**Abstract:**

This web application allows users to upload, view, and interact with videos. The main features of the application include video playback, viewing video metadata, liking/disliking videos, posting comments, and viewing related videos. The video player is embedded in a responsive layout with a collapsible description and metadata such as tags and categories. Users can also interact with the video by liking or disliking it, which updates in real-time. Additionally, users can post comments, which are dynamically added to the comment section below the video. Suggested videos are displayed in a sidebar, allowing users to explore other content. The application uses client-server communication with AJAX to handle likes, dislikes, and comments without page reloads.

**1. Introduction**

This project aims to create a video streaming platform, which allows users to upload, view, like, dislike, and comment on videos. The platform is accessible through a public IP address on a Virtual Private Server (VPS). Users can interact with the website by signing up, logging in, and participating in social activities, such as liking and commenting on videos.

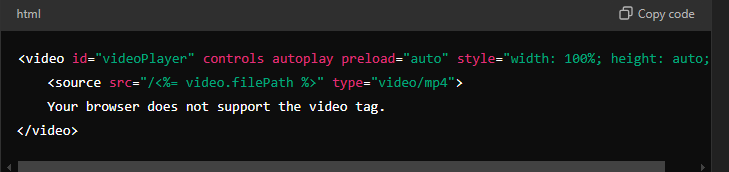
**2. Project Objective**

* **To build a video streaming platform**: A website that allows users to upload and watch videos.
* **To integrate user management**: Features such as login, signup, and session management.
* **To support video interactions**: Like, dislike, and comment on videos.
* **To make it accessible via a VPS**: Enabling access from any device using a public IP that is **199.129.96.0**.

### 3. ****Features and Functionalities****

#### **Video Playback:**

#### The video player is embedded using HTML5 <video> elements. It supports autoplay, preload, and controls to allow users to interact with the video player interface. The video file path is dynamically passed from the backend to the frontend.

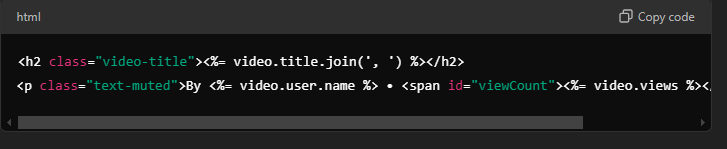


 The video is fetched from the server using the path stored in the database.

 It supports basic controls (play, pause, volume, etc.) and includes a fallback message for unsupported browsers.

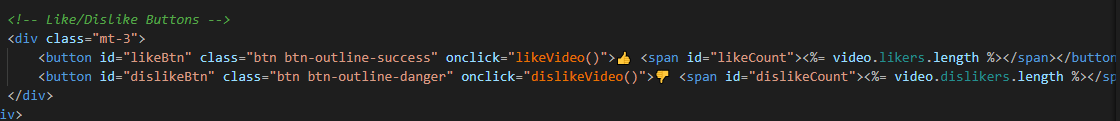
#### **Video Metadata**:

#### The system displays video information like the title, uploader, views, tags, and category. The title and description are presented dynamically from the database.



#### **Like/Dislike Interaction**:

#### The system allows users to like or dislike a video. Each interaction is handled through JavaScript, making asynchronous requests to update the likes and dislikes count without reloading the page.

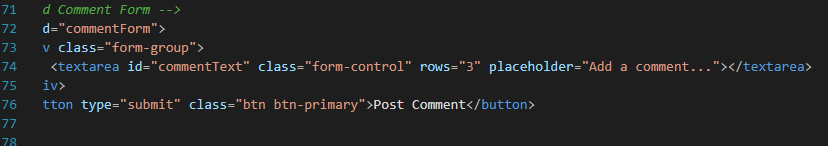


Functions like-video and dislike-Video send POST requests to the server, which updates the database and returns the new counts.



#### **Comment Section**:

#### Users can comment on videos. The comment form submits the comment text to the backend, which is stored in the database and then displayed in real time on the page without requiring a refresh.



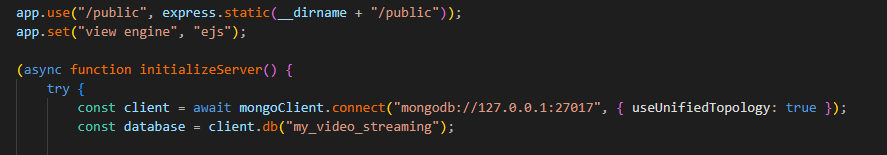
he form sends the comment text to the backend using a POST request.



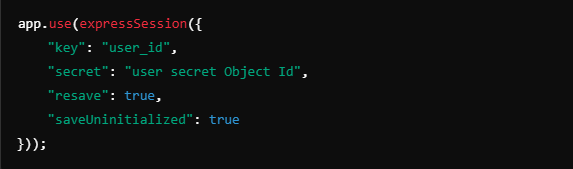
**4.Server-Side Code:**

The backend is built using Node.js with Express. Below is an overview of key components:

**MongoDB Connection**:

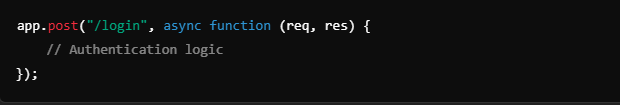


**Session Management:**

****

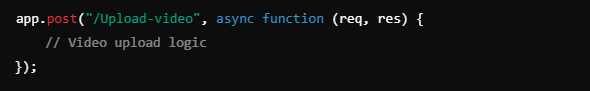
**Signup and Login Routes**:

The /signup and /login routes handle user registration and authentication using bcrypt for password hashing.



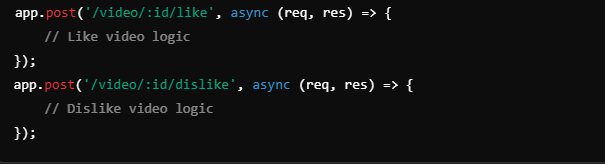
**Video Upload and Streaming**:

Videos are uploaded using formidable and stored in the server's directory. Users can stream videos directly through dynamic URLs.



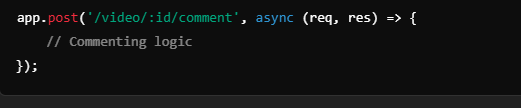
**Like and Dislike Logic**:

Users can like or dislike videos. The respective counts are updated in the database.

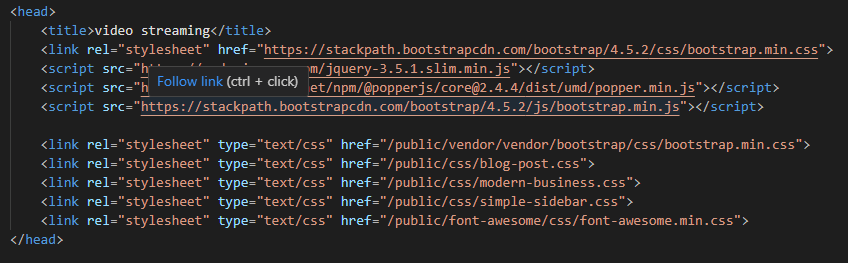


**Add Comments**:

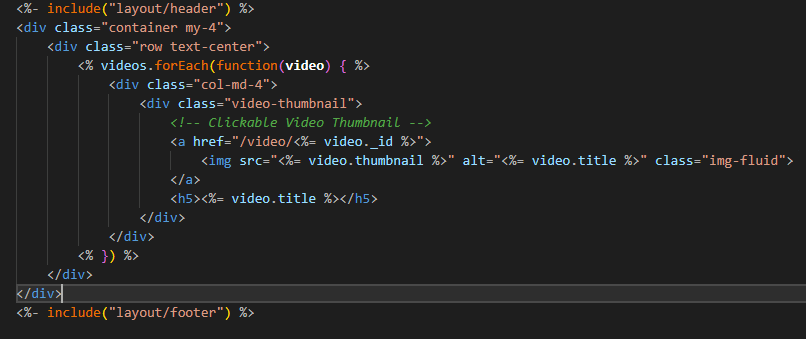
Users can add comments to videos. Comments are stored in the database.



**Header code:**

****

**Index Code:**

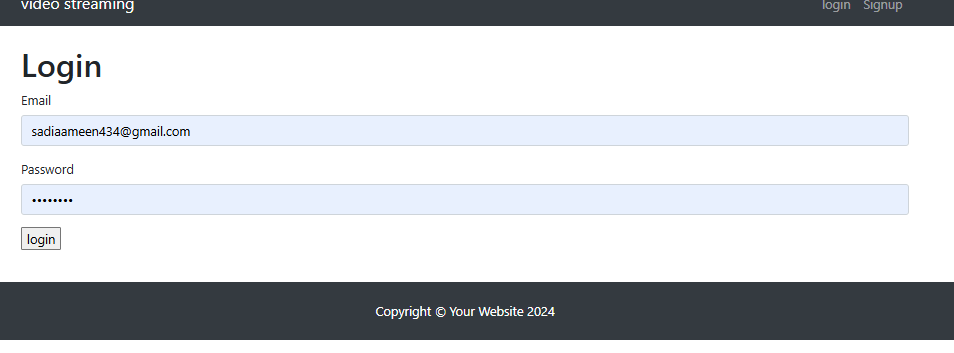
****

**5. Testing**

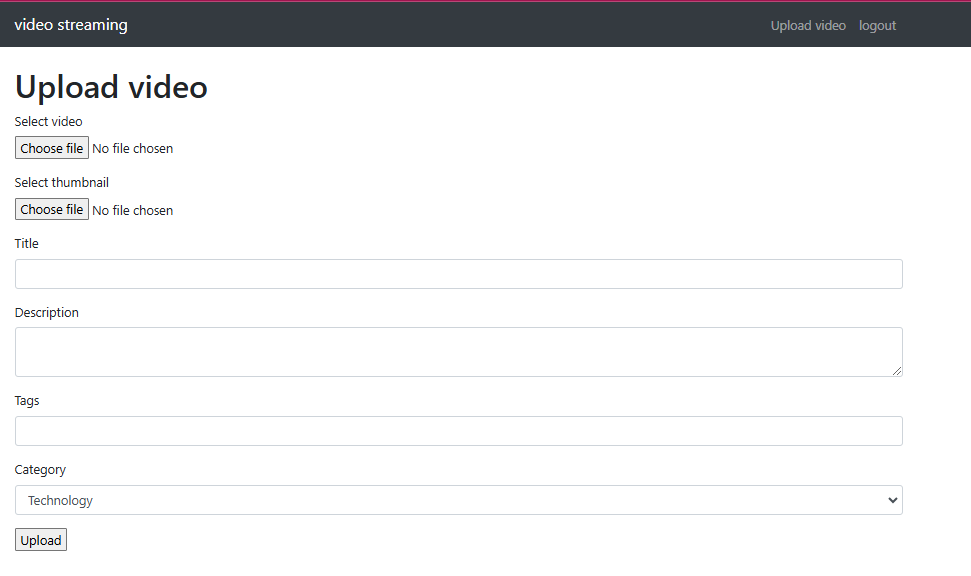
For testing, the following steps were performed:

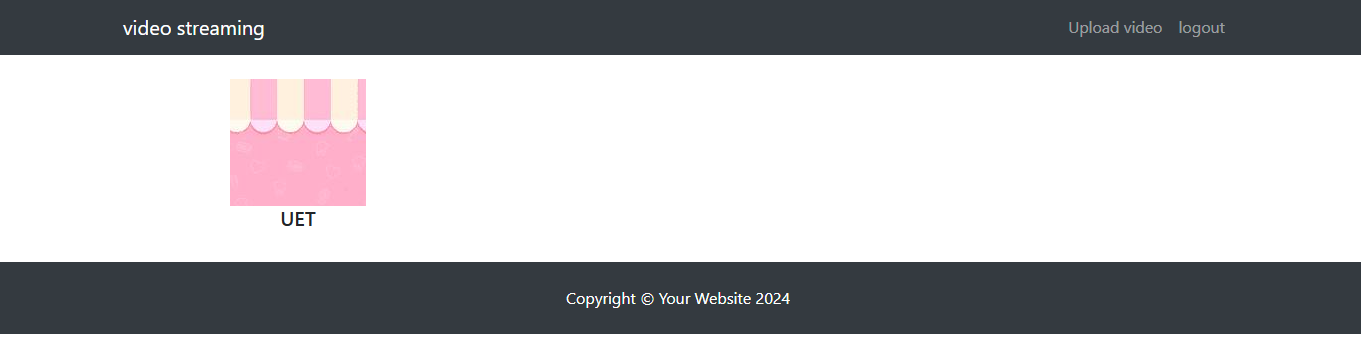
**Functional Testing:**

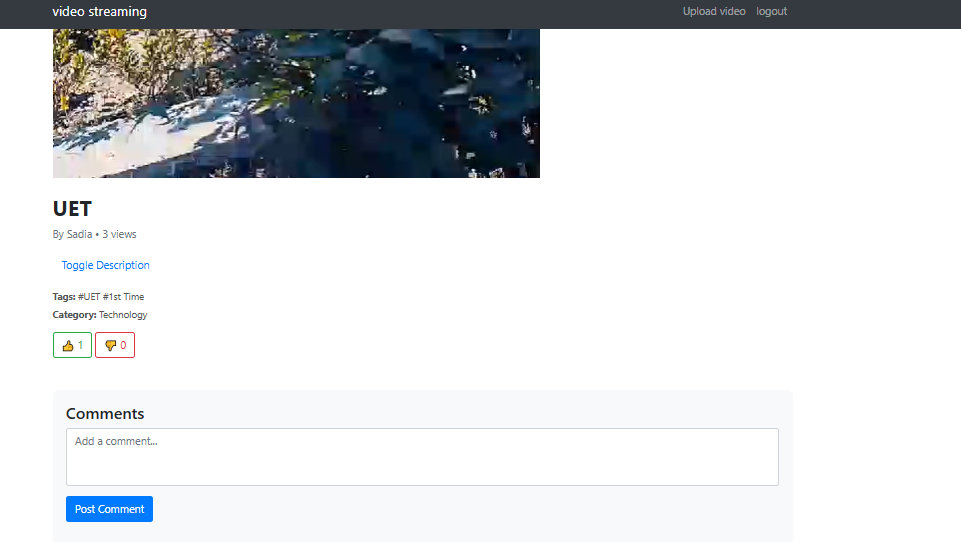
* **Sign Up/Login**: Tested the login and signup process with valid and invalid credentials.
* **Video Upload**: Uploaded various video formats and tested the streaming functionality.
* **Like/Dislike**: Ensured that likes and dislikes are counted correctly and stored in the database.
* **Commenting**: Verified that comments are added correctly to videos.





****

****

****