A full-featured YouTube clone built with the MERN (MongoDB, Express.js, React, Node.js) stack, featuring user authentication, video management, and channel functionality.

Git URL

https://github.com/SravanGunaganti/youtube-clone

Demo Link

Demo video Link

🚀Features

User Authentication

- JWT-based authentication
- User registration and login
- Profile management with avatar
- Protected routes

Video Management

- Video upload with metadata (title, description, thumbnail)
- YouTube-style video player
- Video editing and deletion (owner only)
- View count tracking
- Like/Dislike functionality

Channel System

- Channel creation and customization
- Banner and avatar upload with preview
- Subscribe/Unsubscribe functionality
- Channel statistics (subscribers, videos)

Comments System

- Add, edit, and delete comments
- Like/Dislike comments
- Authentication required for posting

Technology Stack

Backend

- **Node.js** - Runtime environment

- **Express.js** - Web framework
- **MongoDB** - Database
- **Mongoose** - ODM for MongoDB
- **JWT** - Authentication
- **Bcrypt** - Password hashing
- **CORS** - Cross-origin requests
- **Dotenv** - Environment variables

Frontend

- **React** - UI library
- **Vite** - Build tool
- **React Router** - Client-side routing
- **Tailwind CSS** - Styling
- **React Icons** - Icons
- **Axios** - HTTP client
- **React Toastify** - Notifications

Project Structure

```
└── 📁youtube-mern
    └── 📁client
        └── 📁public
            ├── favicon.ico
            ├── vite.svg
        └── 📁src
            └── 📁assets
                └── 📁youtube-icons
                └── 📁youtube-logos
            └── 📁components
                └── 📁channel
                    ├── ChannelInfo.jsx
                    ├── CreateChannelModal.jsx
                    ├── NoChannelPrompt.jsx
                    ├── SignInPrompt.jsx
                    ├── TabsContent.jsx
                    ├── UpdateChannelModal.jsx
                    ├── UpdateVideoModal.jsx
                    ├── UploadVideoModal.jsx
                ├── ChannelNotFound.jsx
                ├── ChannelPage.jsx
                ├── CommentCard.jsx
                ├── ConfirmModal.jsx
                ├── EditProfileModal.jsx
                ├── FilterButtons.jsx
                ├── Header.jsx
```

```
            ├── Home.jsx
            ├── Loader.jsx
            ├── MiniSidebar.jsx
            ├── Overlay.jsx
            ├── PageNotFound.jsx
            ├── RootErrorBoundary.jsx
            ├── Sidebar.jsx
            ├── SignIn.jsx
            ├── SignUp.jsx
            ├── UserProfileDropdown.jsx
            ├── VideoCard.jsx
            ├── VideoGrid.jsx
            ├── VideoNotFound.jsx
            ├── VideoPlayer.jsx
        └── 📁constants
            ├── categories.js
            ├── sidebarLinks.js
        └── 📁context
            ├── AuthContext.jsx
        └── 📁helpers
            ├── useActivePath.js
        └── 📁services
            ├── api.js
        └── 📁utils
            ├── authUtils.js
            ├── utilityFunctions.js
            ├── validators.js
        ├── App.jsx
        ├── index.css
        ├── main.jsx
        ├── RouterWrapper.jsx
    ├── .env
    ├── eslint.config.js
    ├── index.html
    ├── package-lock.json
    ├── package.json
    ├── vite.config.js
└── 📁server
    └── 📁controllers
        ├── channel.controller.js
        ├── comment.controller.js
        ├── user.controller.js
        ├── video.controller.js
    └── 📁middlewares
        ├── authenticateToken.js
        ├── errorHandlers.js
```

```
        ├── validateObjectId.js
        ├── validateUser.js
    └── 📁models
        ├── Channel.model.js
        ├── Comment.model.js
        ├── User.model.js
        ├── Video.model.js
    └── 📁routes
        ├── channel.routes.js
        ├── comment.routes.js
        ├── user.routes.js
        ├── video.routes.js
    └── 📁utils
        ├── generateToken.js
        ├── sendErrorResponse.js
        ├── sendSuccessResponse.js
        ├── validators.js
    ├── .env
    ├── index.js
    ├── package-lock.json
    ├── package.json
    ├── seedData.js
├── .gitignore
└── README.md
```

🚀Getting Started

Prerequisites

- Node.js
- MongoDB (local or Atlas)
- npm or yarn

Installation

1. **Clone the repository**

```
git clone https://github.com/SravanGunaganti/youtube-clone
cd youtube-clone
```

2. **Set up environment variables**
3. Create `.env` file in the server directory:

```
PORT=5050
MONGODB_URI=your_mongodb_connection_string
JWT_SECRET=your_jwt_secret
```

4. For the client, create `.env` in the client directory:

```
VITE_API_URL=http://localhost:5050/api
```

5. **Install dependencies**

```
# Backend
cd server
npm install
# Frontend
cd ../client
npm install
```

6. **Run the application**
7. Backend:

```
cd server
npm run dev
```

   - Seed sample data:

```
npm run seed
```

8. Frontend:

```
cd client
npm run dev
```

9. **Access the application**
10. Frontend: http://localhost:5173
11. Backend API: http://localhost:5050/api

🚲UI/UX Features

- YouTube-inspired interface with professional styling
- Responsive design for all screen sizes
- Consistent color palette and typography
- VideoPlayer: Custom video controls, like/dislike, comments

- ChannelPage: Channel management, video grid, statistics, subscription, owner-only actions
- Navbar: Search, profile dropdown, edit profile popup
- Sidebar: Navigation menu
- VideoCard: Thumbnails, metadata, hover effects
- Home: Video grid, filter buttons, skeleton loading with hover play

🚚Security Features

- JWT Authentication
- Password Hashing using bcrypt
- Input Validation
- CORS Configuration
- Protected Routes
- Owner-only actions for management

API Documentation

Authentication

- `POST /api/auth/register`
- `POST /api/auth/login`
- `GET /api/auth/verify`

Users

- `PUT /api/users/profile`

Channels

- `GET /api/channels/:channelId`
- `POST /api/channels`
- `PUT /api/channels/:channelId`
- `GET /api/my-channel`
- `GET /api/channels/:channelId/exist`
- `DELETE /api/channels/:channelId`
- `POST /api/channels/:channelId/subscribe`
- `GET /api/channels/:channelId/subscription-status`

Videos

- `GET /api/videos`
- `GET /api/videos/:videoId`
- `POST /api/videos`
- `GET /api/videos/:videoId/exist`
- `PUT /api/videos/:videoId`
- `DELETE /api/videos/:videoId`
- `PUT /api/videos/:videoId/like`

- `GET /api/videos/:videoId/like-status`
- `/api/channels/:channelId/videos`

Comments

- `GET /api/comments/video/:videoId`
- `POST /api/comments`
- `PUT /api/comments/:id`
- `DELETE /api/comments/:id`
- `PUT /api/comments/:id/like`
- `GET /api/comments/:id/like-status`

Development Scripts

Backend

- `npm run dev`
- `npm start`
- `npm run seed`

Frontend

- `npm run dev`
- `npm run build`
- `npm run preview`

 Performance Optimizations

- Lazy Loading
- Code Splitting