Project Title	Video Streaming Platform
Technologies	MERN
Domain	Social Media
Project Level	Hard

Table

Table of Contents

1. Problem Statement:

Design a MERN stack-based Video Streaming Platform.

1.1. Overview of MERN Stack and Video Streaming Platform

To create scalable master web applications with backend, frontend, and database components, developers employ the MERN Stack, a set of strong technologies. The quicker and simpler creation of full-stack web applications is made possible by JavaScript. A user-friendly full-stack JavaScript framework for creating applications and dynamic webpages, MERN Stack is a technology.

A video streaming platform is an application in which a user can upload, delete and watch a video which is uploaded there by him or by another user. The platform will also provide additional features with these three in order to make it more interactive and engaging. All the users can share their videos there and will be able to see it on the main page from where they can play it.

Note: YouTube is one of the most widely used video streaming websites. You can have a look at that to get a better understanding of how a streaming platform operates and what features can be anticipated to make your platform more interesting and practical compared to competing platforms.

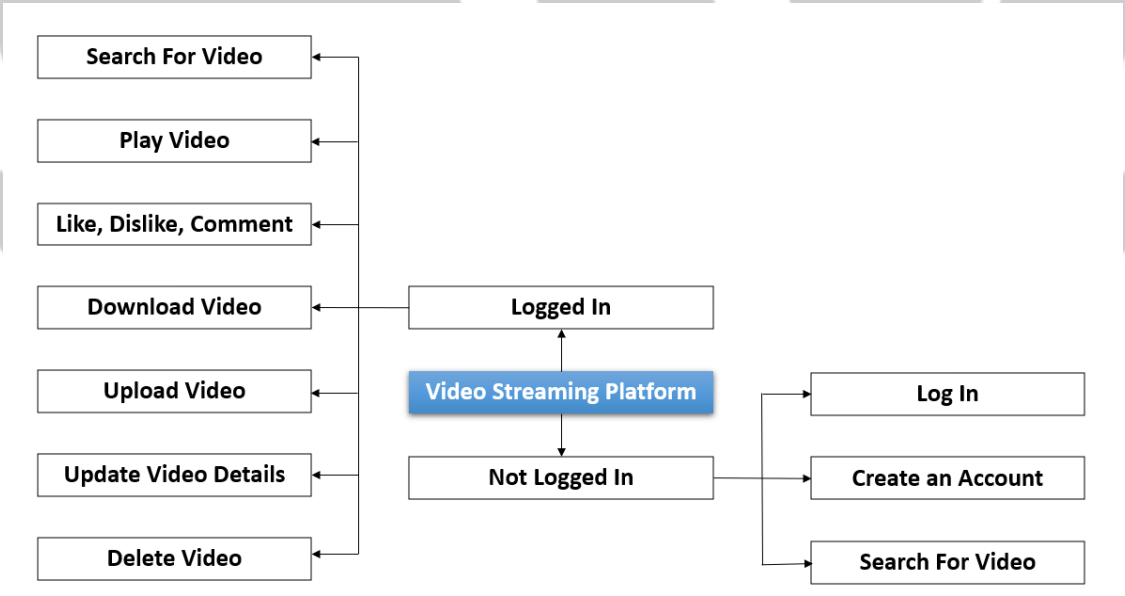
1.1.1 Features of the Video Streaming Application

These are the minimum requirements that your video streaming platform should meet-

a. When a user initially visits the website, they should be able to sign up or log in (if already registered).

b. The user must be logged in before he can play any of the listed videos. If he attempts to play a video without logging in, he will be directed to do log in.
c. A user should be allowed to upload his own videos on the site itself once they have logged in.
d. Any user who is logged in can see, comment, like, and dislike of his video.
e. To provide the person who is going to see the video with more information, the total number of views, comments, likes, and dislikes should be provided underneath the video.
f. The user will receive a dashboard where he may examine all of the videos he has published to date as well as information about each video, including views, likes, and dislikes.
g. From the dashboard, a user can delete any of his videos or alter their titles, descriptions, and other information.
h. So that any user can quickly search for the necessary video, a search function must to be provided at the top.
i. Every video should have a category option when uploading it, such as entertainment, education, politics, etc. The application should offer a category option for the user to choose from when uploading any of his videos.
j. From the site, users can categories the videos.
k. In order to make a video easily discoverable, a user can also search for videos based on their name and category.

Low Level user flow



1.2. Project Objective

• **Video Streaming Platform:** It is a fantastic moment to develop our own video streaming platform at this point because, according to a poll, the average Indian spends about 2-3 hours per day on a video streaming platform like YouTube, Amazon Prime, Netflix, etc. This will make it easier for us to draw large audiences to our platform. This is a technology that is widely used and in high demand.

• **User-friendliness:** The user should have little trouble creating, uploading, and editing his video without getting bogged down by numerous options and functionalities.

• **Easy to Use:** A lot of new users should be able to use the platform with ease if the user interface is simple to use and intuitive.

1.3. Scope of The Project

1. Today's demand may be shown by the massive daily traffic that video streaming services receive.

2. You will be able to see how vast and how they operate thanks to this project.

1.4. Functional and Non-Functional Requirements: -

1.4.1. Functional Requirements

1. **User Registration:** An email address, a username, and a password must be required for the user to register for the application. Users must be able to register themselves upon opening the application, or if they already have an account, they can log in immediately.

2. **Videos:** After logging in, the user should be able to watch, like, and comment on other people's videos. From the dashboard, he will be able to upload, edit, and remove his video.

1.4.2. Non-Functional Requirements

1. **Privacy:** The user shouldn't be able to access the dashboard or watch the video unless he logs into his account.

2. **Robustness:** A backup of the user's uploads should be available in case the user's system breaks so that the data loss can be addressed.

3. **Performance:** The application must load the films promptly on the main page and be small.

1.5. Use Case Table

Authentication System	Register, Login, Logout	User
Search Form	Search Videos	User

User Dashboard	Display all the uploaded videos	User

Table 1. Use Case

2. Project Evaluation Metrics:

2.1. Code:

- You are supposed to write code in a modular fashion
- Safe: It can be used without causing harm.
- Testable: It can be tested at the code level.
- Maintainable: It can be maintained, even as your codebase grows.
- Portable: It works the same in every environment (operating system).
- You have to maintain your code on GitHub.
- You have to keep your GitHub repo public so that anyone can check your code.
- Proper readme file you have to maintain for any project development.
- You should include the basic workflow and execution of the entire project in the readme file on GitHub.
- Follow the coding standards.

2.2. Database:

MongoDB is a source-available cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with optional schemas.

2.3. API Details or User Interface:

You have to expose your complete solution as an API or try to create a user interface for your model testing. Anything will be fine for us.

2.4. Deployment:

Implementation of reverse proxy, load balancing, and security group is mandatory for deployed applications.

2.5. Solutions Design:

You have to submit complete solution design strategies in High-level Document (HLD), Low-level Document (LLD), and Wireframe documents**.**

2.6. System Architecture:

You have to submit a system architecture design in your wireframe document and architecture document.

2.7. Optimization of solutions:

Try to optimize your solution on code level, architecture level, and mention all of these things in your final submission.

Mention your test cases for your project.

3. Submission requirements:

3.1. High-level Document:

You have to create a high-level document design for your project. You can reference the HLD form below the link.

Sample link: HLD Document Link

3.2. Low-level document:

You have to create a Low-level document design for your project; you can refer to the LLD from the link below.

Sample link: LLD Document Link

3.3. Architecture:

You have to create an Architecture document design for your project; you can refer to the Architecture from the link below.

Sample link: Architecture sample link

3.4. Wireframe:

You have to create a Wireframe document design for your project; refer to the Wireframe from the link below.

Demo link: Wireframe Document Link

3.5. Project code:

You have to submit your code to the GitHub repo in your dashboard when the final submission of your project.

Demo link: Project code sample link

3.6. Detail project report:

You have to create a detailed project report and submit that document as per the given sample.

Demo link: DPR sample link

3.7. Project demo video:

You have to record a project demo video for at least 5 Minutes and submit that link as per the given demo.

3.8. The project LinkedIn a post:

You have to post your project details on LinkedIn and submit that post link in your dashboard in your respective field.