

## Task: Build a CRUD Application for Movies

### Objective:

Develop a full-stack web application to manage a movie database. The backend will be built using Express.js, handling CRUD operations and storing data in a JSON file. The frontend will be a Vite-powered React application with at least two pages, providing a user interface for interacting with the movie database.

### Backend (Express.js):

#### 1. Setup and Structure:

- Initialize an Express.js application.
- Structure your application using middleware, routes, and controller functions.

#### 2. Data Storage:

- Use a JSON file to store movie data.
- Implement the `fs` module to read from and write to the JSON file.

#### 3. Endpoints:

- `GET` : Retrieve and display all movies.
- `POST` : Add a new movie to the database.
- `GET` : Search for movies by title or director (partial text match).
- Additional endpoints (choose at least two):
  - `PUT` : Update a specific movie's details.
  - `DELETE` : Remove a movie from the database.
  - `GET` : Filter movies by genre.
  - `GET` : Find movies released in a specific year.

#### 4. Middleware and Query Handling:

- Implement middleware for error handling and logging.
- Handle query parameters for filtering and searching movies.

### Frontend (Vite with React):

#### 1. Setup and Structure:

- Create a new Vite project .
- Structure your application with components and routing.

## 2. Routing and Pages:

- Implement React Router with at least two pages:
  - A home page listing all movies.
  - A search page for finding movies.

## 3. Consuming the API:

- Use Axios or Fetch API to communicate with your Express backend.
- Display the movie data fetched from the backend on your React frontend.

## 4. User Interface:

- Design a user-friendly interface for viewing, adding, and searching movies. (UI)
- Implement forms for adding and updating movies.

## Deliverables:

- Source code for both backend and frontend applications. (Github Repo)
- Documentation including setup instructions and API endpoint descriptions. ([ReadMe.md](#))

## Bonus Challenges (Optional):

- Implement user authentication. (can be local users.json file)
- Add pagination and sorting functionality to the movie list.
- Create a feature for rating and reviewing movies.