

CAPSTONE PROJECT 1 PROPOSAL

Project Proposal:

Movie and Series Explorer

Stack

- Frontend: React.js for building a dynamic and user-friendly interface.
- Backend: Node.js with Express.js for API handling and server-side logic.
- Database: MongoDB for storing user data, lists, reviews, and ratings.
- API: OMDb API for fetching movie and series information.
- Authentication: Firebase Authentication for secure user login and sign-up.

Description	
Stack Focus	The project will be a full-stack application with an even focus on both frontend and backend development.
Type	This will be a website optimized for both desktop and mobile devices (responsive design).
Users	<p>The target audience includes:</p> <ul style="list-style-type: none">• Movie and series enthusiasts who enjoy exploring new content.• Users who wish to organize their watchlists.• Individuals who like sharing their opinions through reviews.

Data	<ul style="list-style-type: none"> • Data Source: OMDb API for fetching movie and series details like titles, ratings, synopses, and genres. • Data Storage: MongoDB for: <ul style="list-style-type: none"> • User accounts and authentication. • User-generated content like lists, ratings, and reviews.
-------------	--

Breaking Down the Project

Task Name	Description
Set up GitHub Repository	Create and initialize the repository with a README file.
Design Database Schema	Outline collections and relationships in MongoDB.
Source API data	Integrate OMDb API and handle API responses.
Set up backend	Implement Node.js server and Express routes.
Set up frontend	Build React components for home, search, lists, and reviews.
User Authentication	Set up Firebase Authentication.

User Flows	<ol style="list-style-type: none"> 1. Exploration: <ul style="list-style-type: none"> • User searches for a movie/series and views detailed information. 2. List Management: <ul style="list-style-type: none"> • User creates, updates, or deletes custom lists. 3. Review Submission: <ul style="list-style-type: none"> • User adds ratings and text reviews for movies/series. 4. Authentication: <ul style="list-style-type: none"> • User signs up, logs in, or resets password securely.
------------	---

Labeling

Potential Issues

- API:
 - Rate limits from OMDb API.
 - Handling API downtime or errors gracefully.
- Sensitive Information:
 - Encrypting passwords and securing user data.
- Data Management:
 - Ensuring data consistency when deleting users or lists.

Label Type	Description
Difficulty	Medium-Hard
Type	Full-stack
Stretch Goals	Social features: Users follow each other to see shared lists. Analytics: Insights like “Most Added to Favorites” or “Top Rated.”