

ReelTIME Web Development Project Report

1. Project Overview

Developer Name: Baggiyam Shanmugam

Objective / Goal / Context

- **Objective:** The objective of the Reeltime web application is to provide users with a comprehensive and user-friendly platform for discovering, managing, and interacting with movies. It allows users to browse, search, and organize movies into personalized categories such as watchlists, favorites, and watched movies. Additionally, Reeltime enables users to connect with friends, share movie recommendations,
 - **Goal:** The primary goal of Reel Time is to create an immersive and social movie discovery experience. By incorporating movie management features, social interaction through friend requests and recommendations, and integration with TMDB (The Movie Database) for fetching movie details, Reeltime offers a seamless, interactive environment for movie enthusiasts to add ,track and suggest films to friends.
 - **Context:** In today's digital age, movie lovers often struggle to find a centralized place where they can both organize their movie collections and share recommendations with friends. Reeltime addresses this by merging movie management with social features, providing a personalized experience that goes beyond just tracking movies.
-

2. Scoping and Technical Feasibility Study

Functional Requirements Analysis

- **User Authentication:**
 - Users can register, log in, and manage their profiles
 - Secure login with password recovery and email verification features.
- **Movie Management:**
 - Users can search and browse movies through integration with the TMDB API, displaying movie details such as poster, trailer, ratings, description, and more.
 - Users can add movies to their personal watchlists, favorites, and watched lists.

- Reeltime allows users to add movies to their personal lists (watchlist, favorites, or watched) even if the movie is not already present in the database. The process works in two steps:
 - **Fetching from TMDB:** Users can search for a movie by entering its title. Reeltime will fetch movie details (e.g., description, poster, trailer, ratings) from the TMDB API.
 - **Manual Entry:** If the movie is not found through the TMDB fetch process, users will have the option to manually add the movie by entering the required details (e.g., title, description, poster URL, release date, etc.).
 - Admin users can edit or delete movies from the database.
 - **Social Features:**
 - Friend request system: Users can send, accept, or reject friend requests.
 - Movie suggestion system: Users can suggest movies to friends once connected.
 - **Admin Panel:**
 - Admins can manage user accounts, movie entries, and suggestions made by users.
 - Control over movie metadata, including auto-fetching details from the TMDB API.
-

User Experience Challenges:

- **Seamless Movie Discovery:** Ensuring that the movie search and browsing experience is quick and intuitive, especially when dealing with large datasets.
- **User Interaction:** Managing friend requests, movie suggestions, and comment updates in real-time.
- **Accessibility:** Ensuring keyboard and screen reader navigation, color contrast, and other accessibility guidelines are met.

Technical Feasibility:

- **Technology Stack:**
 - **Frontend:** React.js with Vite, CSS, and Styled Components.
 - **Backend:** Node.js, Express, MongoDB, and JWT for authentication.
 - **Deployment:** Netlify for frontend, render backend.
- **Resources:**
 - **Solo Developer** – Developed the project independently as a new learner with the MERN stack.
 - **Time Estimate:**
 - **Total Development Time:** 3 weeks for building the core features, integrating additional functionality, and refining the project.

Standards and Compliance:

- **Usability and Accessibility (RGAA):** The app will comply with accessibility guidelines, ensuring text contrast, alt text for images, and navigation ease for all users.
- **Security (ANSSI):** Secure data handling, HTTPS, JWT authentication, and bcrypt for password hashing.
- **SEO Standards:** Proper URL structure, title tags, schema markup, image optimization.

Technology Watch:

- Regularly reading blogs and articles (MDN Web Docs, Smashing Magazine, etc.) and attending conferences like React Conf for the latest updates and best practices.

3. Project Planning

Development Methodology:

- **Agile Methodology:** The project was divided into 3-week sprints, each focused on delivering specific features.
 - **Daily Stand-ups:** Regular check-ins to track progress.
- **Tools and Rituals:**
 - **Project Management:** GitHub.
 - **CI/CD:** GitHub Actions for continuous integration, and deployment.
- **Roadmap:**
 - **Phase 1:** Set up user authentication (login, signup, password recovery).
 - **Phase 2:** Implement core movie features.
 - **Phase 3:** Develop the friend request system
 - **Phase 4:** Develop the Movie suggestions
 - **Phase 5:** Focus on UI/UX improvement and responsiveness.
- **User Stories:**
 - **User Authentication:** "As a user, I want to log in and create an account to access my personalized movie lists."
 - **Movie List Management:** "As a user, I want to add movies to my lists (watchlist, favorites and watched)."
 - **Admin Features:** "As an admin, I want to manage movie details by edit and delete Feature in main movie database"

4. Technical Design

Database Modeling:

- **MongoDB with Mongoose** for data management.
 - **User:** `userID, name, email, password, friends, movies`.
 - **Movie:** `movieID, title, description, poster, releaseDate, genre, rating, userAdded`.
 - **MovieSuggestion:** `movieID, senderId, friendId`
 - **Friendrequest:** `status, sender, receiver`
-

5. Development

5.1 Server-Side Development (Back-End):

- **Node.js with Express** to handle API requests.
- **CRUD Functionalities:** Create, Read, Update, Delete endpoints for users, movies, and comments.
- **Security:** JWT authentication, password hashing using bcrypt, and role-based access control (admin vs regular users).

5.2 Client-Side Development (Front-End):

- **React.js** for dynamic content rendering.
- **Responsive Design:** Ensuring the app works seamlessly across mobile and desktop devices.
- **Accessibility:** Following RGAA guidelines for better accessibility.

5.3 Full-Stack Integration:

- **Axios** to make API requests and update UI with React's state management.
-

6. Conclusion and Future Work

The Reeltime project is a comprehensive and scalable movie management and social interaction platform. By integrating movie discovery features with a social aspect, Reeltime offers users a personalized and interactive movie experience. Future improvements will focus on enhancing user interaction and optimizing performance.

