Project Title: Movie Management System

Project Overview

The objective of this project is to develop a complete Movie Management System where movies can be added, stored, updated, deleted, and retrieved from the database. The application must support uploading and displaying movie poster images, and it should include a search feature based on movie title. The project must include both backend and frontend development.

Functional Requirements

1. The system must allow adding a new movie with the following details:

- Movie Title

- Description

- Genre

- Release Year

- Movie Poster (Image)

2. The system must allow retrieving all movies from the database.

3. The system must allow retrieving a specific movie using its ID.

4. The system must allow updating an existing movie record, including updating the poster image if required.

5. The system must allow deleting an existing movie record, including deleting the stored poster file from the server.

6. The system must support searching movies by title. Search must be non-case-sensitive and return partial matches.

7. Movie poster images must be stored in the server’s folder and the database should store only the filename or relative path.

8. The server must provide public access to the uploaded poster images so they can be displayed in the frontend UI.

Technical Requirements – Backend

1. Development must be done using Node.js and Express.

2. MongoDB must be used as the database with Mongoose for schema and model creation.

3. Multer package must be used to upload and manage movie poster image files.

4. API routes must be created for all CRUD operations and search functionality.

5. Folder structure must be modular and well organized (routes, controllers, models, config, middleware).

6. All APIs must return proper JSON responses with appropriate HTTP status codes.

Technical Requirements – Frontend

1. The frontend must be visually appealing and user-friendly.

2. The frontend must provide screens for:

- Add Movie

- Edit Movie

- Movie List

- Movie Details

3. Movie poster must be displayed using the static image URL served from the backend.

4. The search functionality must be implemented on the movie list page with search input based on movie title.

5. Proper form validations must be implemented on Add and Edit pages.

General Expectations

1. Code should be clean and properly structured.

2. API testing must be done before connecting the frontend.

3. The final application should run without errors on both backend and frontend sides.