Academic Project Report: MovieMind – Interactive Movie Recommendation System

1. Project Team

Name & Surname	Student ID	Roles & Responsibilities
Doğukan Öksüz	240229059	Project lead. Contributed across all JavaScript files, handled API integration and deployment, and played a major role in HTML/CSS development.
Tunahan Aydın	240229029	Developed explore-2.js, contributed to all HTML and CSS structures, and also authored the project report and README file.
Ali Duman	230229027	Developed watchList.js and contributed to all HTML and CSS structures.
Melihcan Tokdemir	230229049	Developed explore.js and contributed to HTML and CSS throughout the project.

2. Introduction

This project aims to develop an interactive web application that provides personalized movie recommendations based on user preferences. MovieMind, the proposed system, utilizes TheMovieDB (TMDB) API to offer dynamic movie suggestions, allows users to create a watchlist, and enhances user experience through browser-based data storage (localStorage).

2.1. Objectives

- To provide users with movie recommendations tailored to their selected genres

- To enable users to save preferred movies in a personal watchlist

- To deliver a responsive interface compatible with all devices

- To create an interactive experience using modern web technologies (HTML5, CSS3, JavaScript)

3. Methodology

3.1. Technical Implementation

3.1.1. HTML Structure

Semantic HTML5 tags were used to ensure an accessible and meaningful structure:

- Header: Navigation bar and logo section
- Main: Primary content area
- Section: Thematic blocks such as genre selection and movie recommendations
- Footer: Copyright and social media links

3.1.2. CSS and Responsive Design

- Flexbox and CSS Grid were employed for a flexible and consistent layout
- Media Queries ensured mobile compatibility:
  - Navigation menu transforms into a hamburger menu
  - Movie cards switch to a single-column layout on mobile devices

3.1.3. JavaScript and DOM Manipulation

- Event Listeners were implemented to manage user interactions:
  - Genre selection buttons (click)
  - Watchlist addition/removal functions
- Dynamic DOM Updates:
  - Movie cards were generated using document.createElement() and .appendChild()
  - Visual states were toggled with classList.add/remove()

3.1.4. API Integration

- TheMovieDB (TMDB) API was used to fetch movie data

- Asynchronous data retrieval was handled using fetch() and .then()

4. Findings and Application Features

4.1. Core Features

- Multi-genre selection for personalized recommendations

- User watchlist functionality

- Data persistence via localStorage

- Responsive and user-friendly interface

4.2. Advanced Features

- Real-time notifications using Toastify.js

- Genre mapping (genreMap) for filtering API data

- Error handling and user feedback mechanisms

4.3. Error Handling and Validation

- Navigation was restricted without genre selection

- Duplicate movie entries were prevented

- API errors were logged via console.log()

5. Testing and Evaluation

5.1. Testing Procedures

- Cross-Browser Testing: Verified functionality on Chrome, Firefox, and Edge

- Mobile Compatibility: Tested across different resolutions using Chrome DevTools

- API Response Checks: Users were notified of failed requests

5.2. Live Demo and Installation

- Live Project Link: https://movierecommendationwebsite.netlify.app/

- Installation Steps:

  1. Clone or download the project from GitHub

  2. Open index.html via a local server

  3. Insert the TMDB API key into the relevant JavaScript files

6. Conclusion and Recommendations

This project presents a user-centric movie recommendation system developed using modern web technologies.

References

- TheMovieDB API Documentation

- MDN Web Docs (JavaScript and DOM Manipulation)

End of Report