Project Title:

Capstone Project: Develop a YouTube Clone Using the MERN Stack

Project Description

This project is a YouTube-like video sharing platform built using the MERN stack (MongoDB, Express, React, Node.js).
Users can authenticate, create channels, upload videos, browse videos by category, and interact through likes, dislikes, and comments.

The application follows RESTful API architecture, JWT-based authentication, and responsive UI design for mobile, tablet, and desktop devices.

Tech Stack

Frontend

- React.js
- Axios
- CSS (Responsive Design)

Backend

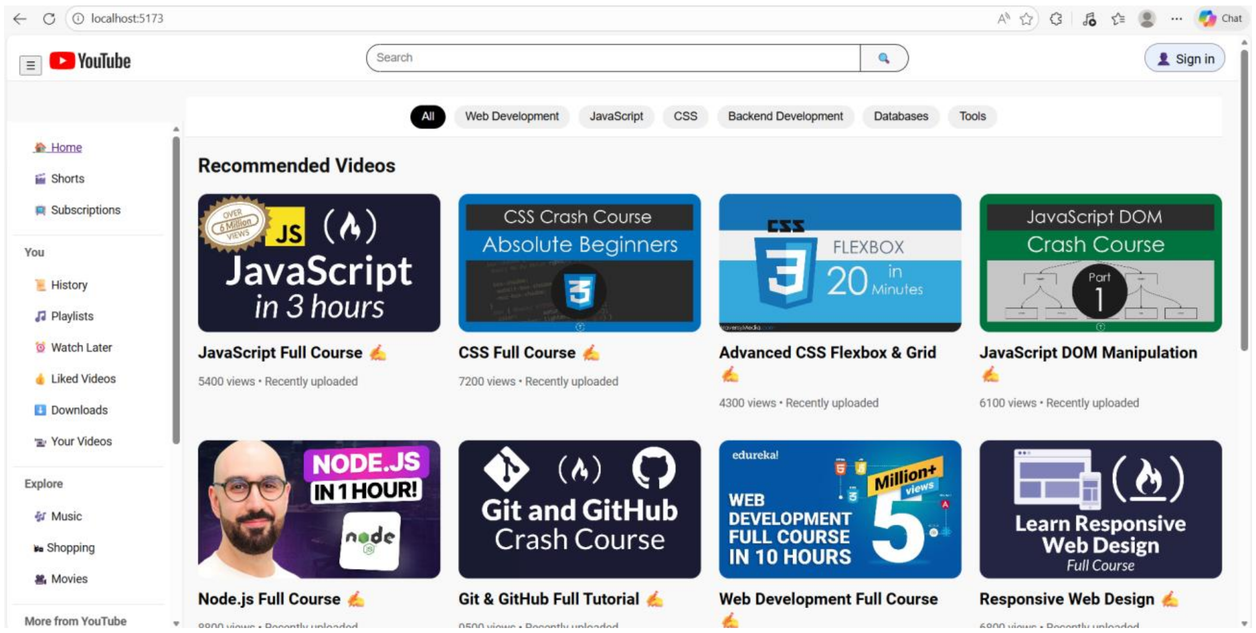
- Node.js
- Express.js
- MongoDB & Mongoose
- JWT Authentication

Tools

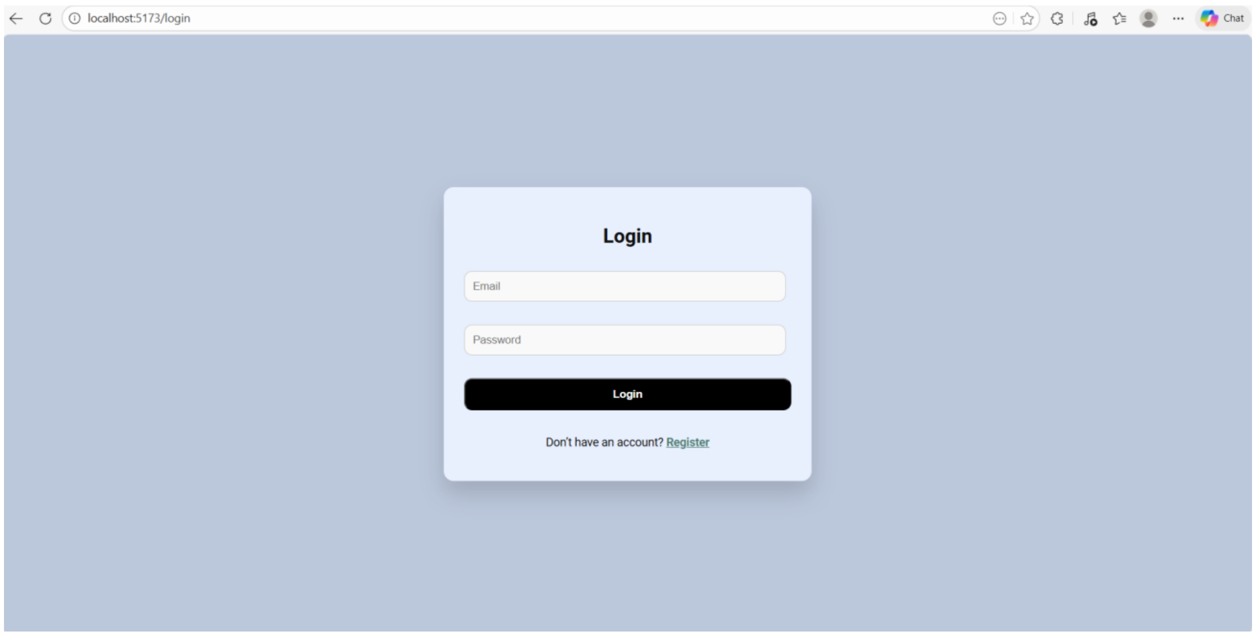
- Thunder Client (API testing)
- MongoDB Compass
- Git & GitHub

UI Screenshots:

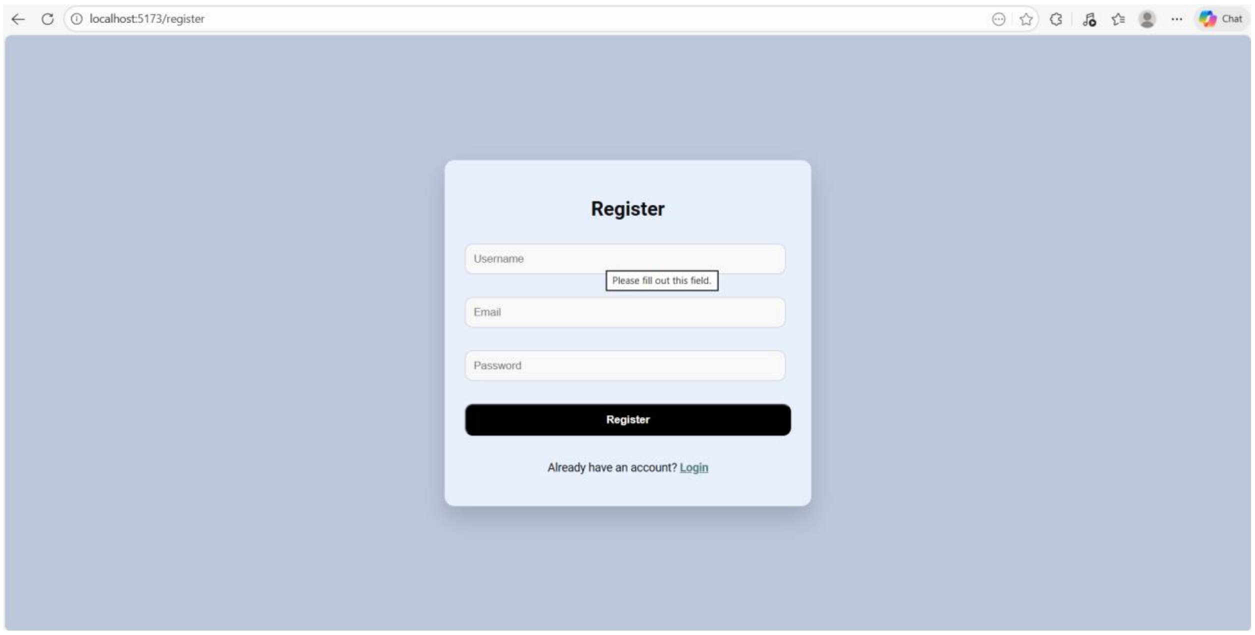
Homepage:



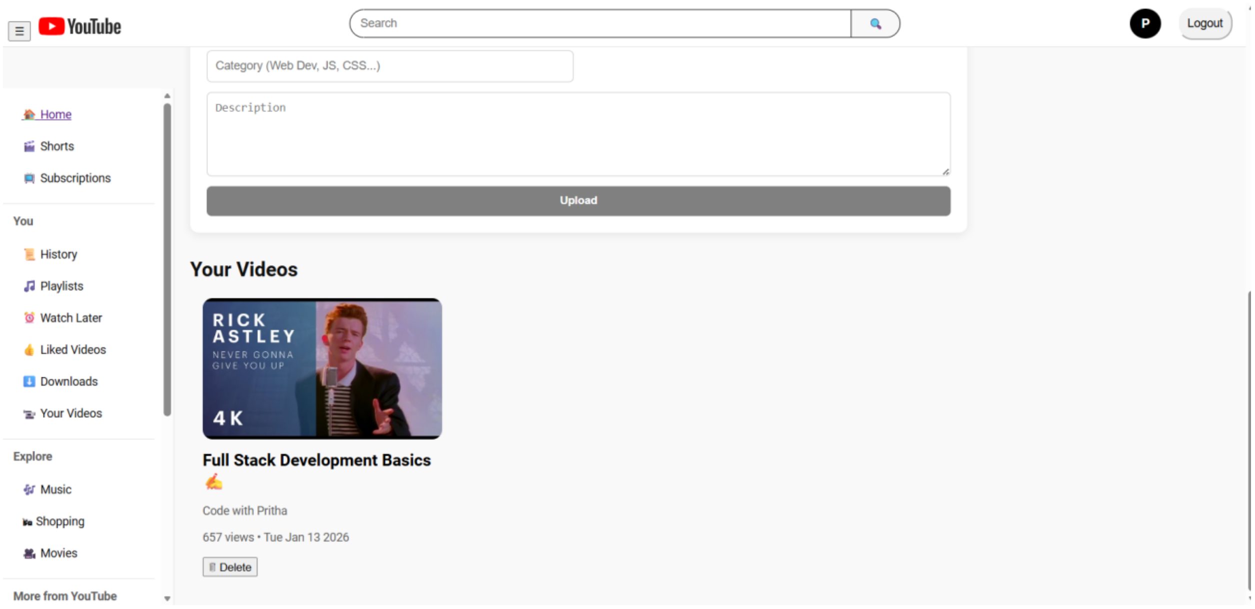
Login:



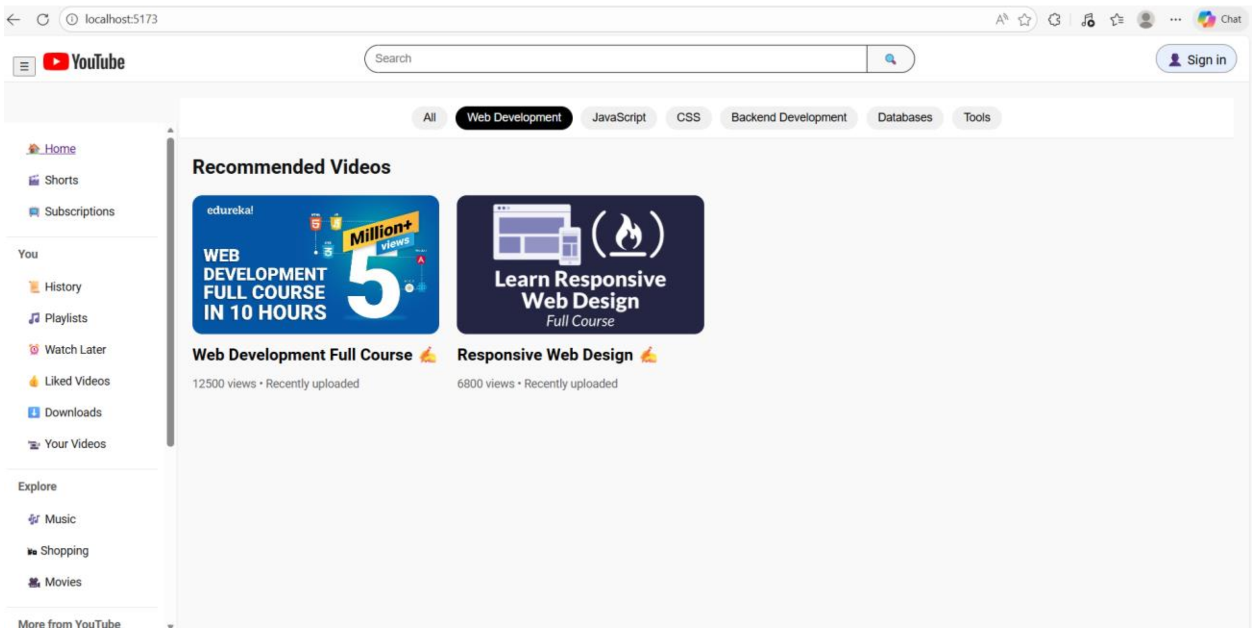
Register:



Uploaded Videos with Delete Button:



Filter by Category:

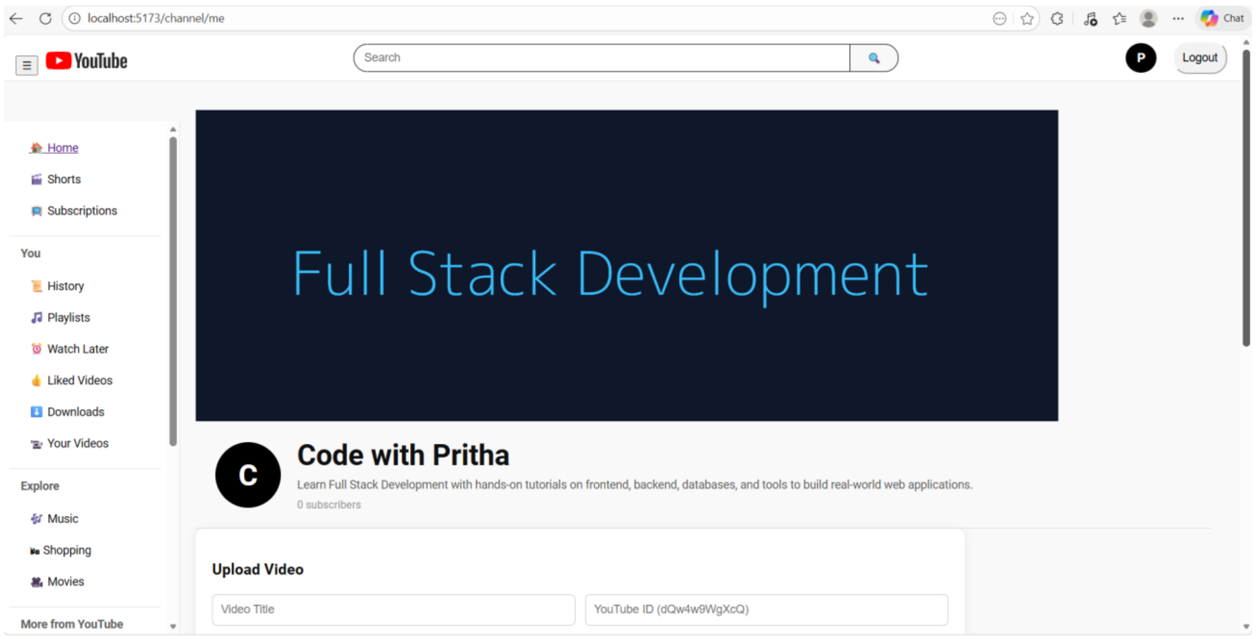


Recommended Videos

Web Development Full Course ✍️
12500 views • Recently uploaded

Responsive Web Design ✍️
6800 views • Recently uploaded

ChannelPage:



Code with Pritha

Learn Full Stack Development with hands-on tutorials on frontend, backend, databases, and tools to build real-world web applications.

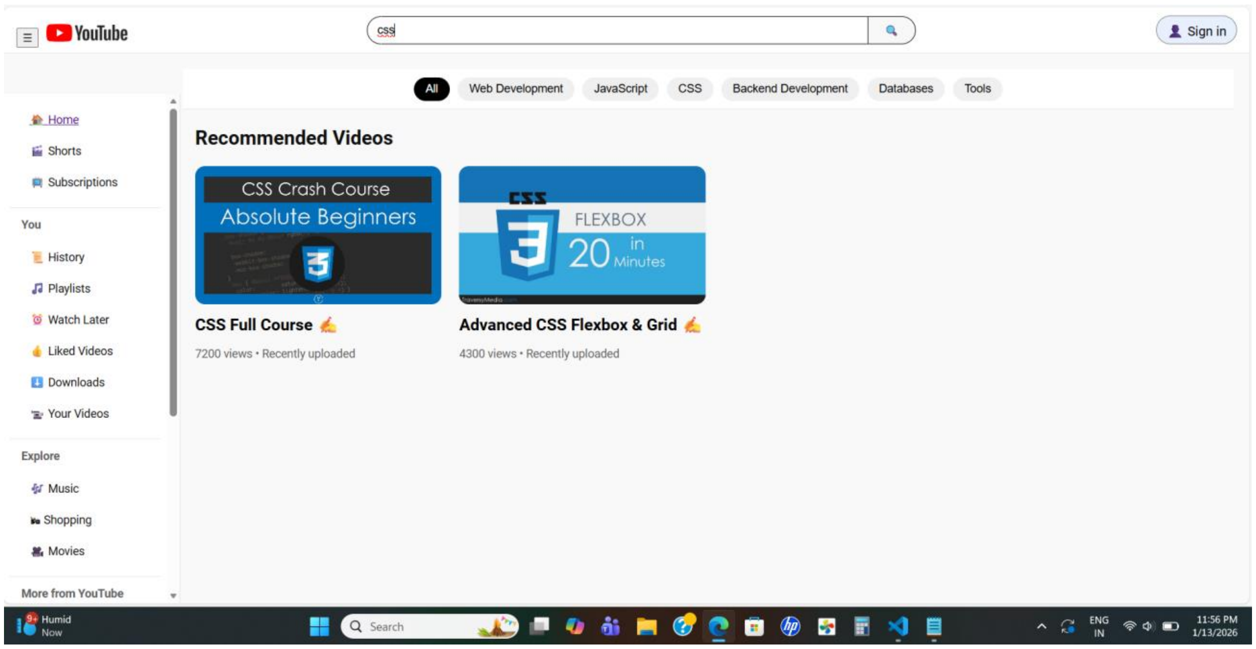
0 subscribers

Upload Video

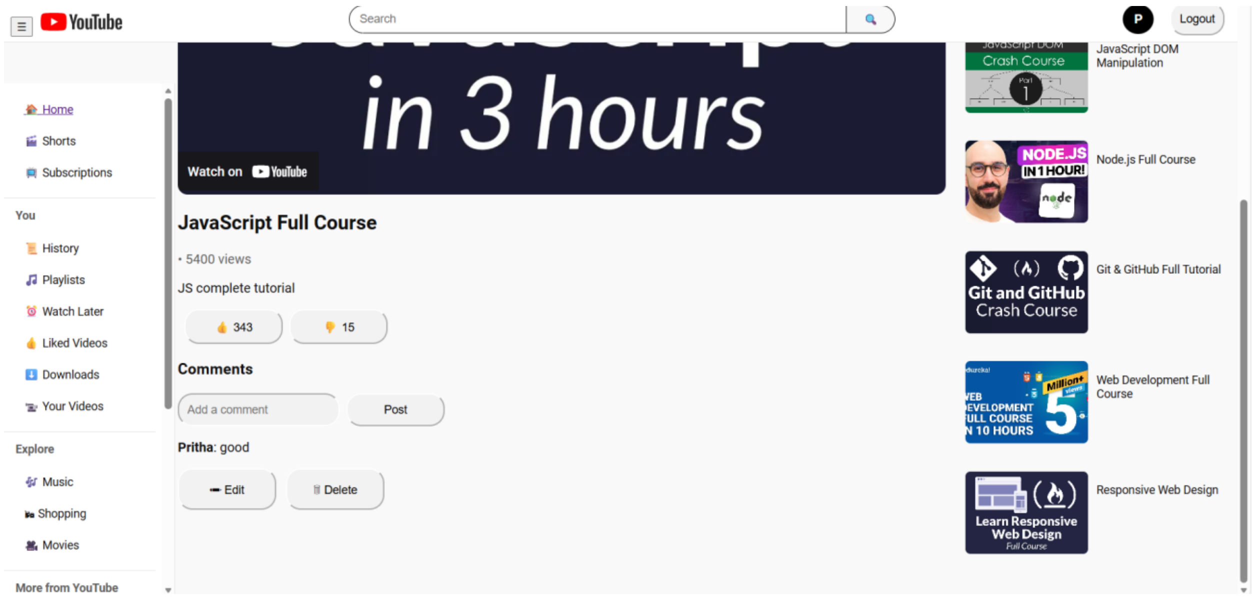
Video Title

YouTube ID (dQw4w9WgXcQ)

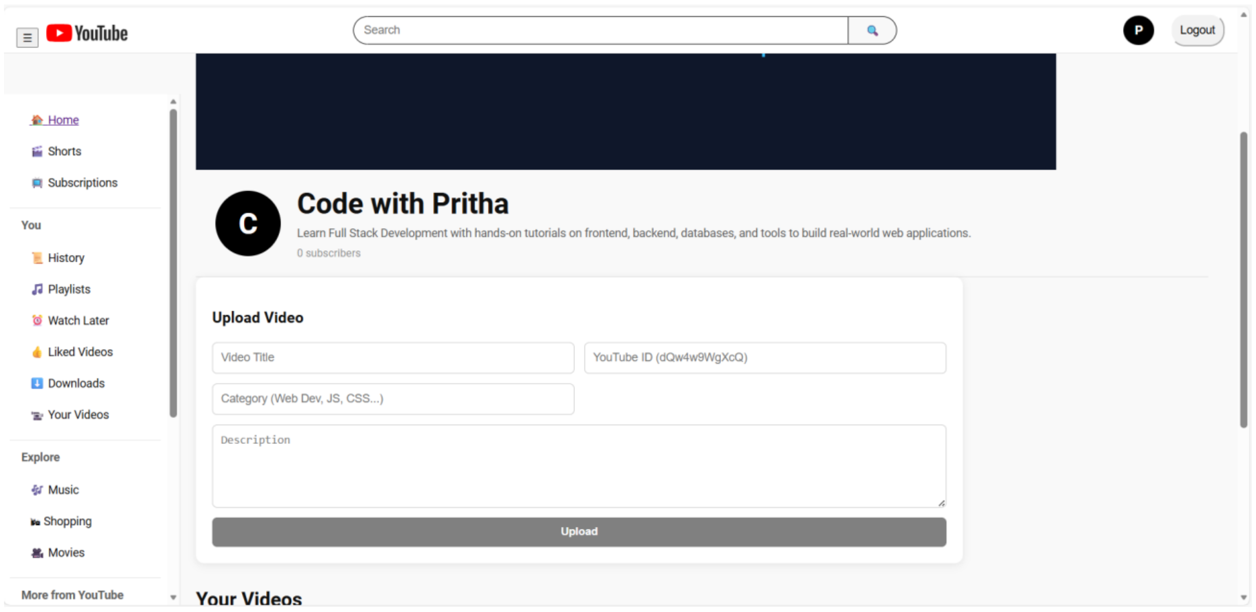
Search and filter:



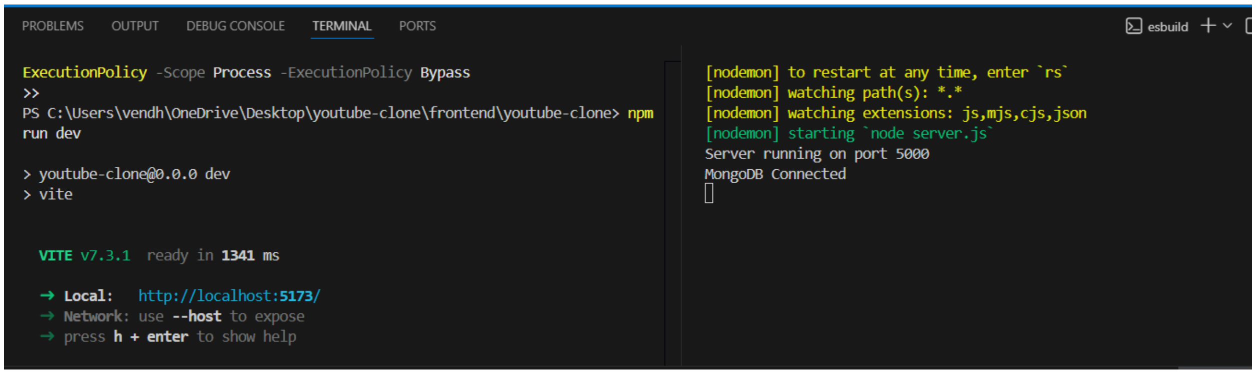
Like,Dislike and Comment:



Upload video banner:



Terminal:



```
ExecutionPolicy -Scope Process -ExecutionPolicy Bypass
>>
PS C:\Users\vendh\OneDrive\Desktop\youtube-clone\frontend\youtube-clone> npm
run dev

> youtube-clone@0.0.0 dev
> vite


  VITE v7.3.1  ready in 1341 ms

  ➜  Local:   http://localhost:5173/
  ➜  Network: use --host to expose
  ➜  press h + enter to show help
```

```
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `node server.js`
Server running on port 5000
MongoDB Connected
```

Description:

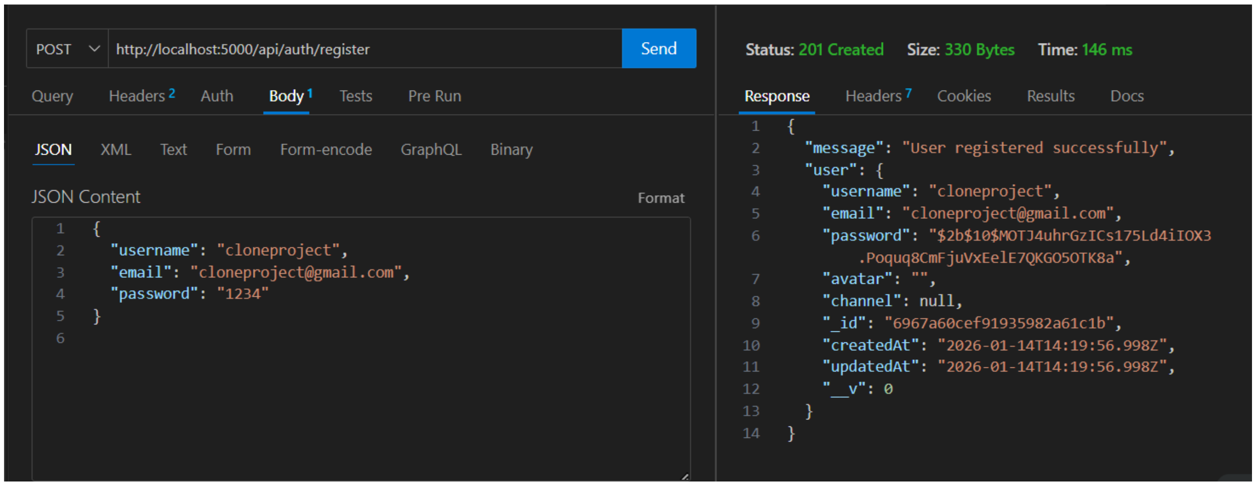
My Backend and Frontend is running Successful.

API Testing (Thunder Client)

All APIs were tested using **Thunder Client** inside VS Code.
Screenshots are included for verification.

User Registration API

Endpoint:
POST /api/auth/register



http://localhost:5000/api/auth/register

```
{
  "username": "cloneproject",
  "email": "cloneproject@gmail.com",
  "password": "1234"
}
```

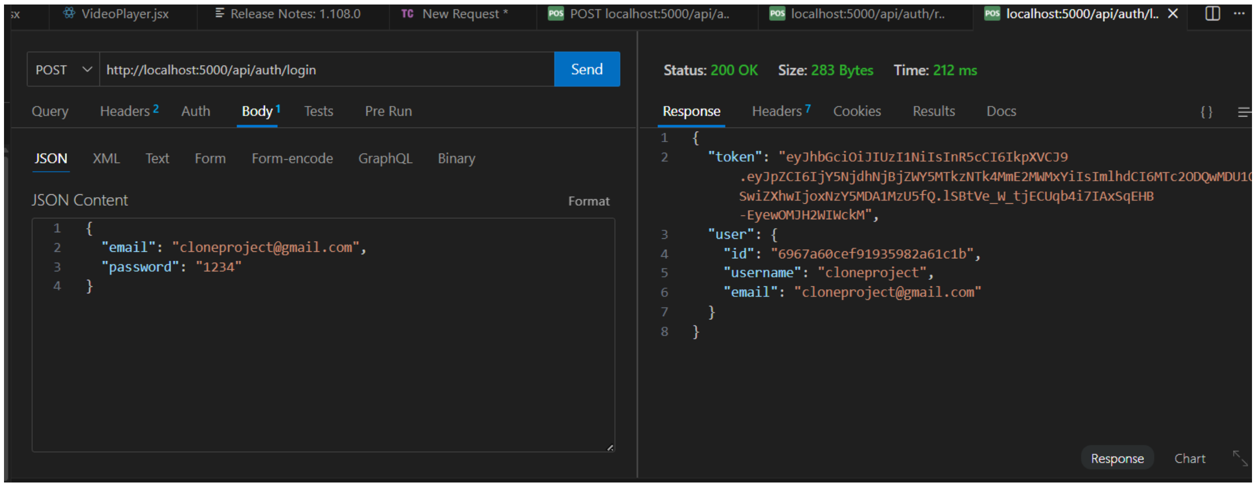
Status: 201 Created Size: 330 Bytes Time: 146 ms

```
{
  "message": "User registered successfully",
  "user": {
    "username": "cloneproject",
    "email": "cloneproject@gmail.com",
    "password": "$2b$10$MOTJ4uhrGzICs175Ld4iIOX3
        .Poquq8CmFjuVxEelE7QKGO5OTK8a",
    "avatar": "",
    "channel": null,
    "_id": "6967a60cef91935982a61c1b",
    "createdAt": "2026-01-14T14:19:56.998Z",
    "updatedAt": "2026-01-14T14:19:56.998Z",
    "__v": 0
  }
}
```

Description:
Registers a new user by providing name, email, and password.

User Login API

Endpoint:
POST /api/auth/login



http://localhost:5000/api/auth/login

```
{
  "email": "cloneproject@gmail.com",
  "password": "1234"
}
```

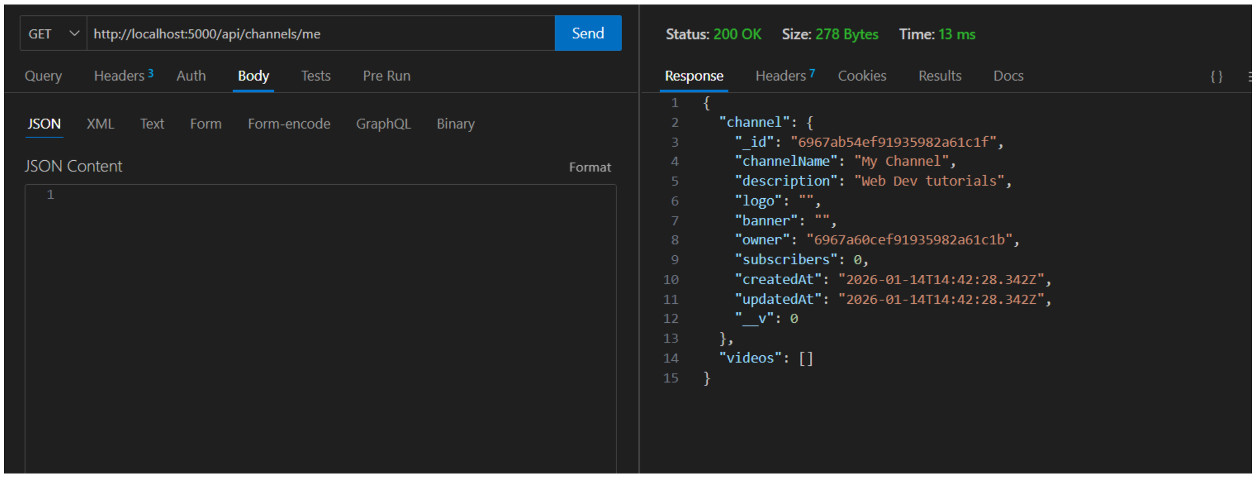
Status: 200 OK Size: 283 Bytes Time: 212 ms

```
{
  "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9
      .eyJpZCI6IjY5NjdhNjBjZWY5MTkzNTk4MmE2MWMxYiIsImlhdCI6MTc2ODQwMDU1
      SwiZXhwIjoxNzY5MDA1MzU5fQ.lSBtVe_W_tjECUqb4i7IAxSqEHB
      -EyewOMJH2WIWckM",
  "user": {
    "id": "6967a60cef91935982a61c1b",
    "username": "cloneproject",
    "email": "cloneproject@gmail.com"
  }
}
```

Description:
Authenticates the user and returns a JWT token used for protected routes.

Get My Channel

Endpoint:
GET /api/channels/me



http://localhost:5000/api/channels/me

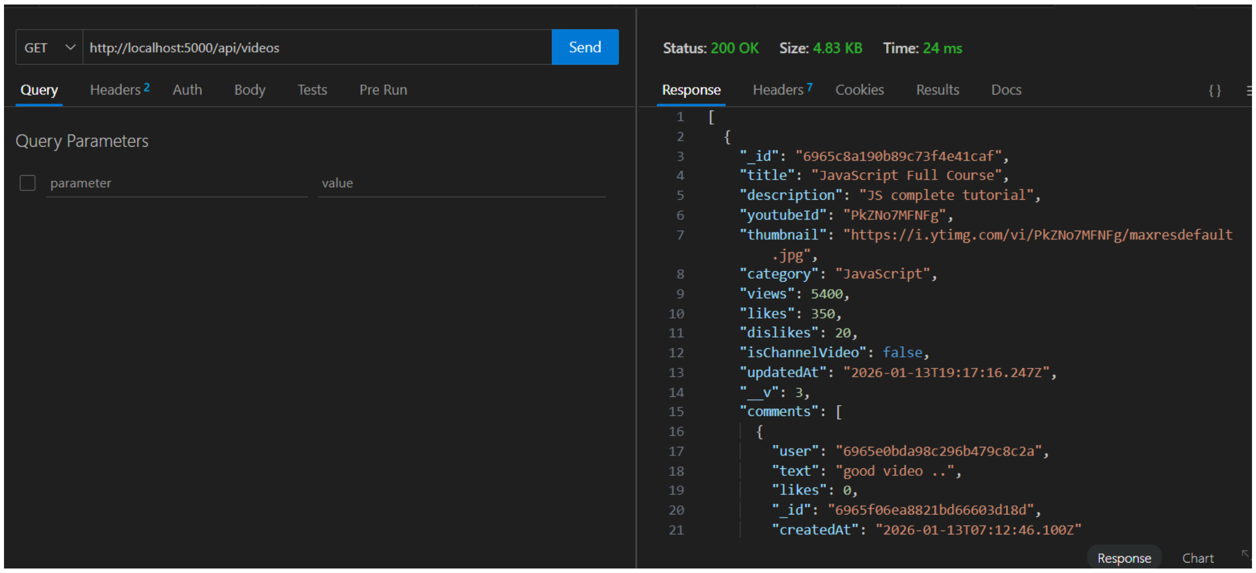
Status: 200 OK Size: 278 Bytes Time: 13 ms

```
{
  "channel": {
    "_id": "6967ab54ef91935982a61c1f",
    "channelName": "My Channel",
    "description": "Web Dev tutorials",
    "logo": "",
    "banner": "",
    "owner": "6967a60cef91935982a61c1b",
    "subscribers": 0,
    "createdAt": "2026-01-14T14:42:28.342Z",
    "updatedAt": "2026-01-14T14:42:28.342Z",
    "__v": 0
  },
  "videos": []
}
```

Description:
Fetches the logged-in user’s channel and uploaded videos.

GET Uploaded Video API

Endpoint:
GET/api/videos



http://localhost:5000/api/videos

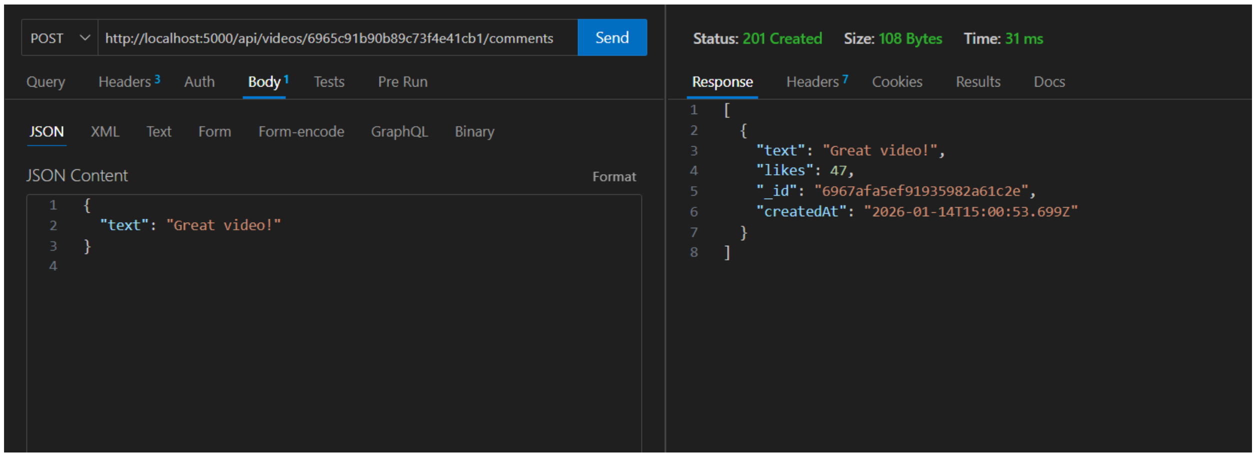
Status: 200 OK Size: 4.83 KB Time: 24 ms

```
[
  {
    "_id": "6965c8a190b89c73f4e41caf",
    "title": "JavaScript Full Course",
    "description": "JS complete tutorial",
    "youtubeId": "PkZNo7MFNFg",
    "thumbnail": "https://i.ytimg.com/vi/PkZNo7MFNFg/maxresdefault
        .jpg",
    "category": "JavaScript",
    "views": 5400,
    "likes": 350,
    "dislikes": 20,
    "isChannelVideo": false,
    "updatedAt": "2026-01-13T19:17:16.247Z",
    "__v": 3,
    "comments": [
      {
        "user": "6965e0bda98c296b479c8c2a",
        "text": "good video ..",
        "likes": 0,
        "_id": "6965f06ea8821bd66603d18d",
        "createdAt": "2026-01-13T07:12:46.100Z"
```

Description:
Get Uploaded video under the authenticated user's channel.
Channel ownership is derived from JWT (secure design).

Add Comment

Endpoint:
POST /api/comments/:videoId



```
http://localhost:5000/api/videos/6965c91b90b89c73f4e41cb1/comments
```

```
{
  "text": "Great video!"
}
```

Status: 201 Created   Size: 108 Bytes   Time: 31 ms

```
[
  {
    "text": "Great video!",
    "likes": 47,
    "_id": "6967afa5ef91935982a61c2e",
    "createdAt": "2026-01-14T15:00:53.699Z"
  }
]
```

Description:
Adds a comment to a specific video.

Database (MongoDB)

- MongoDB Compass used locally
- Sample data exported and included in: db-exports/

Responsive Design

The application adapts to:

- Mobile screens
- Tablet screens
- Desktop screens

Media queries ensure layout consistency across devices.

Application Features

- User Authentication (JWT)
- Channel creation & management
- Video upload & listing
- Like / Dislike system
- Comment add / edit / delete
- Responsive UI (Mobile / Tablet / Desktop)
- Secure API access using middleware

Git & Commits

- Frontend and Backend developed separately
- Multiple meaningful commits
- Clear commit messages
- GitHub repository maintained properly