YouTube Clone - MERN Stack Project Documentation

NAME : Linganandda

1. Introduction

Project Overview

This project is a YouTube Clone built using the MERN (MongoDB, Express.js, React.js, Node.js) stack. The application allows users to view, upload, interact with videos, and manage their own channels.

Objectives

- Develop a full-stack web application that mimics YouTube's core features.
- Implement secure authentication using JWT.
- Enable video uploading, viewing, commenting, and interaction.
- Ensure smooth API interactions with MongoDB for data storage.
- Provide a responsive and user-friendly UI.

Technologies Used 2.1 Frontend

- **React.js** - UI library for building interactive components.
- **React Router** - Handles routing within the application.
- **Axios** - For API requests.
- **CSS/Tailwind** - Styling for UI components.

2.2 Backend

- **Node.js** - JavaScript runtime for backend processing.
- **Express.js** - Backend framework for handling API requests.

- **MongoDB** - NoSQL database for storing user and video data.
- **Mongoose** - ODM for MongoDB to manage data schema.
- **JWT** - Secure authentication for user login.

API Endpoints User Authentication

- POST /register - Register a new user.
- POST /login - User login, returns a JWT token. **Channel Management**
- POST /channel - Create a new channel.
- GET /channel/:id - Get channel details by ID. **Video Management**
- POST /videos - Upload a new video.
- GET /videos/:id - Get video details.
- DELETE /videos/:id - Delete a video. **Comments**
- POST /comments - Add a comment.
- GET /comments/:videoId - Get comments for a video.

. User Authentication (JWT)

- **Login** and **Register** API securely authenticate users.
- JWT token is stored and used for secure access to protected routes.
- Middleware protects API routes requiring authentication.

. Frontend Implementation 7.1 React Components

- **HomePage** - Displays video grid.
- **VideoPlayer** - Video playback and comments.
- **AuthForm** - Handles login/register.

- **ChannelPage** - Shows channel info and videos.

Search & Filter Functionality

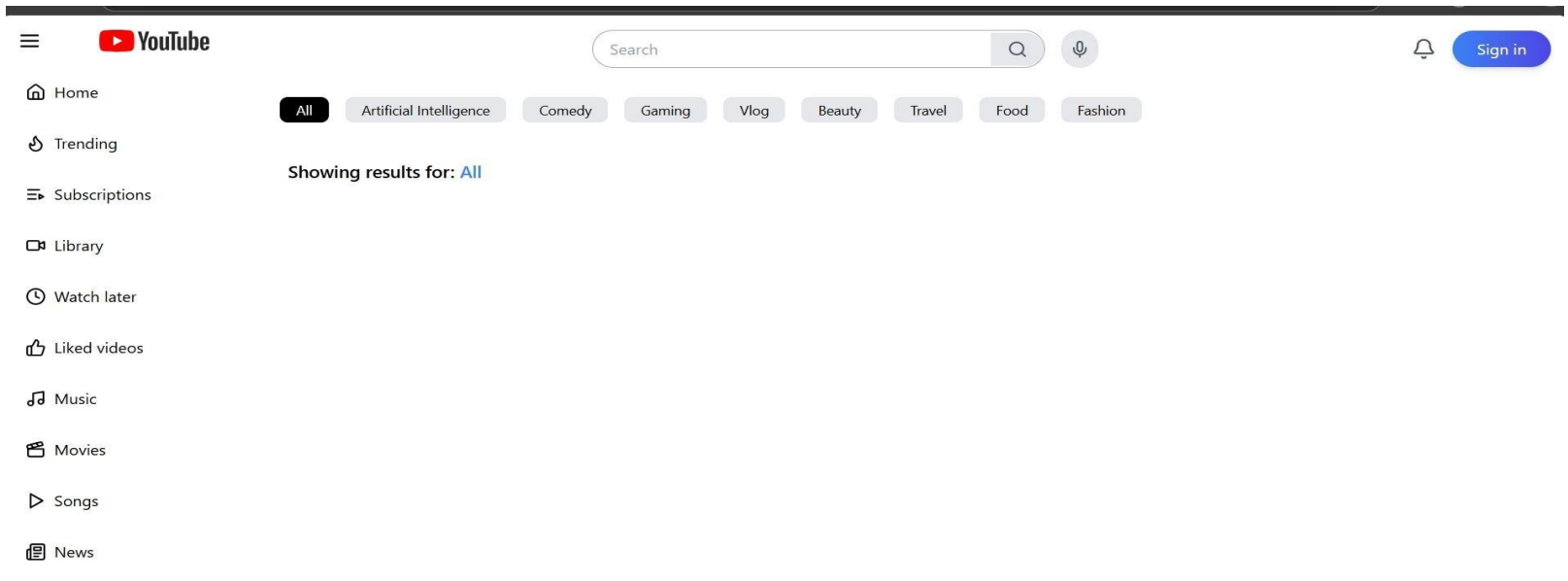
- **Search by title**: Implemented using a search bar.
- **Filter by category**: Allows users to filter videos based on categories.

Responsiveness

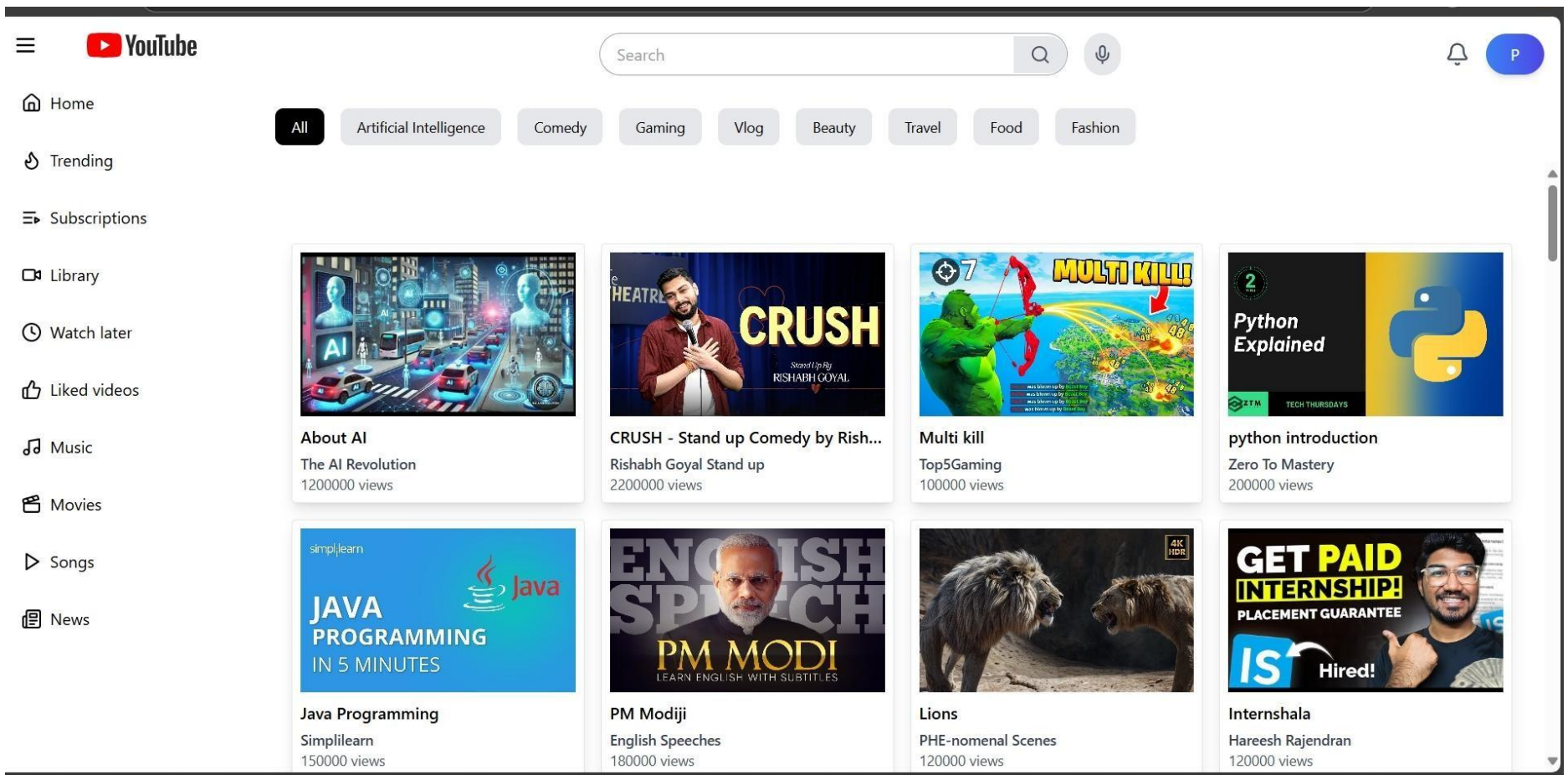
- UI adapts to **mobile, tablet, and desktop** screens.
- **CSS Flexbox and Grid** used for layout.
- Media queries ensure elements resize properly.

PROJECT IMAGES

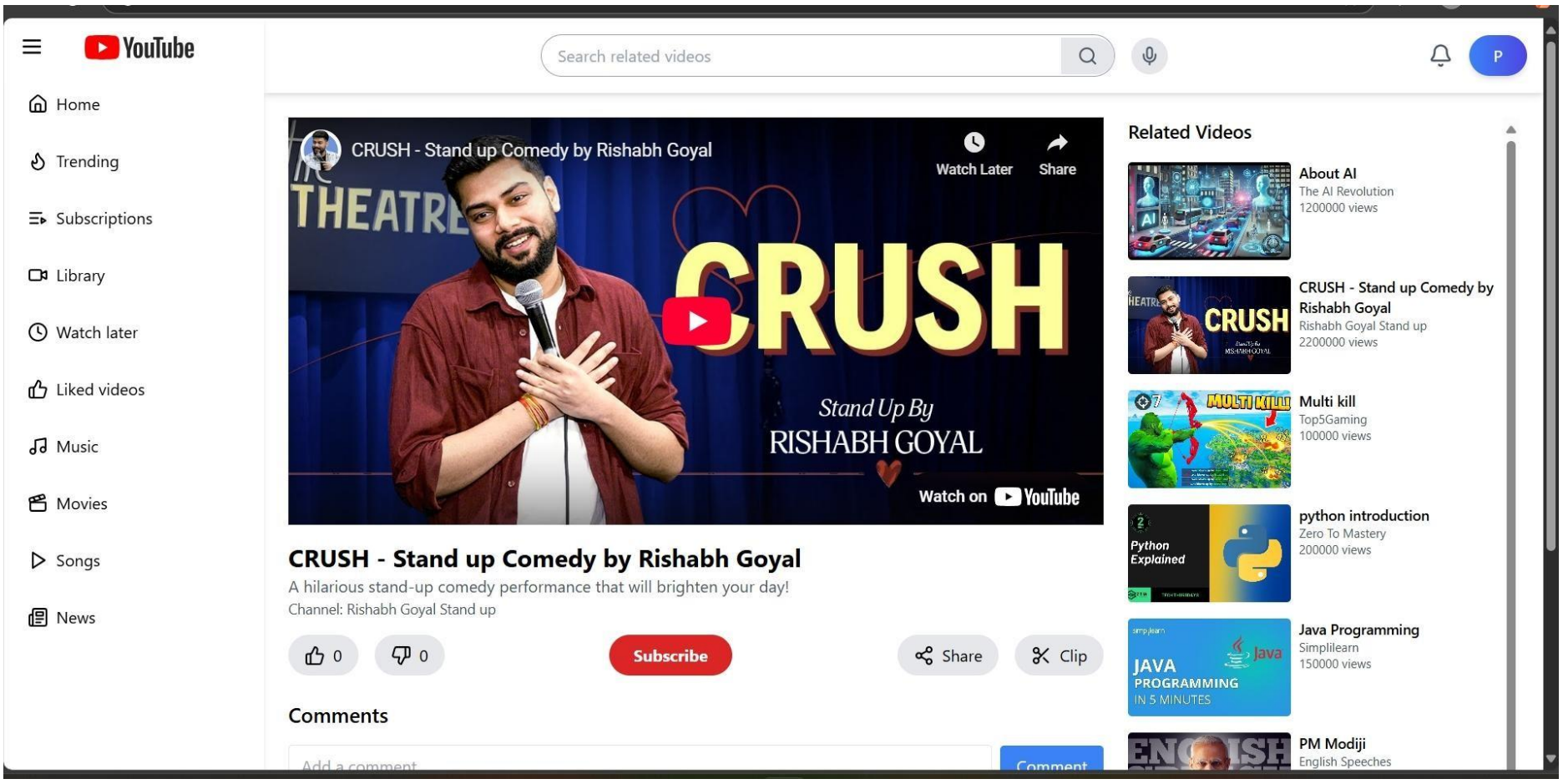
1.SIGNIN PAGE



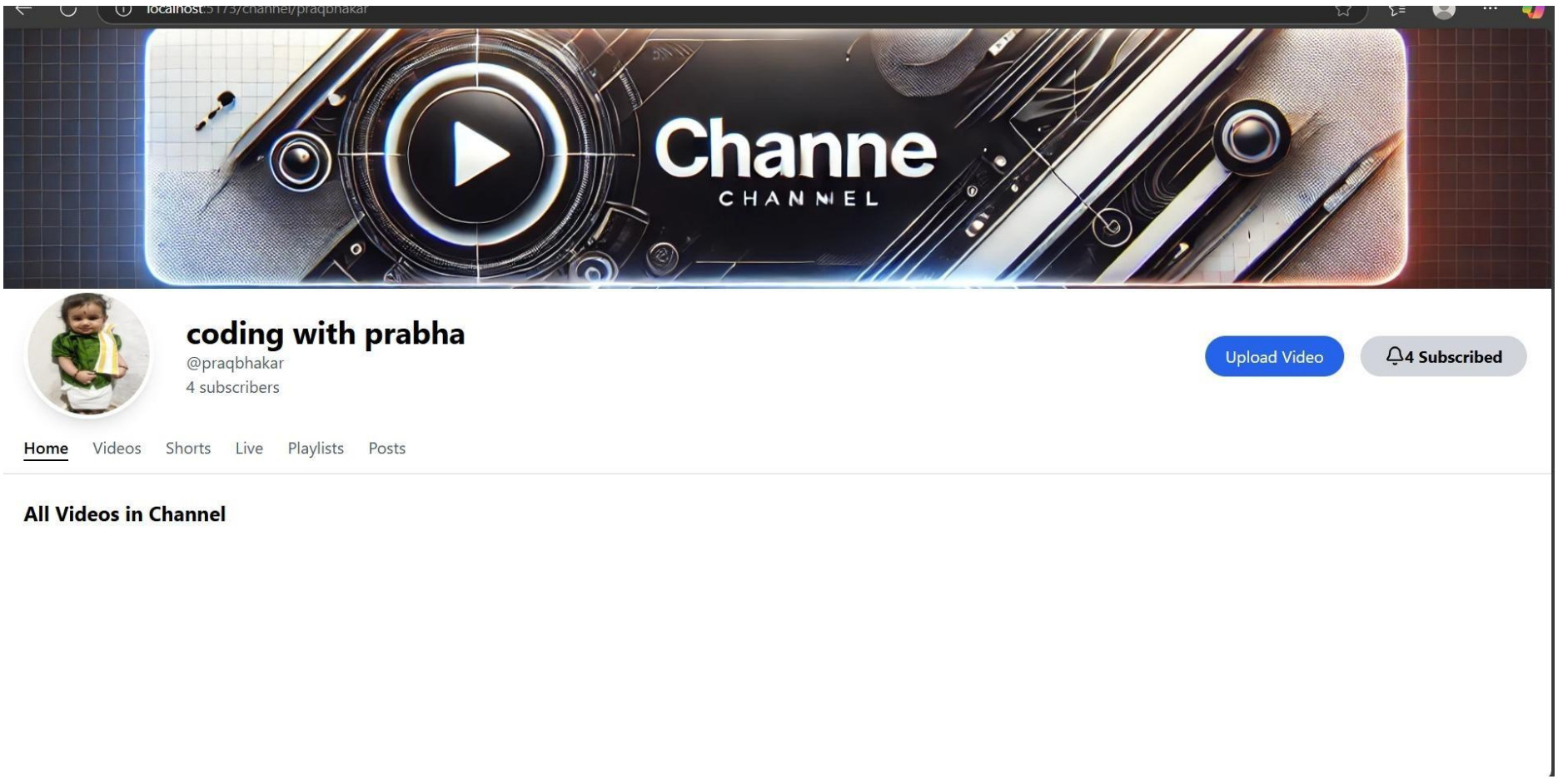
2.HOMEPAGE



3.VIDEOPLAYER PAGE



4.CHANNEL PAGE



Conclusion

- The YouTube Clone successfully integrates **frontend, backend, authentication, and database**.
- Users can **sign up, upload videos, comment, and interact** with content.
- Future improvements: **Live streaming, advanced recommendation system, and notifications**.