

Capstone Project: Develop a YouTube Clone Using the MERN Stack (400 marks)

Objective: Create a full-stack YouTube clone where users can view and interact with videos.

This project will help you understand how to build a real-world application using MongoDB, Express, React, and Node.js (MERN stack).

Requirements:

Front-End (React)

1. Home Page:

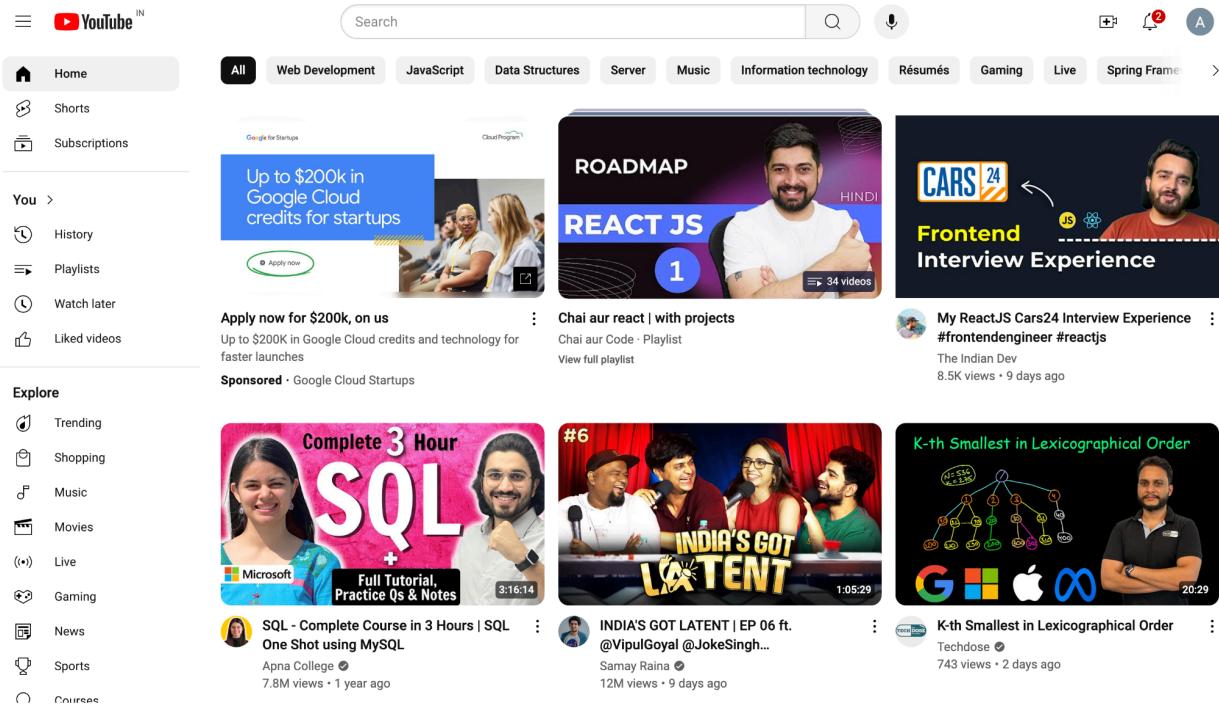
- Display a YouTube Header.
- Display a static sidebar which you can toggle from the top hamburger menu.
- Display filter buttons at the top.
- Display a grid of video thumbnails.
- Each video card should show:
 - Title
 - Thumbnail
 - Channel Name
 - Views

Sample data for videos:

Sample Data:

```
[ { "videoId": "video01", "title": "Learn React in 30 Minutes", "thumbnailUrl": "https://example.com/thumbnails/react30min.png", "description": "A quick tutorial to get started with React.", "channelId": "channel01", "uploader": "user01", "views": 15200, "likes": 1023, "dislikes": 45, "uploadDate": "2024-09-20", "comments": [ { "commentId": "comment01", "userId": "user02", "text": "Great video! Very helpful.", "timestamp": "2024-09-21T08:30:00Z" } ] }]
```

Home Page will look like this:



2. User Authentication:

- Allow users to register and log in with:
 - Username
 - Email
 - Password
- Use JWT for authentication.

Before Sign In , header should have sign in button



When the user clicks on sign in , then it should take to a new URL where a google form to login and register should be present.

User should sign in and after signing in , his/her name should be present at the top and the home page will be displayed.

Sample data for users:

```
{ userId: "user01", username: "JohnDoe", email: "john@example.com", password: "hashedPassword123", avatar: "https://example.com/avatar/johndoe.png", channels: ["channel01"], }
```

3. Search and Filter Functionality:

- Implement a search bar on the homepage.
This search bar is present in the header. Filter videos based on title.
- Filter buttons should be implemented and filters should work based on category.

4. Video Player Page:

- Display the selected video with:
 - Video player
 - Title and description
 - Channel name
 - Like and dislike buttons
 - Comment section (**Ignore** nested comments for now)

Users should be able to add , edit and delete comments. When a new comment is added , then it should be saved in the database along with that video.

React JS roadmap | chai aur react series

Spons Play (k) 0:28 parallels.com

Chai aur Code 419K subscribers

862K views 1 year ago

Visit <https://chaiacode.com> for all related materials, community help, source code etc.

Sara code yaha milta h <https://github.com/hiteshchoudhary/> ...more

23K Share Download

Chai aur react | with projects

Chai aur Code 1 / 34

1. **React JS roadmap | chai aur react series** 30:18 Chai aur Code
2. **Create react projects | chai aur react** 35:16 Chai aur Code
3. **Understand the react flow and structure** 35:59 Chai aur Code
4. **Create your own react library and JSX** 45:50 Chai aur Code
5. **Why you need hooks and project** 28:18 Chai aur Code
6. **06 Virtual DOM, Fibre and reconciliation** 21:21 Chai aur Code
7. **Tailwind and Props in reactjs** 21:21 Chai aur Code

A Add a comment...

V Pinned by Swiss View @Factory-qq2wk 2 months ago

The Schynige Platte Bahn has a retro atmosphere, and your video camera captured the expansive panorama in detail and sharply from the perspective of this mountain railway, showing various angles. The green of the trees and mountains, enhanced by reflections of light, not only portrays shades but also brings out beautiful variations in color. ...

Read more

59 Reply

2 replies

@lockdown19 1 month ago Word 'Beautiful' isn't enough to describe this

33 Reply

4 replies

@iannorton2253 1 month ago I was lucky enough to work there in 2008, both summer and winter. It really is this beautiful and spectacular. I went on this train ride in winter, when all was covered with snow.

8 Reply

1 reply

@paulgrima3812 2 months ago Around every bend the WOW factor hasn't changed !!! Incredibly Gorgeous countryside

7 Reply

10:35 5.4K views 3 months ago

INDIA'S SMARTEST ENGINEERING STUDENT

Tanmay Bhat 1.2M views • 17 hours ago New

Japan itna SAAF kaise hai II Moto Vlog II Indian in Japan II

Ajay Pandey 1.9M views • 1 month ago

Why India Built This Impossible Railway Line

Lastly 1.3M views • 6 months ago

Peaceful Morning Walk in Swiss Town Interlaken Switzerland ...

Swiss View 50K views • 13 days ago

Dangerous Journey to India's Loneliest Village

Abhijeet Chauhan 1.4M views • 1 month ago

Alaska in 8K 60p HDR (Dolby Vision)

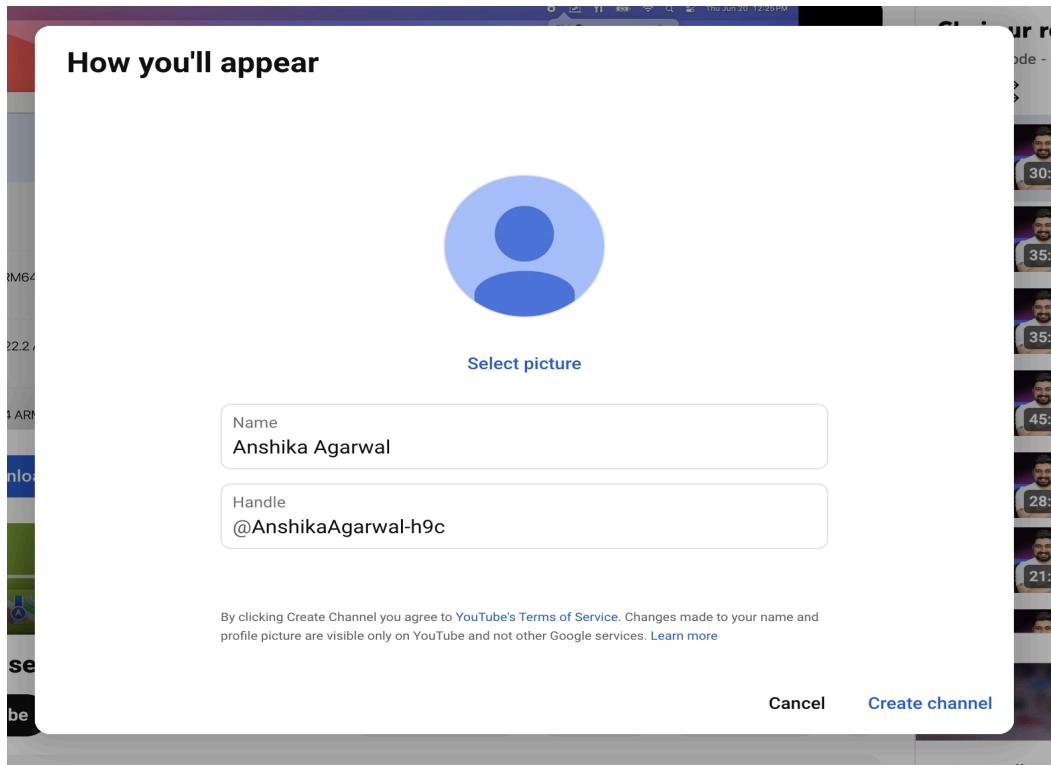
Jacob + Katie Schwarz 11M views • 2 years ago

Mount Titlis Cable Car | 3,020 metres in 30 mins | Switzerlan...

Reel_Vortex 290 views • 3 weeks ago

5. Channel Page:

- Option to create a channel only after the user is signed in.
- Display a list of videos which belong to that particular channel.
- Allow the user to edit or delete their videos.



Sample Channel Page



Internshala •

@InternshalaOfficial · 117K subscribers · 802 videos

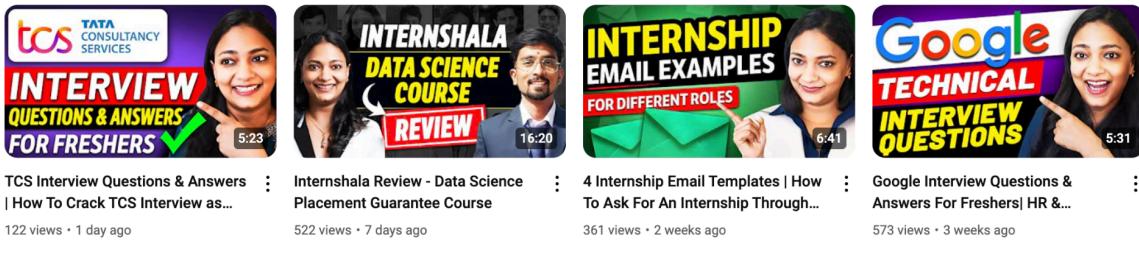
Welcome to official YouTube channel of Internshala. ...more

internshala.com and 5 more links

[Subscribe](#)

[Home](#) [Videos](#) [Shorts](#) [Live](#) [Playlists](#) [Community](#)

[Latest](#) [Popular](#) [Oldest](#)



TCS Interview Questions & Answers
| How To Crack TCS Interview as...

122 views · 1 day ago

Internshala Review - Data Science
Placement Guarantee Course

522 views · 7 days ago

4 Internship Email Templates | How
To Ask For An Internship Through...

361 views · 2 weeks ago

Google Interview Questions &
Answers For Freshers| HR &...

573 views · 3 weeks ago

EPISODE-42

EPISODE-41

EPISODE-40

6. Responsive Design:

- Ensure the application is fully responsive across devices.

Sample data for Channel:

```
{ channelId: "channel01", channelName: "Code with John", owner: "user01", description:  
"Coding tutorials and tech reviews by John Doe.", channelBanner:  
"https://example.com/banners/john_banner.png", subscribers: 5200, videos: ["video01",  
"video02"], }
```

Back-End (Node.js, Express)

1. API Endpoints:

- **User Authentication:**
 - Sign up, login, and token-based authentication using JWT.
- **Channel Management:**
 - API to create new channel and fetch any information from that channel.
- **Video Management:**
 - API to fetch, update, and delete videos.

- **Comments:**
 - API to add and fetch comments.
- 2. **Database (MongoDB):**
 - Store users, videos, channels and comments in MongoDB collections.
 - Store file metadata (e.g., video URL, thumbnail URL) in the database.

Technologies to Use:

- **Frontend:** React, React Router, Axios
- **Backend:** Node.js, Express.js, MongoDB
- **Authentication:** JWT (JSON Web Tokens)
- **Database:** MongoDB (MongoDB Atlas or local instance)
- **Version Control:** Git

Submission Requirements:

- Source code uploaded to a GitHub repository.
- A detailed README file explaining the project setup, features, and usage.
- A short video demo showcasing the application's features.

Rubrics for the YouTube Clone Assignment (Out of 500 Marks)

Category	Criteria	Marks
Front-End (React)		170
Home Page UI/UX	Header, Sidebar, Video Grid, Filters	40
User Authentication	Registration, JWT-based login, Sign-in functionality	40

Video Player Page	Video player, comments section, like/dislike buttons	50
Channel Page	Ability to create/edit/delete videos, channel page layout	40
Back-End (Node.js & Express)		120
API Design	Proper design of routes for authentication, video management, and comments	40
Data Handling (MongoDB)	Storing and fetching data correctly for users, videos, comments, and channels	40
JWT Integration	Secure JWT-based authentication and protected routes	40
Search & Filter Functionality		40
Search by Title	Working search functionality on the home page	20

Filter by Category	Proper filter implementation based on video category	20
Responsiveness		30
Mobile/Tablet/Desktop Layout	Proper layout and functionality across different screen sizes	30
Code Quality & Documentation		40
Code Structure	Proper folder structure, clean code, adherence to coding best practices	20
Documentation	Clear comments, README file explaining project setup and usage	20

Rubric Breakdown:

1. Front-End (React) – 200 Marks:

- **Home Page UI/UX (60 Marks):** Evaluate the completeness of the header, sidebar, video grid, filter buttons, and responsiveness on various devices.
- **User Authentication (50 Marks):** Ensure proper implementation of user registration, JWT-based login, and sign-in functionality.
- **Video Player Page (50 Marks):** Check for the functioning video player, comment management, and interactive buttons (like/dislike).
- **Channel Page (40 Marks):** Evaluate the user's ability to create, edit, and delete their videos and the proper channel page layout.

2. Back-End (Node.js & Express) – 150 Marks:

- **API Design (50 Marks):** Assess the routing and functionality for users, videos, and comments, ensuring best practices are followed.
- **Data Handling (MongoDB) (50 Marks):** Check how well the data is stored and retrieved, including video metadata, user info, comments, and channel details.
- **JWT Integration (50 Marks):** Ensure that JWT-based authentication is secure and all protected routes work correctly.

3. Search & Filter Functionality – 50 Marks:

- **Search by Title (25 Marks):** Verify the search bar functionality works correctly for searching videos by title.
- **Filter by Category (25 Marks):** Ensure category-based filters are functional and correctly implemented.

4. Responsiveness – 50 Marks:

- **Mobile/Tablet/Desktop Layout (50 Marks):** Test how well the app adapts across different screen sizes, ensuring all features are usable and the layout remains consistent.

5. Code Quality & Documentation – 50 Marks:

- **Code Structure (25 Marks):** Check for clean, well-organized code with proper folder structure, adhering to best practices.
- **Documentation (25 Marks):** Ensure the project is well-documented with clear comments and a README explaining the setup, usage, and purpose of the project.