

Full-Stack Intern Assignment

MovieHub – Recommendation Board

Objective

Build and deploy a **Movie Recommendation Board** where users can suggest films, vote on them, and discuss picks. Keep the UI simple and functional.

Create a full-stack web app called **MovieHub** allowing people to:

- Recommend movies to watch
- Upvote/downvote recommendations
- Comment on any recommendation
- Admins can moderate content

Requirements

1) Authentication

- Basic signup/login. Use the library of your choice.

2) Database (MySQL or PostgreSQL or MongoDB)

Provide an **ERD diagram** and implement at least these tables/collections:

- **users** (id, name, email, password_hash, role, created_at)
- **movies** (id, title, description, added_by, created_at)
- **votes** (id, user_id, movie_id, vote_type, created_at)
vote_type can be up/down or +1/-1
- **comments** (id, user_id, movie_id, body, created_at)

3) Core Features

- **Suggest:** Authenticated users can add a movie (title + short description).
- **Vote:** Upvote/downvote a movie (one vote per user per movie; updating the vote is fine).
- **Rank:** Show list of movies sorted by score (upvotes – downvotes).
- **Discuss:** Users can comment under a movie.

4) Roles (Authorization)

- **User:** suggest movies, vote, comment; can edit/delete only their own comments (optional).
- **Admin:** view all, remove movies or comments, see **Top Movies** (overall leaderboard page/API).

5) Hosting

- Deploy the app (frontend + backend) on a free platform (Render, Railway, Vercel, Netlify, etc.).
- Share a **working live link**.

6) AI Usage

If you used AI at any step, add a short note in [README.md](#):

- Where was it used?
- Why was it helpful?
- What did you learn?

Deliverables

1. **GitHub Repository** containing:
 - Source code (frontend + backend)
 - [README.md](#) with:
 - Setup instructions
 - Database schema diagram (image or link)
 - Deployment link
 - Notes on AI usage
2. **Working Deployment Link** (public URL)

Evaluation Criteria

- Data model quality & relationships (users–movies–votes–comments)
- Correctness of voting, ranking, and comments
- Role handling (User vs Admin) and access control
- Hosting/deployment readiness and clarity of setup
- Code readability and basic structure
- Clear documentation of AI usage (if any)

Tips

- Prioritize correctness over visual polish.
- Keep routes/API simple and well-documented in the README.
- Seed a few sample movies for easy testing.