Project Overview

Project Title: CineCritic+ – A Secure Movie Rating Management System

Overview:

CineCritic+ is a secure backend-powered web system that allows users to register, log in, and manage a movie catalog through RESTful APIs. The system supports JWT-based authentication and includes CRUD operations for movies, category-based filtering, and top-rated movie retrieval.

Project Objectives

- Build a secure and scalable movie rating management API.
- Implement JWT authentication for protected routes.
- Ensure structured code using MVC principles.
- Enable CRUD operations and movie filtering.

Target Audience

- Movie reviewers and content creators.
- Developers integrating movie data.
- Film enthusiasts managing private collections.

Technology Stack

Component	Technology
Backend	Node.js, Express.js
Database	MongoDB
Authentication	JWT (jsonwebtoken)
Middleware	Logger, Validator, Auth, ErrorHandler
UI	HTML, CSS, Javascript
Testing	Postman

Functional Modules

- Authentication Module – Handles user registration, login, and JWT-based access.
- Movie Management Module – Performs CRUD operations on movies.
- Filtering Module – Filters movies by category or top ratings.
- Middleware & Security Module – Includes logger, validation, auth verification, and error handling.
- Database Module – Manages MongoDB connection, schemas, and data storage.
- Frontend Module – Provides simple HTML/CSS forms to interact with the backend.

API Endpoints

Authentication Endpoints

Method	Endpoint	Description
POST	/api/auth/register	Register a new user
POST	/api/auth/login	Login user and return JWT token

Movie Management Endpoints *(JWT Protected)*

Method	*Endpoint*	*Description*
GET	*/api/movies*	*Fetch all movies*
GET	*/api/movies/:id*	*Fetch movie by ID*
POST	*/api/movies*	*Add a new movie*
PUT	*/api/movies/:id*	*Update movie details*
DELETE	*/api/movies/:id*	*Delete a movie*

Movie Filter & Analytics Endpoints

Method	*Endpoint*	*Description*
GET	*/api/movies/category/:category*	*Filter movies by category*
GET	*/api/movies/top-rated*	*Get movies with rating ≥ 8.5*

Roles and Responsibilities

UI Dev - Tejash Raju K V
Backend Lead - Manoj C
Auth Specialist - Sankalp
Tester - Rohan S

Project Timeline and Expected Deliverables

Day	Backend Lead	Auth Specialist	UI Developer	Tester / Documentation
Day 1 – Setup & Authentication	- Initialize Node.js + Express structure - Setup MongoDB connection - Create models folder & config files	- Design User Schema - Implement `/register` & `/login` routes with JWT	- Design HTML/CSS layout (Register & Login forms)	- Prepare Postman workspace - Verify DB connection - Start Project Scope document
Day 2 – Movie CRUD & Middleware	- Create Movie Schema - Build CRUD routes (`GET`, `POST`, `PUT`, `DELETE`)	- Apply `verifyToken` middleware - Handle protected routes	- Create AddMovie & UpdateMovie pages - Style basic interface	- Test Auth + CRUD routes in Postman - Log bugs / errors - Update Scope progress
Day 3 – UI Integration, Testing & Docs	- Final backend polish - Implement filter routes (`/category`, `/top-rated`)	- Refine JWT validation flow - Debug token expiry & protection	- Build ViewMovies page - Connect UI with backend (Fetch API)	- Complete Postman Collection export - Write Reflection Answers - Finalize PDFs for submission