.

2. ***Validation errors:***

    The system highlights issues (e.g., invalid email format, mismatched passwords) and invoke the user to correct them.

Postconditions:

1. A new account is created and information is stored in the database.
2. The user can log in by using the registered credentials.

Extensions:

1. ***Invalid Email Format:***

   The system ask the user to enter a valid email address.

2. ***Weak Password:***

   The system suggests a stronger password that meets the security credentials (e.g., minimum length, special characters).

3.1.2 UC Number: 1.2

UC Name: Login

Functional Requirement No: FR2

Primary Actors/Stakeholders: User (YouTube Viewer)

Secondary Actors/Stakeholders: Admin

Description:

The users must authenticate themselves by logging in using their email and password in order to access the system. If the user forgets their password, they can use the "Forgot Password" method to reset it.

Preconditions:

1. The user must have a registered account.
2. The system must be operational.

Main Success Scenario (MSS):

1. The user navigates to the login page on the system.
2. The user enters their email and password.
3. The system validates the credentials.
4. The system authenticates the user and grants access to the dashboard.
5. The user can now have access to the main features of the system.

Alternative Scenarios:

1. Invalid credentials:

The system displays an error message, "Invalid email or password," and provoke the user to try again.

2. ***Forgot Password:***

a) The user clicks on the "Forgot Password" link.
b) The system urge the user to enter their registered email address.
c) The system sends a password reset link to the user's email.
d) The user clicks the link, enters a new password, and confirms it.
e) The system validates the new password and updates it in the database.
f) The user is redirected to the login page with a confirmation message: "Password successfully reset. Please log in."

Postconditions:

1. The user is successfully logged in and gains system access.
2. Password reset is successfully completed if the "Forgot Password" option is used.

Extensions:

1. ***Incorrect Password:***

a) The system invoke the user to re-enter the password.
b) If the re-entered password is incorrect, the system suggests using the "Forgot Password" feature.

2. ***Forgot Password Email Not Found:***

If the entered email does not exist in the database, the system displays an error message: "No account found with this email address."

3.1.3 UC Number: 1.3

UC Name: Fetch Comments from YouTube
Functional Requirement No: FR3
Primary Actors/Stakeholders: User
Secondary Actors/Stakeholders: YouTube API

Description: The user enter a YouTube video link, then the system retrieves comments related to the video with the help of YouTube API.

Preconditions:

1. The user must enter a valid YouTube video link.
2. The system must has access to the YouTube API (Application Program Interface).

Main Success Scenario (MSS):

1. User gives the YouTube video link.

2. User clicks on the “Analyze” button.
3. System validates the link format.
4. System sends request to YouTube API to retrieve comments.
5. YouTube API send back comments data.
6. For analysis, system stores the comments for short time.

Alternative Scenario:

1. If user enter invalid URL then system gives error message “Enter a valid URL”.

Postconditions:

1. For analysis, comments are successfully retrieved and stored.

Extensions:

1. If API not retrieve data then system displays an error message “Please try again later”.

3.1.4 UC Number: 1.4

UC Name: Analyze Comments Sentiment
Functional Requirement No: FR4
Primary Actors/Stakeholders: System

Description: The system performs sentiment analysis on all comments, then categorize into positive, negative, suggestion and questions.

Preconditions:

1. Comments have been successfully retrieved from YouTube.

Main Success Scenario (MSS):

1. System takes stored comments.
2. System perform operation on all comment one by one through a sentiment analysis model.
3. Each comment is categorized as positive, negative, suggestion and questions.
4. Results will store for report generation.

Alternative Scenario:

1. If comments are not retrived then display error message to user. ”Comments not retrived”.

Postconditions:

1. Sentiment classifications of all comments are stored for reporting generation.

Extensions:

1. System neglect the troublesome comment and move to next.

3.1.5 UC Number: 1.5

UC Name: Detect Abusive Language in Comments
Functional Requirement No: FR5
Primary Actors/Stakeholders: System

Description: The system analyze and highlight any abusive language in comments.

Preconditions:

1. Comments have been successfully retrieved from Youtube API.
2. Sentiment analysis has been done.

Main Success Scenario (MSS):

1. System scans each comment one by one for abusive language detection using predefined rules.
2. Abusive comments are highlight for reporting.
3. Highlighted information is stored for all abusive comments.

Alternative Scenario:

1. None.

Postconditions:

1. Abusive comments are highlight for add in the report.

3.1.6 UC Number: 1.6

UC Name: Calculate Engagement Metrics
Functional Requirement No: FR6
Primary Actors/Stakeholders: System

Description: The system calculates engagement metrics, including the proportion of positive, negative, abusive, questions, and suggestion.

Preconditions:

1. Sentiment analysis and abusive language detection are done.

Main Success Scenario (MSS):

1. System calculates the percentage of positive, negative, abusive, questions, and suggestions.
2. The system stores engagement metrics for report generation.

Alternative Scenario:

1. None.

Postconditions:

1. Engagement metrics are stored for report generation.

Extensions:

1. None.

3.1.7 UC Number: 1.7

UC Name: Generate Report with Visualizations
Functional Requirement No: FR7
Primary Actors/Stakeholders: User
Secondary Actors/Stakeholders: System

Description: The system generates a report with visualizations based on the analysis results, including sentiment distribution, abusive comments ratio, questions and suggestion, with the help of engagement metrics.

Preconditions:

1. Comment analysis (positive, negative, abusive language, questions and suggestion) is complete.

Main Success Scenario (MSS):

1. System generates visualizations like graphs and charts for positive, negative, question, suggestion and abusive comment.
2. System displays visual report to the user.

Alternative Scenario:

1. None.

Postconditions:

1. The visual report is accessible for the user on interface.

3.1.8 UC Number: 1.8

UC Name: Export Report
Functional Requirement No: FR8
Primary Actors/Stakeholders: User
Secondary Actors/Stakeholders: System

Description: The user exports the generated report as a PDF or Word file for offline view.

Preconditions:

1. The visual report is completely generated.

Main Success Scenario (MSS):

1. User selects the export option (PDF or Word).
2. System gives access to the user for download the file.

Alternative Scenario:

1. Some time internet problem occurs, then system notifies to the user and suggests try again for export.

Postconditions:

1. The report is successfully downloaded by the user.

3.2 Use Case Model

Registration:

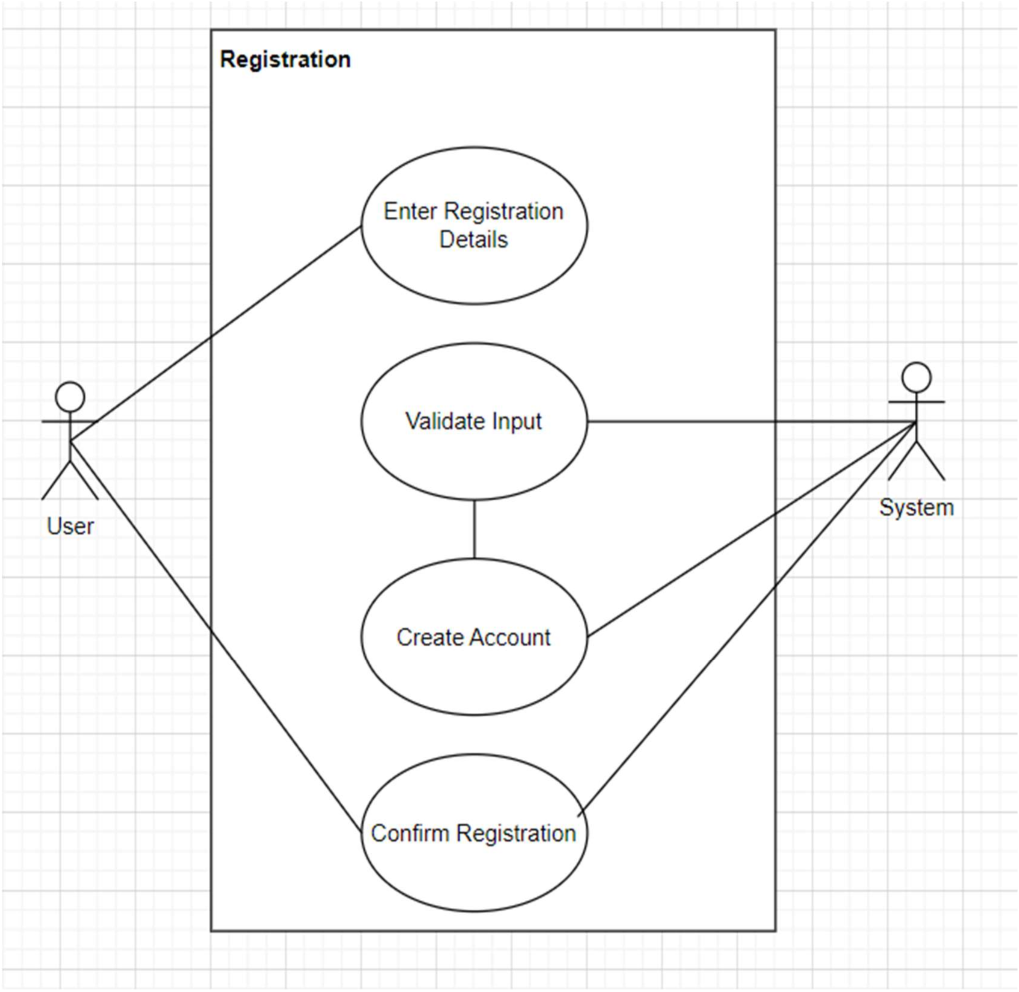


Figure 3.1: Registration

Login:

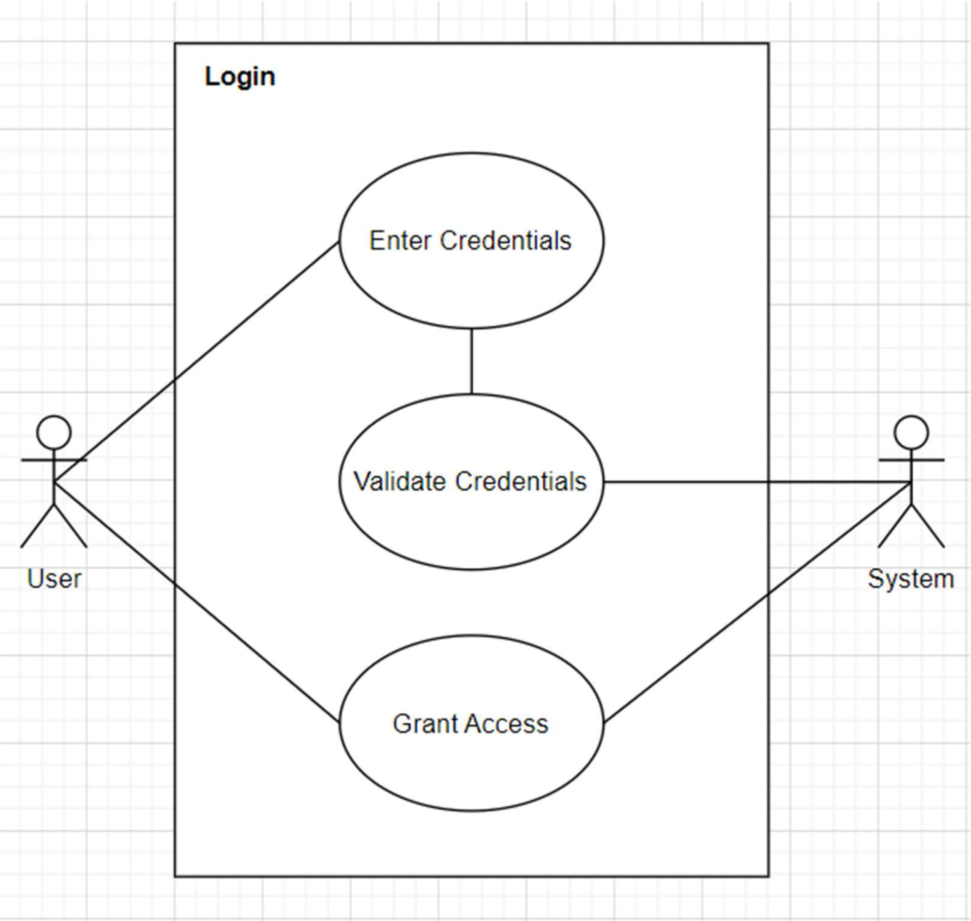


Figure 3.2: Login

Fetch Comments From Youtube:

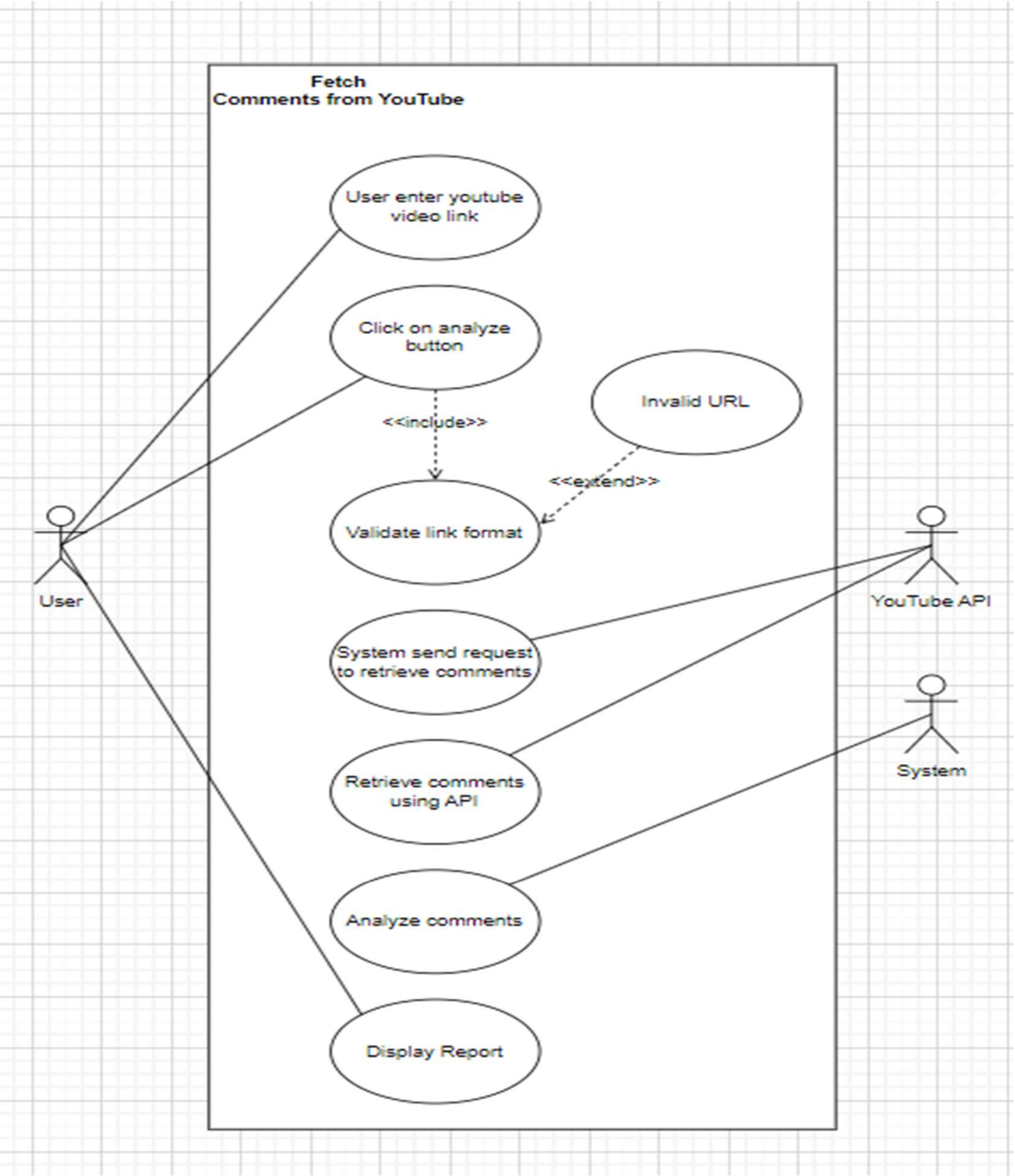


Figure 3.3: Fetch Comments From Youtube

Export Report:

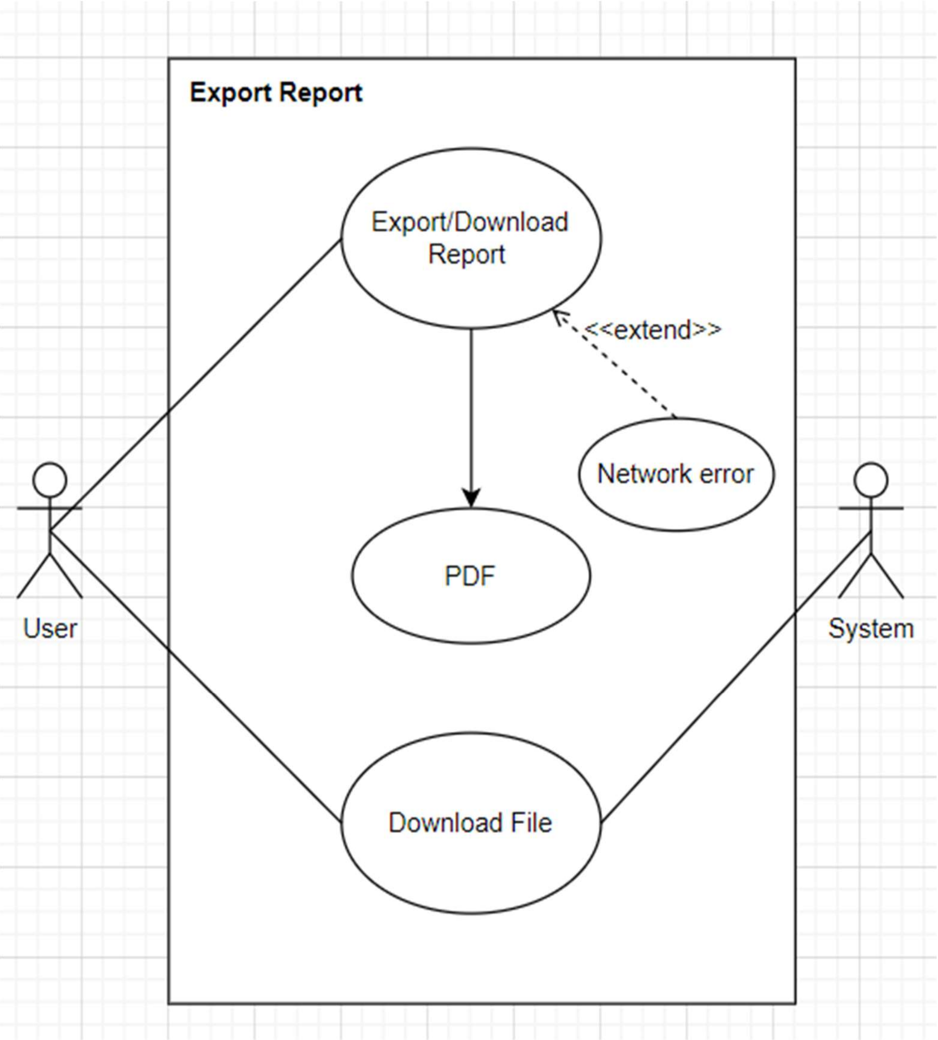


Figure 3.4: Export Report

Chapter 4

Design [with Description of each diagram]

4.1 Architecture Diagram

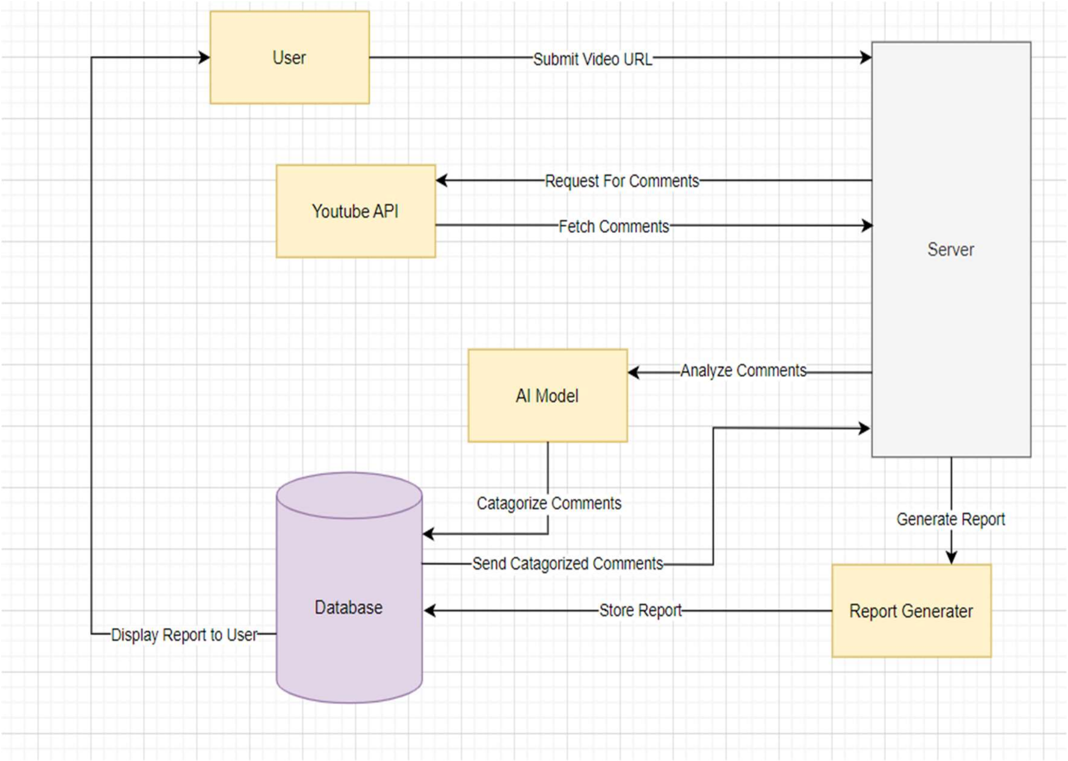


Figure 4.1: Architecture Diagram

4.1.1 Description

This architecture diagram demonstrates the workflow and components of the **YouTube Comments Analyzer** system, breaking down the whole interaction procedure into various parts:

1. **User**:
   - The user submits the URL of the YouTube to the system through the user interface.
2. **Server**:
   - The server handles requests received from the user and acts as a hub (central communication point) for all components.
   - It sends the URL of the video to the **YouTube API**, which then fetch the comments.
3. **YouTube API**:
   - The YouTube API is responsible for retrieving the comments associated with the given URL of the video and sends the data back to the server.
4. **Database**:
   - Once the process completed, categorized comments and reports are also stored here for reference.
5. **AI Model**:
   - The AI model analyzes the comments which are fetched from the database.

- It categorizes the comments into groups such as positive, negative, suggestions, and questions.
- The comments are sent back to the database for storage after categorization.

6. **Report Generator**:

- The report generator is responsible for creating a comprehensive report, including visualizations and statistics after processing the categorized comments.
- This report is stored in the database.

7. **Display to User**:

- The system displays the generated report to the user, providing comprehensive details of the analysis of YouTube comments.

4.2 ERD

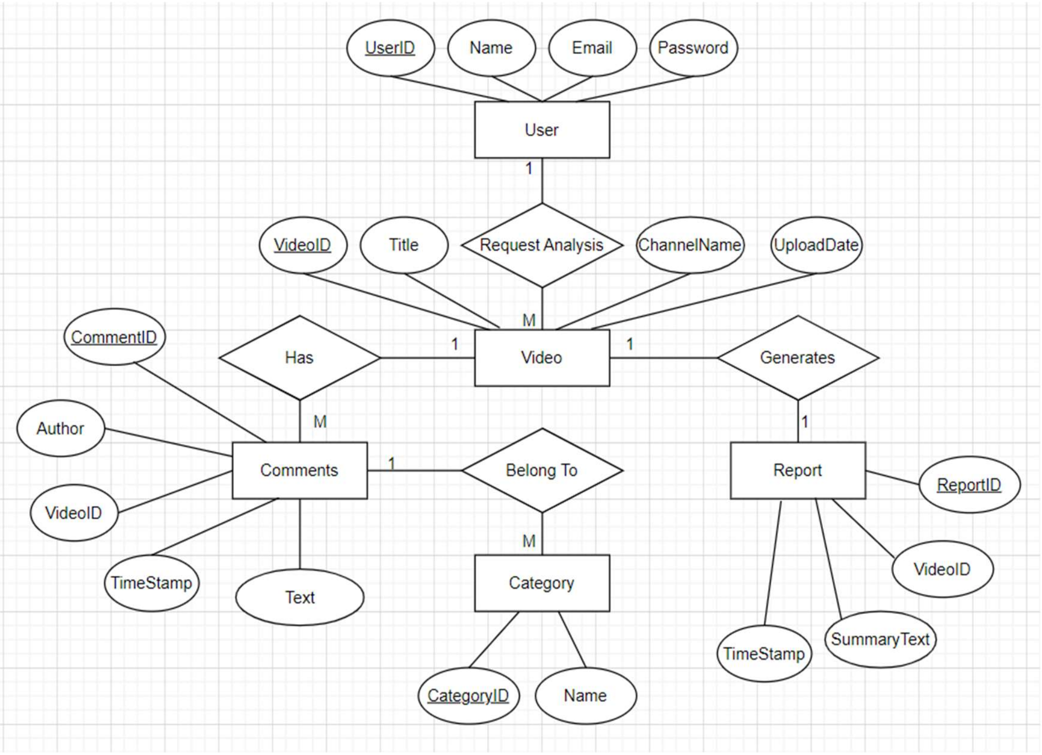


Figure 4.2: ERD

4.2.1 Description

1. **User**:
   - ***Attributes**:*
     - UserID: UUID (Primary Key): Unique identifier of a user.
     - Name: String: User's name.
     - Email: String: Email address of the user.
     - Password: String: User's password for authentication.
   - ***Relationships**:*
     - A user can upload as many videos as he can.
2. **Video**:
   - ***Attributes**:*
     - VideoID: UUID (Primary Key): Unique identifier of a video.
     - TitleName: String: Title of the video.
     - ChannelName: String: Channel's name that uploaded the video.
     - UploadDate: Date: The Date on which the video was uploaded.
   - ***Relationships**:*
     - Each video belongs to a specific user.
     - A video can have multiple comments.
     - Each video can be associated with one or more categories.
     - A video can have reports.

3. **Comments**:
   - ***Attributes***:
     - CommentsID: UUID (Primary Key): Unique identifier for a comment.
     - Text: String: Content of the comment.
     - Author: String: Author of the comment.
     - TimeStamp: DateTime: Date and time when the comment was posted.
   - ***Relationships***:
     - Comments are associated with a specific video.
4. **Report**:
   - ***Attributes***:
     - ReportID: UUID (Primary Key): Unique identifier for a report.
     - TimeStamp: DateTime (Primary Key) Date and time of the report.
     - SummaryText: String: Summary or description of the report.
     -
   - ***Relationships***:
     - Each report is linked to a specific video.
5. **Category**:
   - ***Attributes***:
     - CategoryID: UUID (Primary Key): Unique identifier for a category.
     - Name: String: Name of the category.
     - Description: String: Description of the category.
   - ***Relationships***:
     - A video can belong to multiple categories and eac category can include multiple videos (many-to-many relationship).

Relationships

- ***User and Video***: One-to-Many
  - A user can upload multiple videos, but each video is uploaded by one user.
- ***Video and Comments***: One-to-Many
  - A video can have multiple comments, but each comment is associated with a single video.
- ***Video and Report***: One-to-Many
  - A video can have multiple reports, but each report belongs to one video.
- ***Video and Category***: Many-to-Many
  - A video can belong to multiple categories, and a category can include multiple videos.

4.3 Data Flow diagram (Level 0 and 1)

(Level 0)

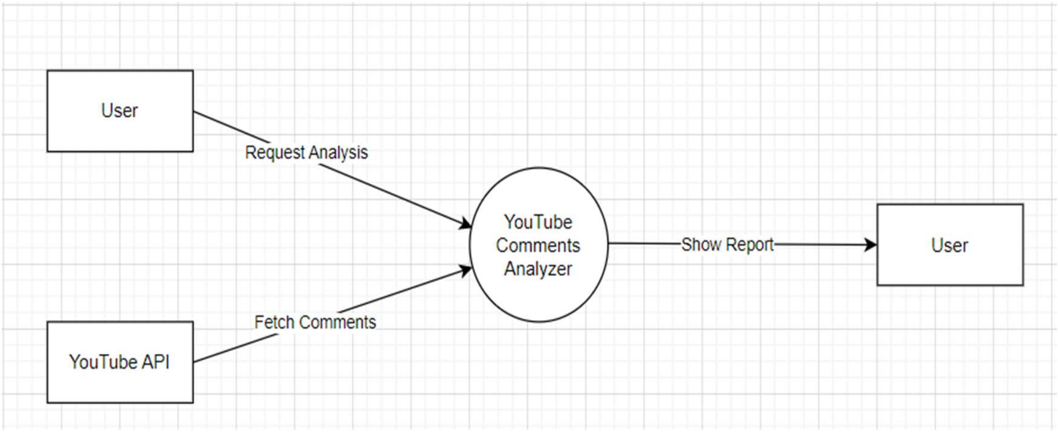


Figure 4.3: Data Flow Diagram (Level 0)

4.3.1 Description

1. ***User**:*

- User initiates a request for analyzing YouTube comments by interacting with the system.

2. ***YouTube API**:*

- API fetches comments from YouTube video based on the requested data.

3. ***YouTube Comments Analyzer**:*

- Comments analyzer acts as the central system, processing the fetched comments and performing analysis.
- It generates a report based on the analysis.

4. ***Output to User**:*

- The processed report is displayed back to the user.

***Workflow**:*

- The user sends an "Analysis Request" to the YouTube Comments Analyzer.
- The analyzer fetches data (comments) from the YouTube API.
- After processing the comments, the analyzer generates a report, which is shown back to the user.

(Level 1)

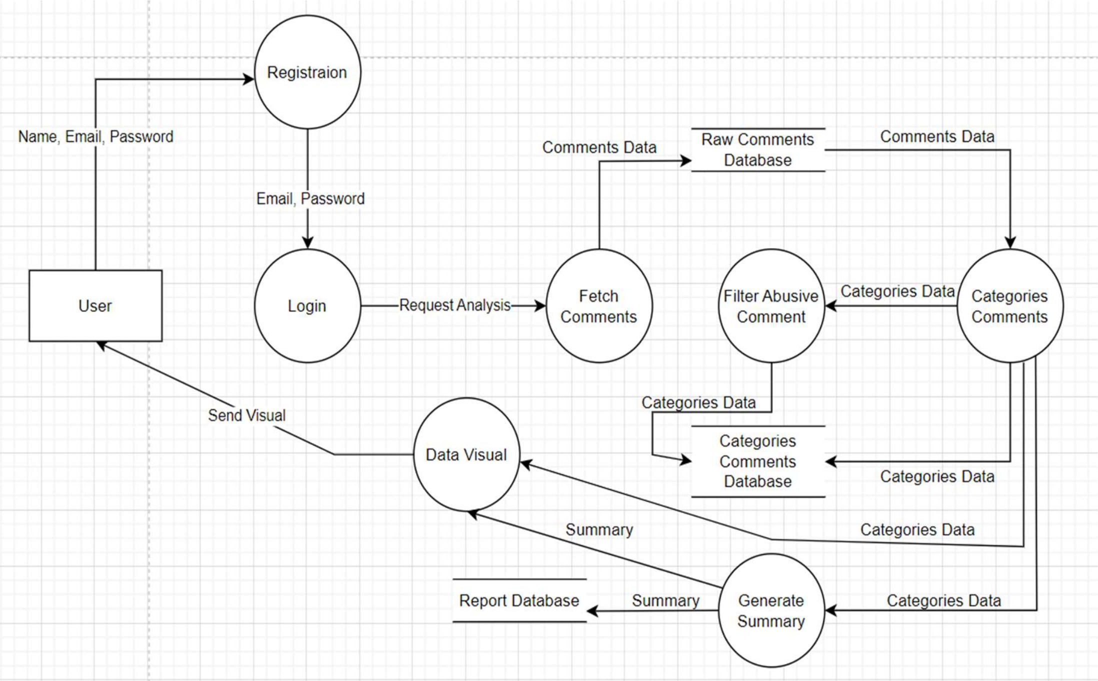


Figure 4.4: Data Flow Diagram (Level 1)

4.3.2 Description

1. ***User***:
- User interacts with the system to register, log in, and request comment analysis.
- Visual reports are received after analysis.

2. ***Registration***:
- It collects user details (Name, Email, Password) for account creation.

3. ***Login***:
- It authenticates the users with their credentials.

4. ***Fetch Comments***:
- Fetch comments data from external sources or APIs.
- It stores raw comments in the **Raw Comments Database**.

5. ***Filter Abusive Comment***:
- Analyzes comments to detect and flag abusive content.
- Updates filtered data in the **Comments Database**.

6. ***Categories Comments***:
- It categorizes comments based on predefined criteria.
- It stores categorized comments in the **Categories Comments Database**.

7. ***Data Visual***:
- Display visual representations of the analyzed and categorized data.
- Sends visuals to the user for deep understanding.

***8. Generate Summary**:*

- It summarizes the analysis results.
- Stores reports (results) in the **Report Database**.

Workflow:

- The **User** registers first and then logs into the system using their credentials.
- The user requests an analysis, invoking the **Fetch Comments** module.
- Raw comments are analyzed:
  - **Abusive Comment Filtering** removes harmful or abusive content.
  - **Categorization** organizes comments into groups based on specific criteria.
- The processed result are stored in the database and passed to the **Data Visual** module.
- Visual representations and summaries are generated and displayed to the user.
- Summaries are stored in the **Report Database** for future reference

4.4 Class Diagram

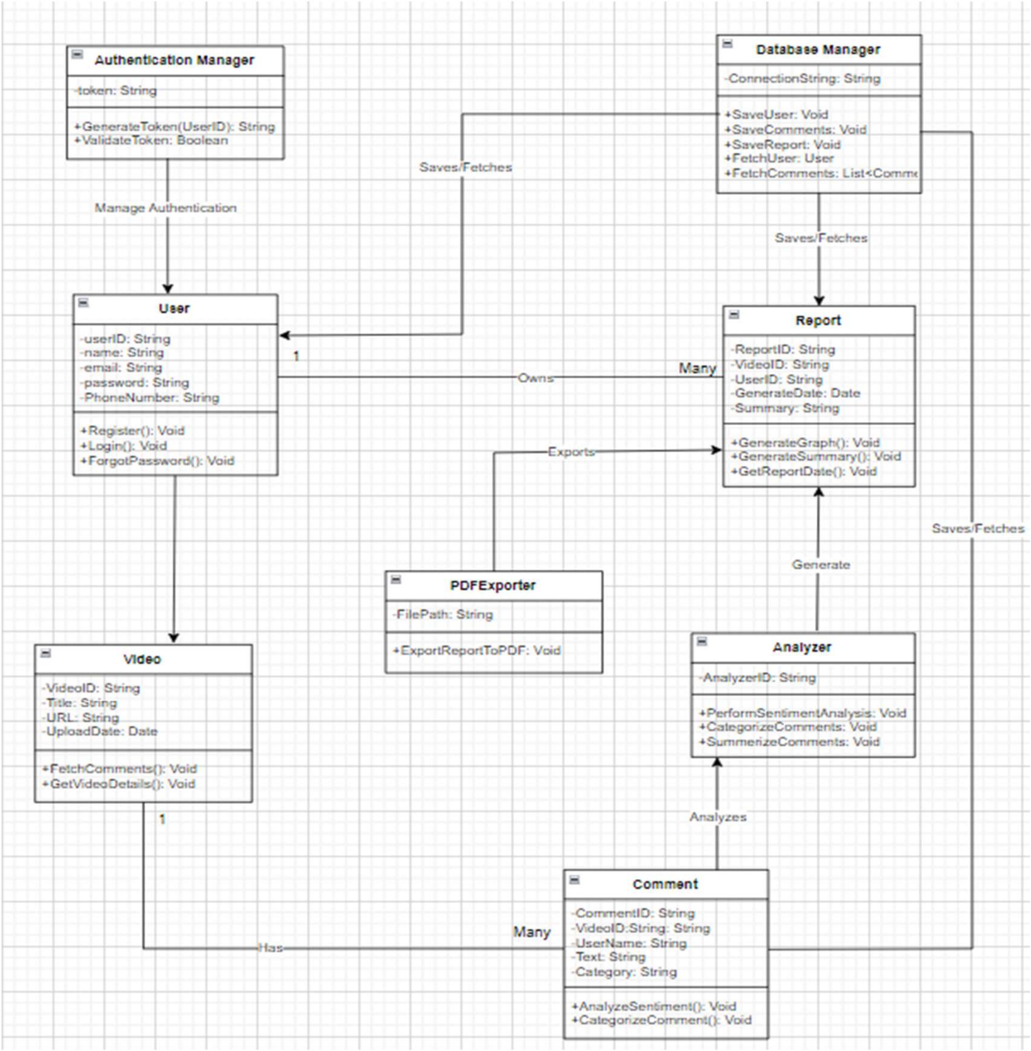


Figure 4.5: Class Diagram

4.4.1 Description

4.4.2 *User Class*

- ***Attributes:***
  userID: Unique identifier for the user.
  name: Name of the user.
  email: Email address.
  password: Account password.
  phoneNumber: Contact number.
- ***Methods:***
  Register(): Registers a new user into the system.
  Login(): Authenticates a user.
  ForgotPassword(): Allows the user to recover the password.

4.4.3 *AuthenticationManager Class*

- ***Attributes:***
  token: Session token used for authentication.
- ***Methods:***
  GenerateToken(userID): Generates a token for the given user.
  ValidateToken(): Validates the session token.

4.4.4 *Video Class*

- ***Attributes:***
  videoID: Identifier for the YouTube video.
  title: Title of the video.
  URL: Link to the video.
  uploadDate: Date when the video was uploaded.
- ***Methods:***
  FetchComments(): Retrieves comments from the video.
  GetVideoDetails(): Gets metadata of the video.

4.4.5 *Comment Class*

- ***Attributes:***
  commentID: Identifier for the comment.
  videoID: ID of the video the comment belongs to.
  userName: Author of the comment.
  text: Content of the comment.
  category: Category such as positive, negative, question, suggestion, abusive.
- ***Methods:***
  AnalyzeSentiment(): Determines sentiment of the comment.
  CategorizeComment(): Classifies the comment.

4.4.6 *Analyzer Class*

- ***Attributes:***
  analyzerID: Unique identifier for the analyzer.
- ***Methods:***
  PerformSentimentAnalysis(): Analyzes all comments.
  CategorizeComments(): Sorts comments into categories.
  SummarizeComments(): Creates a summary of comment analysis.

4.4.7 *Report Class*

- ***Attributes:***
  reportID: Identifier for the report.
  videoID: ID of the video the report is based on.
  userID: ID of the user generating the report.

generateDate: Date the report was created.
summary: Summary text of the analysis.

- ***Methods:***
  GenerateGraph(): Creates a visual chart of results.
  GenerateSummary(): Generates report summary.
  GetReportDate(): Returns report date.

4.4.8 *PDFExporter Class*

- ***Attributes:***
  filePath: Location of the saved PDF.
- ***Methods:***
  ExportReportToPDF(): Exports report data to a PDF file.

4.4.9 *DatabaseManager Class*

- ***Attributes:***
  connectionString: Connection details for the database.
- ***Methods:***
  SaveUser(): Saves user data to the database.
  SaveComments(): Stores video comments.
  SaveReport(): Saves report data.
  FetchUser(): Retrieves user information.
  FetchComments(): Loads video comments from the database.

4.4.10 *Relationships*

User ↔ AuthenticationManager: AuthenticationManager manages login tokens for users.
User ↔ Report: A user owns multiple reports.
Video ↔ Comment: A video has many comments.
Analyzer ↔ Comment: Analyzer processes comments for sentiment and category.
Report ↔ Analyzer: Report is generated based on analyzer’s output.
Report ↔ PDFExporter: Report is exported to PDF.
DatabaseManager is responsible for storing and fetching data for User, Report, and Comment classes.

4.5 Sequence Diagram

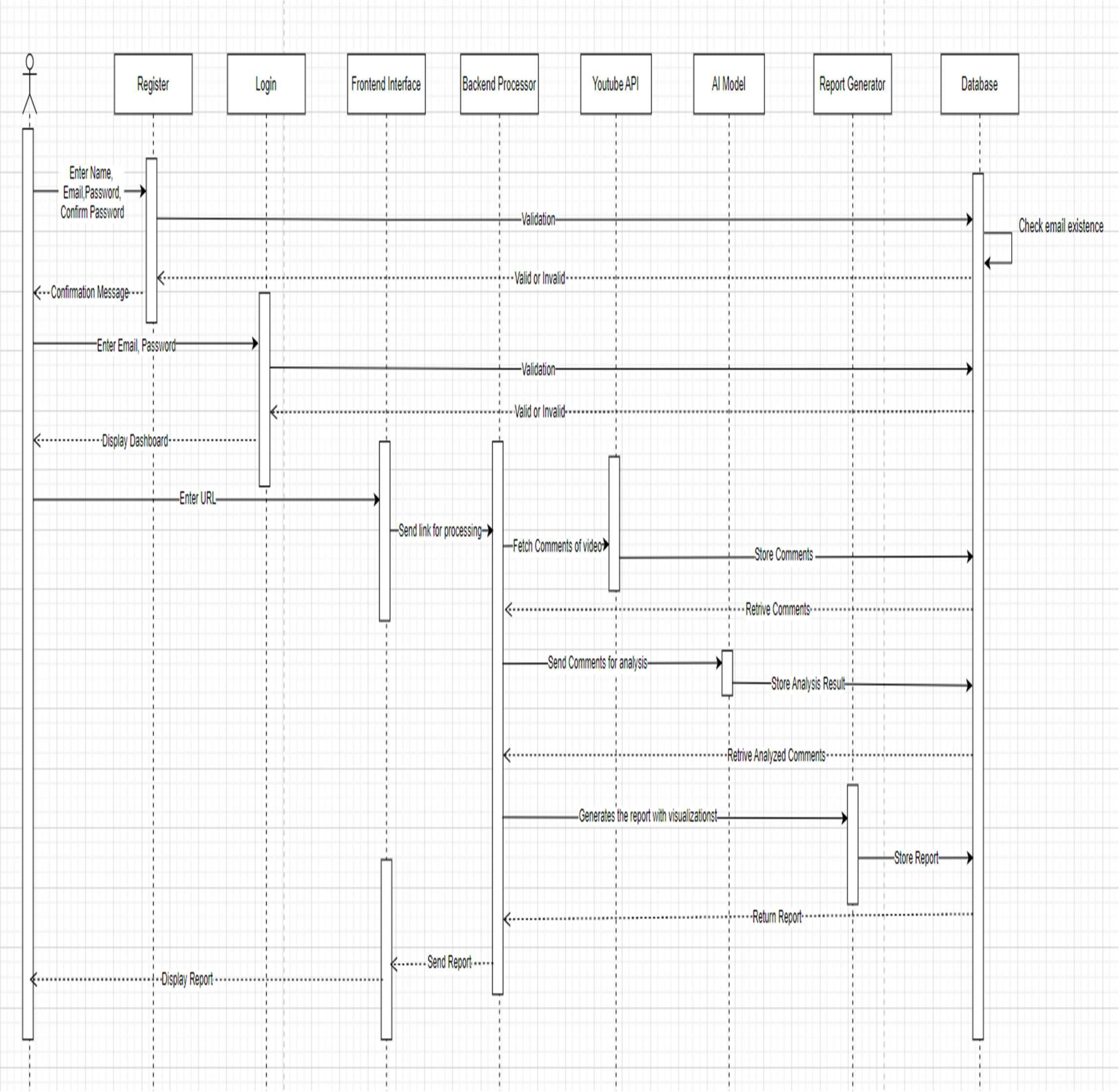


Figure 4.6: Sequence Diagram

4.5.1 Description

1. **User (Actor):**
- The person who is interacting with the system, performing registration, login, and submitting links for analysis.
2. **Register Module:**
- Handles user registration.
3. **Login Module:**
- Validates user credentials for login.
4. **Frontend Interface:**
- Manages user input and displays results.
5. **Backend Processor:**
- Handles backend operations such as processing links, fetching comments, and coordination between components.
6. **YouTube API:**
- Fetches comments from the provided YouTube video link.
7. **AI Model:**
- Performs sentiment analysis, abuse detection, and categorization.
8. **Report Generator:**
- Creates reports and visualizations based on analysis results.
9. **Database:**
- Stores user credentials, comments, and analysis results

Working Flow

1. ***Registration:***
- ***User Action:*** The user enters their name, email, password, and confirms their password in the registration module.
- ***System Validation:***
  - Checks if the email is already exists in the database.
  - Confirms the password match.
  - Sends a confirmation message for successful registration.
2. ***Login:***
- ***User Action:*** The user enters their email and password in the login module.
- ***System Validation:***
  - Validates the entered credentials with the database.
  - If valid, redirects the user to the dashboard.
3. ***Link Submission for Video Analysis:***
- ***User Action:*** The user enters a YouTube video URL in the frontend interface.
- ***Processing:***
  - The frontend sends the link to the backend processor for further processing.
4. ***Fetching Comments:***
- ***Backend Processor Action:*** Uses the YouTube API to fetch comments from the provided video.
- ***Data Handling:*** The fetched comments are stored temporarily in the database.
5. ***Analyzing Comments:***
- ***AI Model Action:*** Processes the comments to:
  - Analyze sentiment (positive, negative, neutral).

    - Detect abusive or offensive content.
    - Extract suggestions or questions.
- The analysis results are stored in the database.

6. Report Generation:

- ***Report Generator Action:*** Based on the analysis results, generates a report with visualizations (charts, graphs).
- Stores the report in the database.

7. Display Report:

- ***User Action:*** The user receives the report with visualizations through the frontend interface.

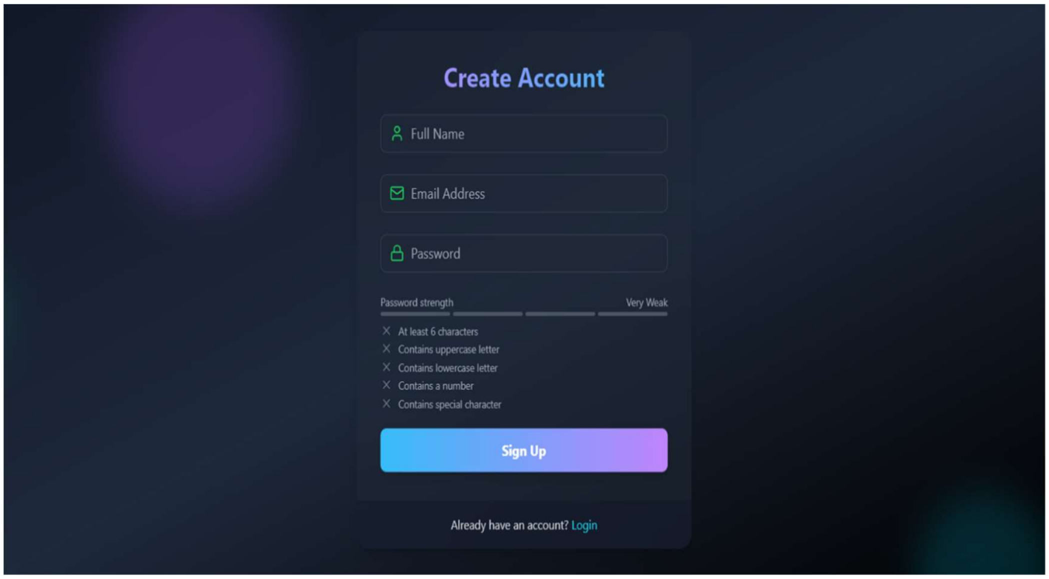
.

Chapter 5

Graphical User Interfaces (GUI)

5.1 Mockups:

Mockup 5.1: Register



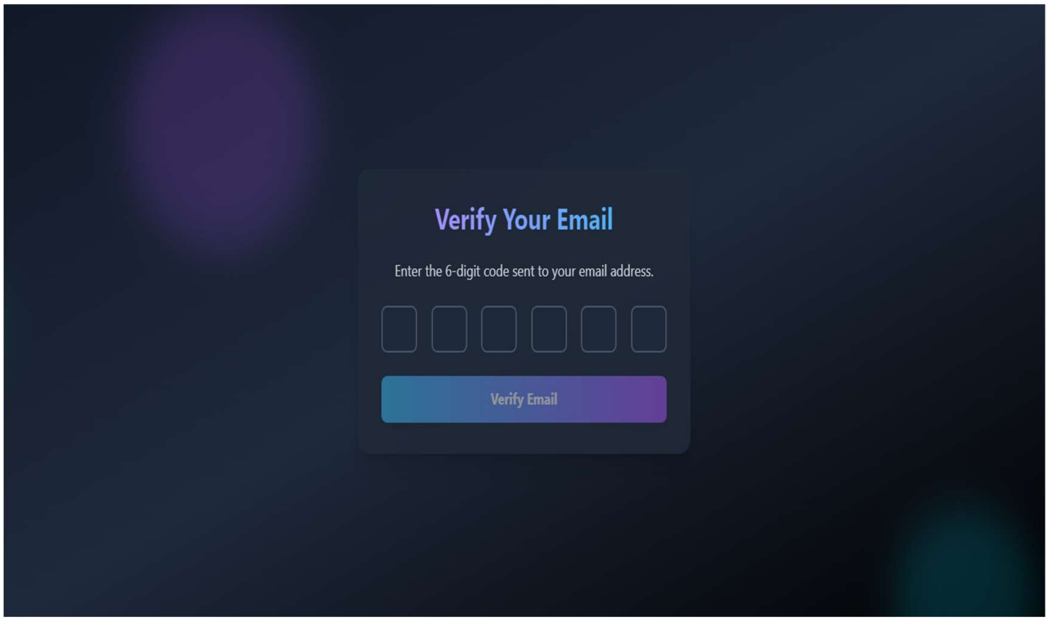
Create Account

✗ At least 6 characters
✗ Contains uppercase letter
✗ Contains lowercase letter
✗ Contains a number
✗ Contains special character

Already have an account? Login

Figure 5.1: Register

Mockup 5.2: Email Verification



Verify Your Email

Enter the 6-digit code sent to your email address.

Figure 5.2: Email Verification

Mockup 5.3: Login

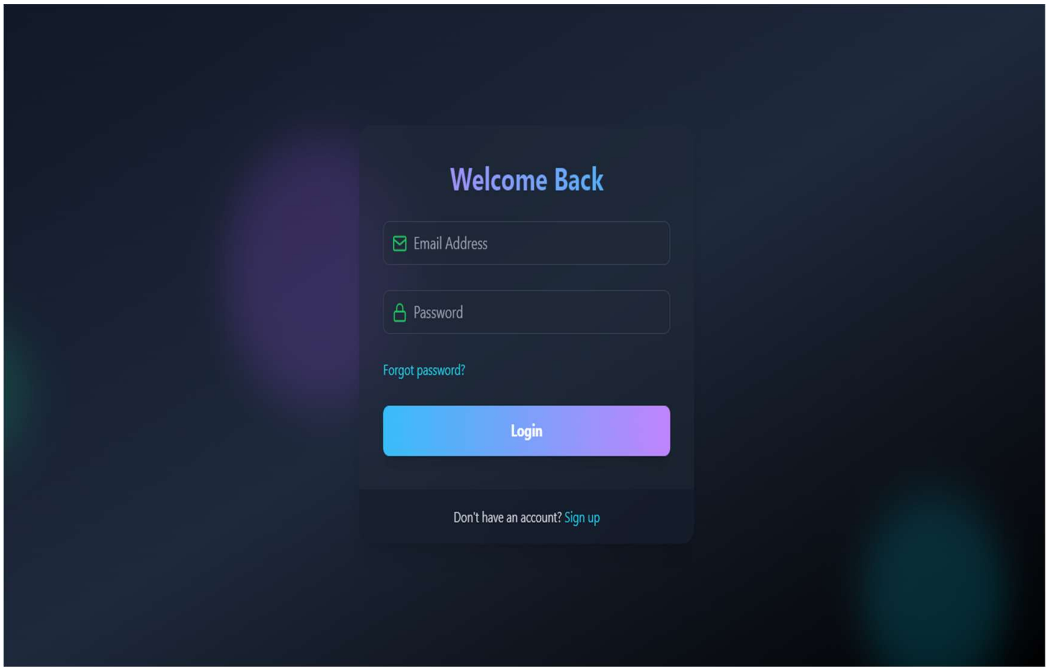


Figure 5.3: Login

Mockup 5.4: Forget Password

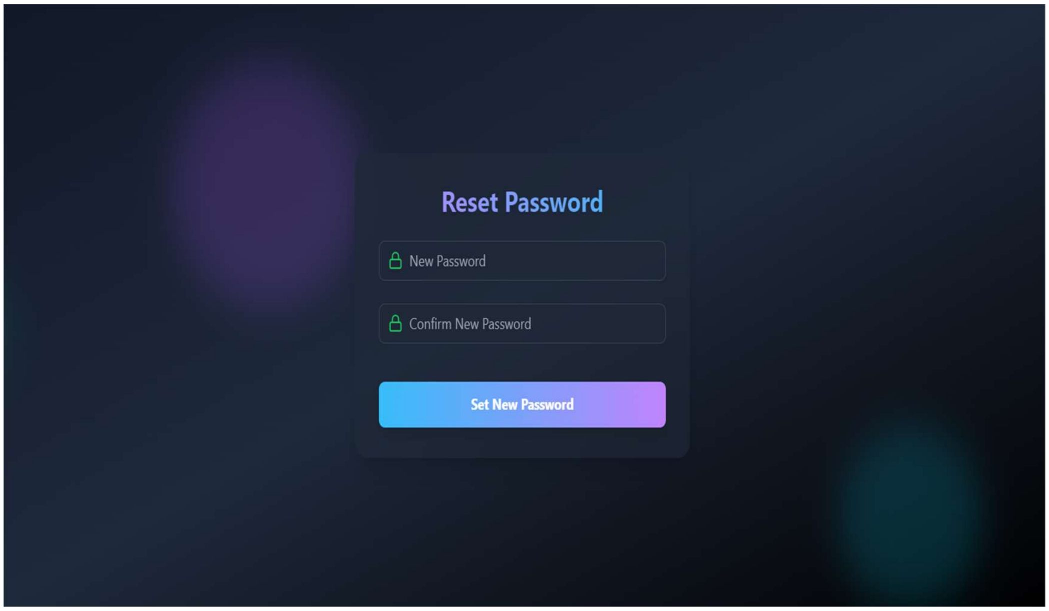


Figure 5.4: Forgot Password

Mockup 5.5: Reset Password

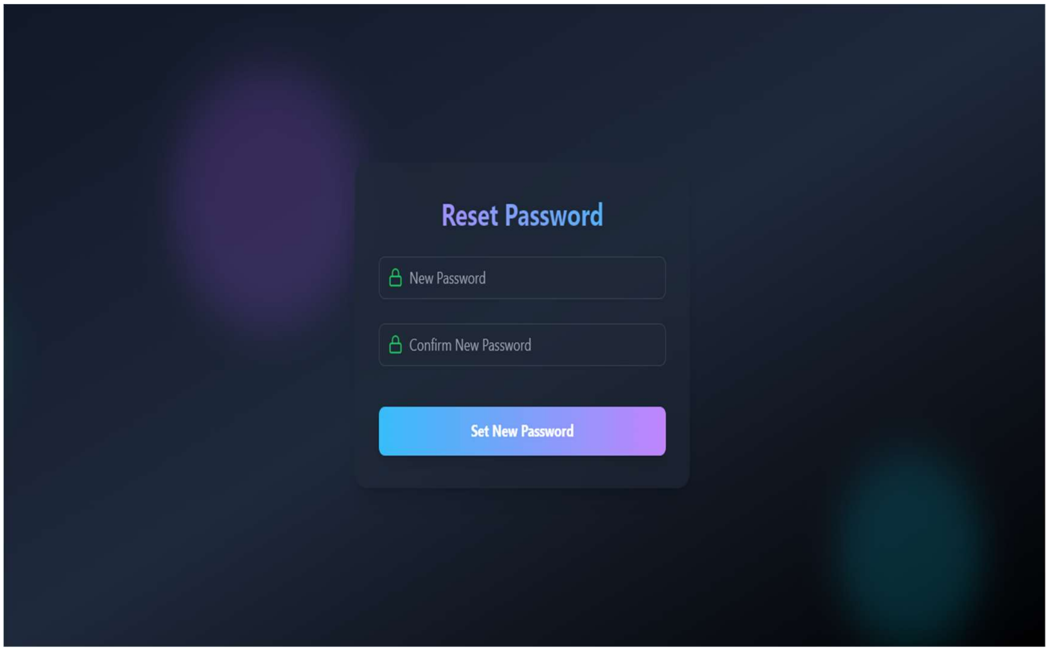
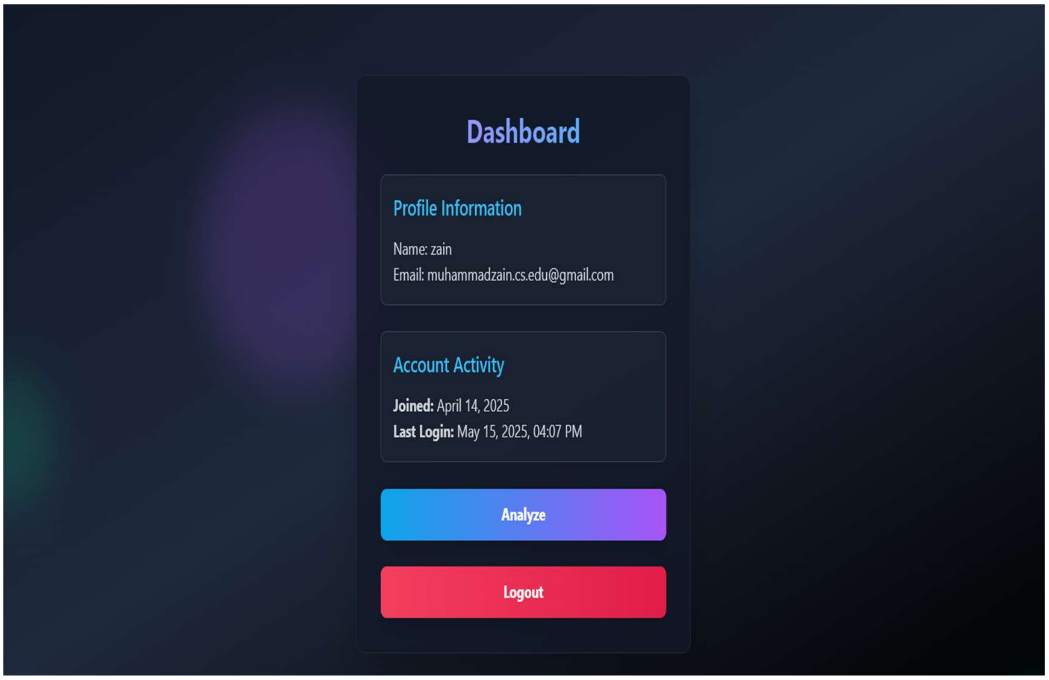


Figure 5.5: Reset Password

Mockup 5.6: Dashboard



Name: zain

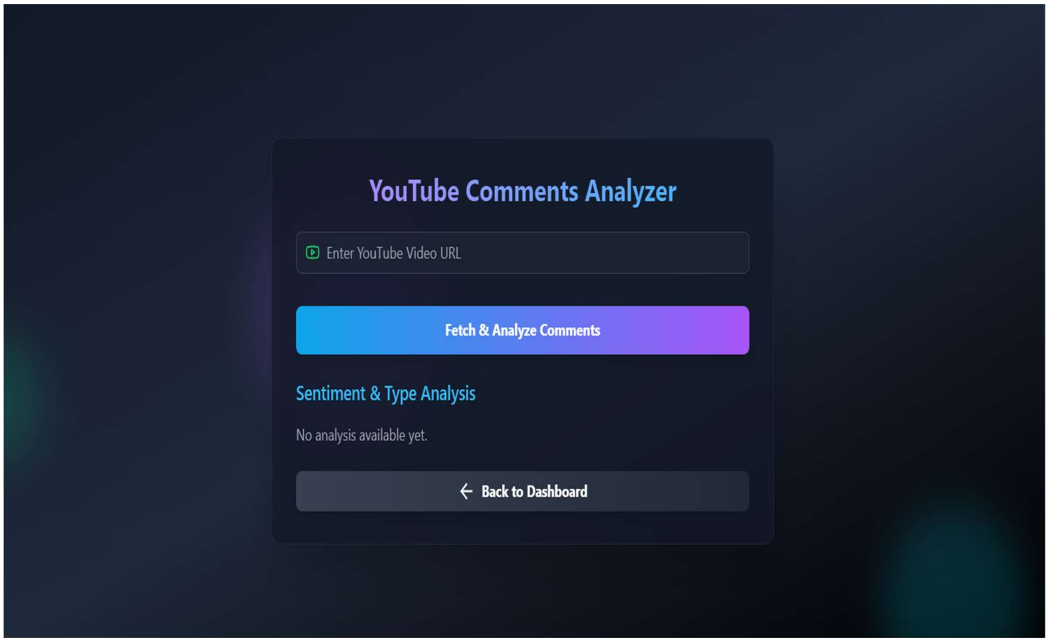
Email: muhammadzain.cs.edu@gmail.com

Joined: April 14, 2025

Last Login: May 15, 2025, 04:07 PM

Figure 5.6: Dashboard

Mockup 5.7: Home Page



YouTube Comments Analyzer

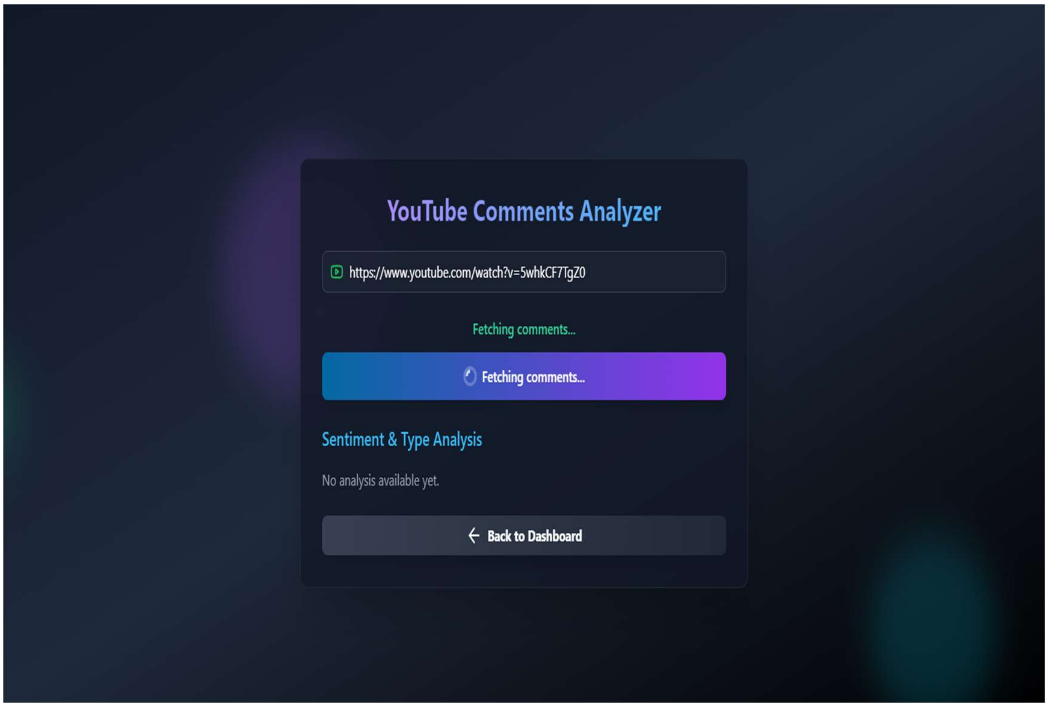
Enter YouTube Video URL

Sentiment & Type Analysis

No analysis available yet.

Figure 5.7: Home Page

Mockup 5.8: Fetching



YouTube Comments Analyzer

https://www.youtube.com/watch?v=5whkCF7TgZ0

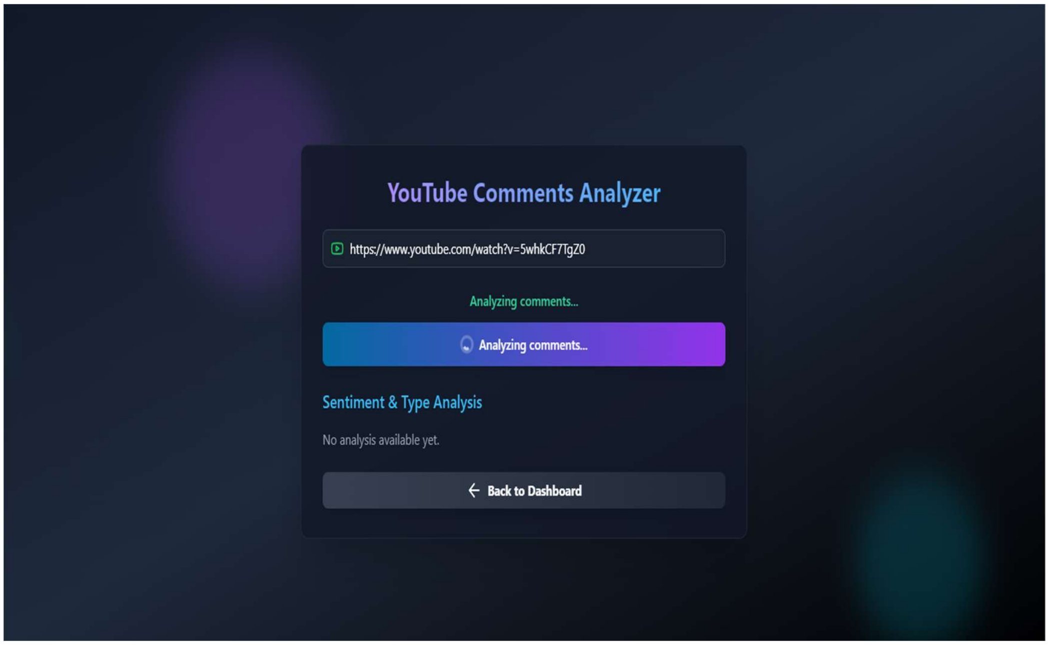
Fetching comments...

Sentiment & Type Analysis

No analysis available yet.

Figure 5.8: Fetching

Mockup 5.9: Analyzing



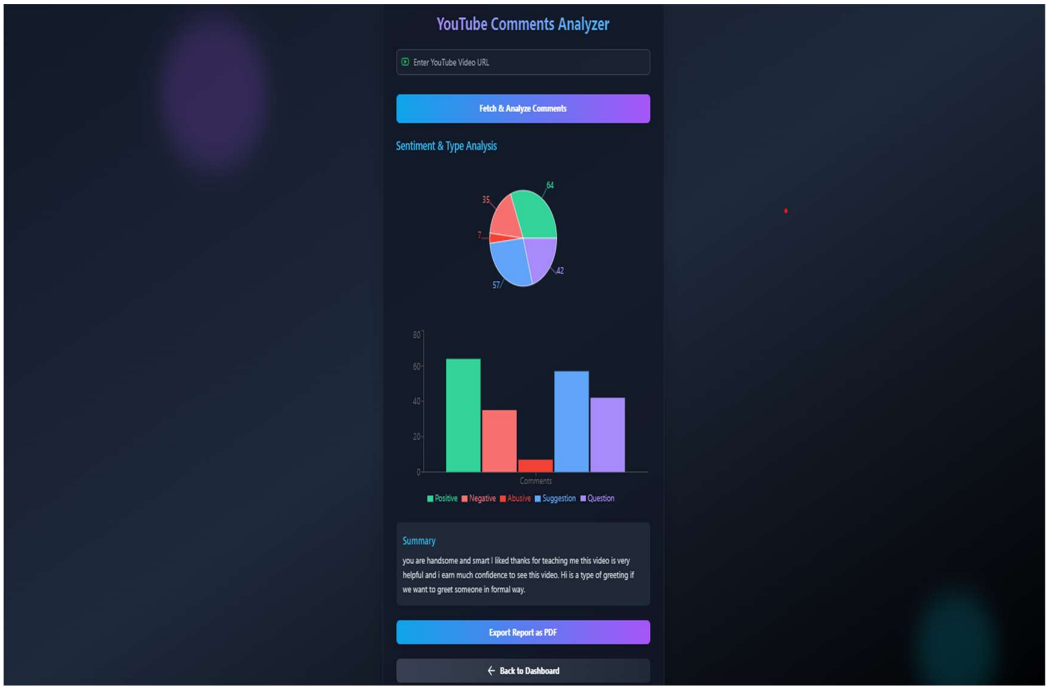
https://www.youtube.com/watch?v=5whkCF7TgZ0

Analyzing comments...

No analysis available yet.

Figure 5.9: Analyzing

Mockup 5.10: Report



Summary

you are handsome and smart I liked thanks for teaching me this video is very helpful and i earn much confidence to see this video. Hi is a type of greeting if we want to greet someone in formal way.

Figure 5.10: Report

Chapter 6

Testing

6.1 Introduction

Testing is a complex phase in the SDLC that ensures the system which is developed, meets specified requirements and functions are working correctly under all conditions. The YouTube Comments Analyzer was tested by these techniques: black-box and white-box to validate the performance, functionality, and reliability.

6.2 Test Scenario

Scenario ID	Scenario Description	Expected Outcome
TS01	User registration with valid data	Account created and redirected to login
TS02	Login with incorrect credentials	Error message displayed
TS03	Entering valid YouTube video URL	Comments fetched and display results
TS04	Submitting video without a URL	Error or validation message shown
TS05	Generating report and downloading it	Report downloaded in PDF format

Table 6.1: Test Scenario

6.3 Test Plan

The testing plan ensures that all modules of the project are tested systematically:

- **Test Type:** Functional, Black Box, White Box
- **Tested By:** Development Team
- **Test Environment:** Localhost
- **Tools Used:** Postman (API Testing), Browser DevTools, Console, MongoDB Compass

Module	Tested Feature	Status
User Authentication	Registration, Login, Validation	Passed
YouTube Video Processing	URL handling, Comment fetching	Passed
AI Integration	Sentiment & Category analysis	Passed
Visualization	Charts, Graphs	Passed
Report Generation	Export to PDF	Passed

Table 6.2: Test Plan

6.4 Definition of Test Cases

A test case is a set of conditions under which a tester will determine whether a system satisfies the requirements and works correctly.

Each test case includes:
Test Case ID
Objective
Input Data
Expected Result
Actual Result
Status (Pass/Fail)

6.5 Test Case

Test Case ID: 01
Test Case Name: Registration
Test Priority: High
Preconditions: The user must not have an existing account with the same email.
Post conditions:

1. The user account is created.
2. The user is redirected to the login page.

SN	Action	Inputs	Expected Outcome	Actual Output	Test Application	Test result	Test comments
1	Launch Application	http://localhost:5173/signup	Signup Page	Signup Page	Google Chrome	PASS	Launch Successful
2	Enter invalid detail	http://localhost:5893/sign	Page load unsuccessfully	Page load unsuccessfully	Google Chrome	PASS	Launch unsuccessfully
3	Enter valid details	Name: Ali, Email: ali@gmail.com, Password: ****	Registration Success	Registration Success	Google Chrome	PASS	Account Created
4	Enter invalid details	Name: 123, Email: ali@com, Password: ****	Registration unsuccessfully	Registration unsuccessfully	Google Chrome	PASS	Account not created

Table 6.3: Registration

Test Case ID: 02
Test Case Name: Login
Test Priority: High
Preconditions: The user must be registered.
Post conditions: The user logs into the system, the user accesses the dashboard.

SN	Action	Inputs	Expected Outcome	Actual Output	Test Application	Test result	Test comments
1	Launch Application	http://localhost:5173/login	Login Page	Login Page	Google Chrome	PASS	Launch Successful
2	Enter credentials	Email: ali@gmail.com, Password: ****	Login Successful	Login Successful	Google Chrome	PASS	Logged In
3	Enter invalid credentials	Email: ali@gmail, Password: ****	Login unsuccessfully	Login unsuccessfully	Google Chrome	PASS	Login Error

Table 6.4: Login

Test Case ID: 03
Test Case Name: Forgot Password
Test Priority: High
Preconditions: User must be registered.
Post conditions: Password will recovered

SN	Action	Inputs	Expected Outcome	Actual Output	Test Application	Test result	Test comments
1	Launch Application	http://localhost:5173/forgot-password	Forgot Password Screen	Forgot Password Screen	Google Chrome	PASS	Forgot Password Screen Opened
2	Enter valid credentials	Email:abc123@gmail.com	Reset link sent	Reset link sent	Google Chrome	PASS	Link received
3	Enter invalid credentials	Emal:ABC @gmail.com	User not Found	User not Found	Google Chrome	PASS	Password will not recovered

Table 6.5: Forgot Password

Test Case ID: 04
Test Case Name: Analyzer
Test Priority: High
Preconditions: User must be logged in.
Post conditions: Comments are fetched and displayed result.

SN	Action	Inputs	Expected Outcome	Actual Output	Test Application	Test result	Test comments
1	Launch Application	http://localhost:5173/comments	Analyzer Screen	Analyzer Screen	Google Chrome	PASS	Analyzer Load
2	Enter Video URL	https://youtube.com/watch?v=example	Display Result	Display Result	Google Chrome	PASS	Result Show
3	Enter invalid URL	https://youtatch?v=example	Invalid URL Message	Invalid URL Message	Google Chrome	PASS	Display error message
4	Download Report	Click on Export Report	Report Downloaded	Report Downloaded	Google Chrome	PASS	Report download in PDF format

Table 6.6: Analyzer

6.6 Test Case Result

6.6.1 Black Box Test Cases

Test Case ID	Test Description	Expected Output	Actual Output	Result
BB01	Submit empty registration form	Show validation errors	Show validation errors	PASS
BB02	Enter invalid video URL	Show error message	Show error message	PASS
BB03	Generate report with no comments	Show message: No data	Show message: No data	PASS

Table 6.7: Black Box Test Cases

6.6.2 White Box Test Cases

Test Case ID	Function/Module	Test Description	Result
WB01	Comment Fetch API	Validates response structure and data parsing	PASS
WB02	Sentiment Analysis Function	Checks accuracy and label assignment	PASS
WB03	Chart Rendering Component	Ensures correct data binding and rendering	PASS

Table 6.8: White Box Test Cases

Chapter 7

Conclusion and Future work

7.1 Conclusion

The YouTube Comments Analyzer is a web-based solution which is used for analysis of user feedback on YouTube videos. With the help of AI model, the system effectively categorizes comments into positive, negative, questions, suggestions, and abusive. This categorization helps the creators to gain deeper insights of viewers.

This analyzer is integrate with the help of MERN stack and AI model. This project includes an interactive dashboard with visual reports and downloadable complete report. This analyzer has user-friendly interface, also with secure login and history tracking feature which ensures ease of access and usability for end-users.

This tool makes ease for content creator to take decision by quick and accurate comment analysis. This project achieves another objective that convert raw YouTube comments into meaningful report and summary.

7.2 Future Work

In the future, this project can be enhanced by adding advanced natural language processing models, such as transformer-based architectures, for improve accuracy of sentiment. Multiple language support can also be added to analyze comments in different languages. Additionally, real-time comment analysis like live video's comments takes and analyze it and generate its report. And integration of different other social media platforms like Facebook, Twitter, and Instagram can make this system a universal comment analysis tool, which can analyze comments of different social media platform and in different languages. Lastly, improving data visualization and integrating notification systems, when abusive comment comes, it gives alerts message.