**MOVIE REVIEW RATING SYSTEM**

# **A PROJECT REPORT**

***Submitted by***

**HARINI .U(920422205035)**

***in partial fulfillment for the award of the degree of***

**BACHELOR OF TECHNOLOGY**

**IN**

**INFORMATION TECNOLOGY**



**DEPARTMENT OF INFORMATION TECHNOLOGY**

**KAMARAJ COLLEGE OF ENGINEERING AND TECHNOLOGY**

**(An Autonomous Institution - Affiliated to Anna University, Chennai)**

**K.VELLAKULAM, VIRUDHUNAGAR - 625 701**

**NOVEMBER 2024**

**KAMARAJ COLLEGE OF ENGINEERING AND TECHNOLOGY**

**(An Autonomous Institution- Affiliated to Anna University, Chennai)**

**K.VELLAKULAM, VIRUDHUNAGAR - 625 701**

**BONAFIDE CERTIFICATE**

Certified that the project report “**ARRIVING STANDARDS FOR MDR (MOVING DIE RHEOMETER) BY STATISTICAL METHODS”** is the bonafide work **Harini.U(920422205035)”** who carried out the project work under my supervision.

# **Dr. E. VAKAIMALAR** **Dr. R. ARTHY**

**Head of the Department,**  **SUPERVISOR,**

Associate Professor, Assistant Professor,

Dept. of Information Technology, Dept. of Information Technology,

Kamaraj College of Engg & Tech, Kamaraj College of Engg &Tech, K.Vellakulam, K.Vellakulam,

Virudhunagar - 625 701. Virudhunagar - 625701.

**ABSTRACT:**

The Movie Review Rating System provides users with a platform to rate, review, and explore movies in a dynamic and interactive environment. Developed using the MERN stack, this system leverages MongoDB for efficient data storage of user reviews, ratings, and movie details. Express.js, combined with Node.js, powers the server-side logic, ensuring smooth communication between the front-end and database. React.js offers a responsive and user-friendly interface where users can easily browse movies, read reviews, and submit their own feedback. Additionally, the system supports real-time updates, allowing users to instantly see how their reviews affect overall movie ratings. The architecture ensures scalability, flexibility, and ease of maintenance. This platform enhances user engagement with personalized movie suggestions, robust review filtering, and a rich interactive experience, making it a comprehensive tool for both movie enthusiasts and casual viewers alike.

**ACKNOWLEDGEMENT:**

I would like to express my sincere gratitude to Dr. E. Vakailamar, Head of the Department of Information Technology, for their continuous encouragement and valuable insights throughout the development of the "Movie Review Rating System" project. Special thanks to Dr. R. Arthy, my supervisor, whose guidance and expertise were instrumental in shaping the direction of this project. Their support helped me gain in-depth knowledge of the MERN stack and its application in creating sustainable, eco-friendly solutions for Movie reviews. This project has been an invaluable learning experience, enhancing both my technical and problem-solving skills.

**TABLE OF CONTENT**

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO.** | **TITLE** | **PAGE NO.** |
|  | [**ABSTRACT**](bookmark://_Abstract)  **LIST OF FIGURES** | iii  vi |
| **1** | **INTRODUCTION**  1.1 HTML  1.2 CSS  1.3 JavaScript  1.4 MERN Stack | **1**  1  2  3  4 |
| **2** | **METHODOLOGY**  2.1 Objective  2.2 Problem Statement  2.3 Block Diagram  2.4 Module Explanation | **7**  7  7  8  **9** |
| **3** | **RESULTS AND DISCUSSION** | **12** |
| **4** | **CONCLUSION** | **18** |
| **5** | **REFERENCES** | **19** |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **FIGURE NO.** | **TITLE** | **PAGE NO.** |
| 2.1 | Block Diagram | 8 |
| 3.1 | Home page | 13 |
| 3.2 | Movie review page | 13 |
| 3.3 | Registration page | 14 |
| 3.4 | About page | 14 |
| 3.5 | Login Page | 15 |
| 3.6 | Reviewit page | 15 |
| 3.7 | Review Page | 16 |
| 3.8 | Star rating page | 16 |
| 3.9 | Storing data in MongoDB | 17 |

**CHAPTER 1**

**INTRODUCTION**

**1.1 HTML:**

**HTML:**

HTML (HyperText Markup Language) is the standard markup language used to create the structure of web pages. It defines the content of a web page by using elements like headings, paragraphs, images, links, and forms. HTML provides the skeleton or foundation for all web content, allowing browsers to interpret and display the structure to users.

In the context of this project, HTML (via JSX in React) plays a crucial role in structuring the movie review system. By using various semantic HTML tags like <header>, <nav>, <section>, and <footer>, we were able to organize the content in a way that is both accessible and meaningful. Additionally, interactive elements like forms (<form>, <input>, <button>) were used to capture user reviews efficiently.

* **Semantic HTML**: Helps to make the webpage more accessible by using meaningful tags, such as <header>, <main>, and <footer>.
* **Forms and Input Fields**: Used to gather user input, where fields like movie titles, review text, and ratings are captured via <input>, <textarea>, and <form> elements.
* **Links and Navigation**: The project uses HTML anchors (<a>) and navigation elements (<nav>) for routing between different sections, ensuring smooth user transitions across pages like the review submission form and review list display.

**1.2 CSS**

**CSS:**

css (Cascading Style Sheets) is a design language used to enhance the visual presentation of HTML elements. In this movie review system project, CSS was crucial for creating a visually engaging and user-friendly interface. By applying a combination of global styles, reusable class-based styles, and component-specific customizations, the project achieves a cohesive look and feel aligned with the theme of movies and entertainment.

1. **Theme Customization**: The overall theme of the website reflects a cinematic experience. A dark color palette, is applied to backgrounds and headers to mimic the movie theater atmosphere, while vibrant accent colors are used for buttons and ratings to stand out.
2. **Layout and Responsiveness**: CSS Flexbox and Grid layout techniques were extensively used to build responsive and flexible designs, ensuring the movie review system works well on all screen sizes. The reviews and ratings are neatly organized for easy navigation across devices.
3. **Typography**: Typography plays a crucial role in establishing a clean, professional aesthetic:
   * **Font Choices**: Sleek and modern sans-serif fonts (such as Helvetica or Roboto) were chosen to keep the design minimalistic, ensuring that the reviews are easy to read.
4. **Buttons and Interactions**: CSS provides styling for interactive elements like buttons. The buttons are designed to be visually appealing and easy to interact with, featuring hover effects, rounded corners, and animations to enhance user engagement during actions like submitting reviews or ratings.

**1.3 JAVASCRIPT:**

JavaScript is a high-level, versatile programming language that enables interactive and dynamic features on web pages. It plays a crucial role in making web applications responsive, providing a seamless user experience by dynamically updating content and handling user inputs without requiring a full page reload.

In this movie review system project, JavaScript is primarily used through React.js, a popular JavaScript library for building user interfaces. React.js efficiently manages the state of the application, handling real-time updates to user reviews, ratings, and form submissions. Some of the key JavaScript features in this project include:

* **Review Submission Functionality**: JavaScript enables users to submit movie reviews and ratings dynamically. The reviews are immediately displayed on the page without needing a page reload.
* **Page Navigation**: React Router is used to handle navigation between pages like "Home," "Movie Reviews," and "Submit Review." This provides a smooth, single-page application (SPA) experience, where users can move between different sections of the website without page refreshes.
* **Form Handling**: JavaScript manages the entire flow of form handling during review submission, including input validation for fields such as movie title, review text, and rating. It ensures that the review is correctly submitted to the server and provides feedback to the user about the success or failure of their submission.
* **Event Handling**: The "Submit Review" and "Rate Movie" buttons are powered by JavaScript, allowing real-time responses to user clicks. When a user interacts with these buttons, JavaScript functions are triggered to execute the appropriate actions, such as submitting the review or updating the ratings on the page.

**1.4 MERN STACK:**

The MERN stack is a popular web development framework that combines four technologies: MongoDB, Express.js, React.js, and Node.js. This stack allows developers to build full-stack applications using JavaScript across the entire development process, from client-side to server-side and database management.

#### **Components of the MERN Stack**

1. **MongoDB**:
   * A NoSQL database that uses a document-oriented data model, allowing data to be stored in flexible, JSON-like formats (BSON).
   * It supports horizontal scaling, making it suitable for handling large volumes of data.
   * MongoDB’s rich query language and indexing support facilitate efficient data retrieval.
2. **Express.js**:
   * A minimal and flexible web application framework for Node.js, designed for building robust APIs and web applications.
   * It simplifies server-side development by providing features like middleware, routing, and request handling.
   * Express enables developers to set up middleware to respond to HTTP requests, serving static files, and handling errors effectively.
3. **React.js**:
   * A JavaScript library for building user interfaces, particularly single-page applications (SPAs), developed by Facebook.
   * React promotes the creation of reusable UI components, enhancing maintainability and performance.
   * It utilizes a virtual DOM to optimize rendering, ensuring that only the necessary parts of the UI are updated during state changes.
4. **Node.js**:
   * A JavaScript runtime built on Chrome’s V8 engine, allowing developers to execute JavaScript server-side.
   * Node.js is event-driven and non-blocking, making it suitable for handling multiple concurrent connections efficiently.
   * It provides a rich ecosystem of libraries and tools via npm (Node Package Manager), enabling rapid development.

### **How the MERN Stack Works**

The MERN stack operates as a cohesive unit where each component works together to create a full-stack application. Here's how the workflow typically unfolds:

1. **Client-Side (React)**:
   * Users interact with the application through the React front end, where components render the UI and manage user interactions.
   * React uses a unidirectional data flow, meaning that data is passed down from parent to child components, ensuring predictable behavior.
   * When users perform actions (like submitting a form), React captures these events and triggers corresponding functions to handle the logic.
2. **Communication (Axios/Fetch API)**:
   * React communicates with the server using HTTP requests, typically through libraries like Axios or the Fetch API.
   * When a user action necessitates data retrieval or submission (e.g., fetching user profiles), React makes an API call to the Express server.
3. **Server-Side (Express and Node.js)**:
   * The Express server receives HTTP requests from the React front end.
   * It processes these requests, executes necessary business logic, and interacts with the MongoDB database.
   * Express routes define the endpoints and handle different HTTP methods (GET, POST, PUT, DELETE) for specific functionalities, such as user authentication or data manipulation.
4. **Database Interaction (MongoDB)**:
   * When the Express server needs to read or modify data, it queries MongoDB using Mongoose (an ODM for MongoDB).
   * MongoDB stores data in collections, and Mongoose allows for schema definition and validation, providing an interface to interact with the database.
   * The results of these queries are returned to the Express server, which then sends a response back to the React front end.

**Chapter2**

**METHODOLOGY**

**2.1 OBJECTIVE:**

The development of the Movie Review Rating System follows a modular and iterative approach using the MERN stack, ensuring efficient handling of data, user interactions, and real-time updates.

**2.2 PROBLEM STATEMENT:**

In the digital age, movie enthusiasts lack a comprehensive and interactive platform that allows them to share their opinions on films, discover new movies, and engage with fellow users in real time. While there are numerous movie review websites, many face issues such as slow updates, limited interactivity, and inadequate real-time feedback on user reviews. Additionally, many platforms do not offer personalized movie recommendations based on user preferences or provide an efficient way to aggregate and display real-time ratings

**2.3 BLOCK DIAGRAM:**

**REGISTRATION**

**LOGIN**

**LOGIN SUCCESSFUL**

**REVIEW**

**STAR RATING**

**REVIEW**

**SUCCESSFULL**

**2.4 MODULE EXPLANATION:**

**1. Frontend: React.js (Client-Side)**

* **Purpose**: React.js is used to create a dynamic, responsive user interface (UI) for users to interact with the movie review system.
* **Key Components**:
  + **MovieList Component**: Displays the list of movies and associated reviews. This component fetches data from the backend and renders the movies with their reviews and ratings.
  + **ReviewForm Component**: Allows users to submit new reviews for a specific movie. It includes form fields like movie title, review text, and rating. This component uses state to handle form inputs and submissions.
  + **RatingComponent**: Displays a star-based or numeric rating system where users can rate a movie. It can be interactive, allowing users to select their rating.
  + **Navigation and Routing (React Router)**: Manages page navigation, enabling users to move between different sections, like the home page, movie review listing, and the review submission page, without page reloads.

**2. Backend: Express.js & Node.js (Server-Side)**

* **Purpose**: The backend is responsible for managing API endpoints, handling requests, processing data, and interacting with the database.
* **Key Modules**:
  + **Movie Routes (movieRoutes.js)**: Contains the various API routes related to movie data and reviews. These routes define the endpoints for:
    - Fetching all movies and their reviews (GET /movies)
    - Fetching a single movie and its reviews (GET /movies/:id)
    - Submitting a new review for a movie (POST /movies/:id/reviews)
  + **Review Routes (reviewRoutes.js)**: Contains the routes that handle review-specific operations, like adding, editing, or deleting a review for a movie.
  + **Middleware**: Custom middleware might be used for authentication (to protect routes), request validation, and error handling. For example, authentication middleware ensures that only logged-in users can submit reviews.
  + **Controllers (movieController.js & reviewController.js)**: These are the logic handlers that process incoming requests and interact with the database. For instance, the addReview function would handle the logic for saving a new review to the database and returning an updated list of reviews.

**3. Database: MongoDB (Database Layer)**

* **Purpose**: MongoDB is used to store the movie data, user reviews, and ratings in a flexible, schema-less NoSQL format.
* **Key Models**:
  + **Movie Model (Movie.js)**: Represents the structure of a movie document in MongoDB. It may include fields like title, description, releaseDate, and an array of reviews.
  + **Review Model (Review.js)**: Represents the structure of a review document. It includes fields like movieId (which associates the review with a specific movie), userId (the user who submitted the review), rating, and comment.
  + **User Model (User.js)**: Manages user accounts for the system. It might include fields such as username, email, password, and role (to distinguish between regular users and admins).

**4. Backend Logic**

* **Authentication and Authorization**:
  + **User Authentication**: The system can use JWT (JSON Web Token) or another authentication mechanism to manage user login and access control. Users must log in to submit reviews or view their review history.
  + **Protected Routes**: Certain API routes (like submitting a review) are protected to ensure only authenticated users can access them.
* **Validation**: Backend validation ensures that submitted reviews meet the necessary criteria (e.g., valid movie ID, valid rating, non-empty review comment).
* **API Integration**: The backend exposes RESTful API endpoints that the frontend (React) interacts with. The frontend sends HTTP requests (GET, POST, etc.) to these APIs, which return JSON data for rendering.

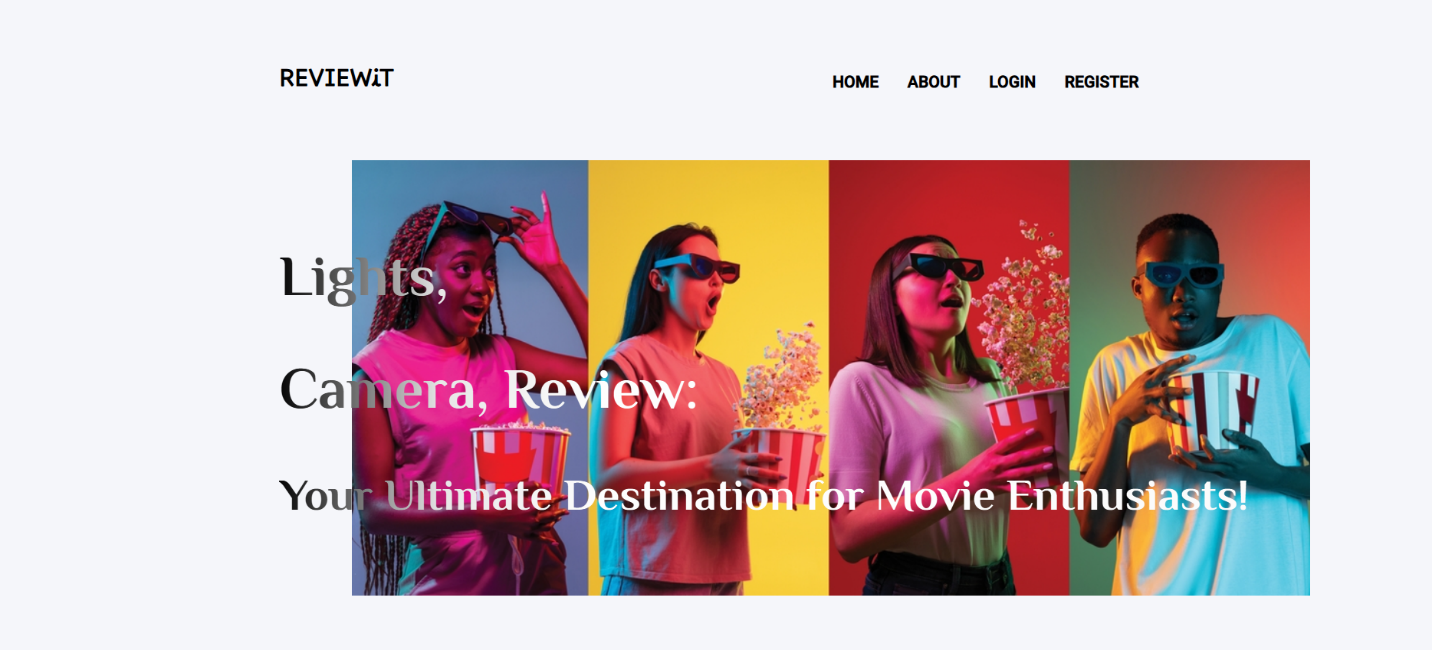
**CHAPTER 4**

**RESULTS AND DISCUSSION**

The movie review system built using the MERN stack offers a clean and user-friendly design. Users can easily browse movies, view reviews, and submit their own ratings and reviews. The movie details, including titles, descriptions, and user ratings, are dynamically displayed, and the review submission process works smoothly, allowing users to interact with the system without page reloads. On the backend, the system effectively manages user-generated reviews and stores them in the database, ensuring scalability to handle more movies and reviews as the platform grows.

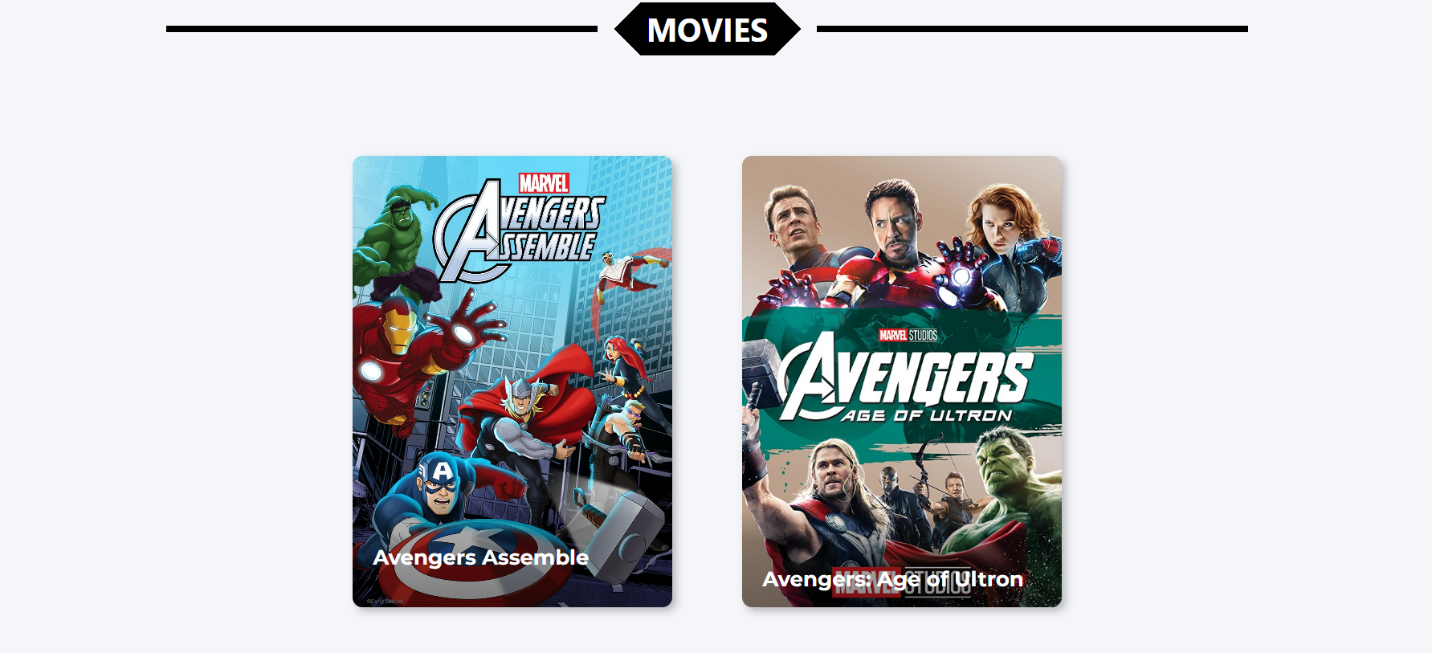
While the platform is functional, it could benefit from additional features such as filtering movies by genre, sorting reviews by rating or date, and enabling users to like or comment on reviews. Performance improvements, such as optimizing the loading time for images or large movie data sets, would further enhance the user experience. Ensuring secure authentication and authorization for user-generated content and preventing spam or malicious activity are important considerations. Future updates could include personalized movie recommendations using AI, integrating social sharing features, and fostering community engagement through discussion forums or review challenges to make the platform more interactive and engaging for movie enthusiasts.

**SCREENSHOTS:**

****

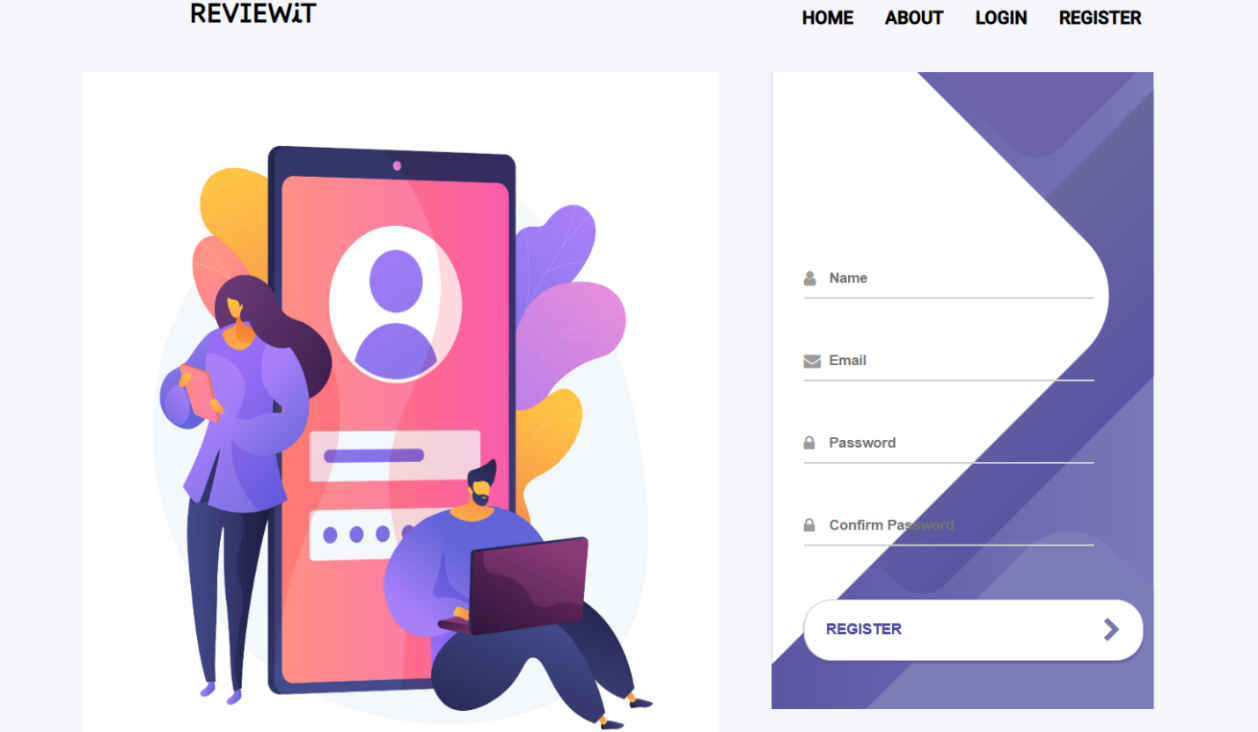
**Figure 3.1 Home page**

The figure 3.1 It shows the home page of movie rating review system

****

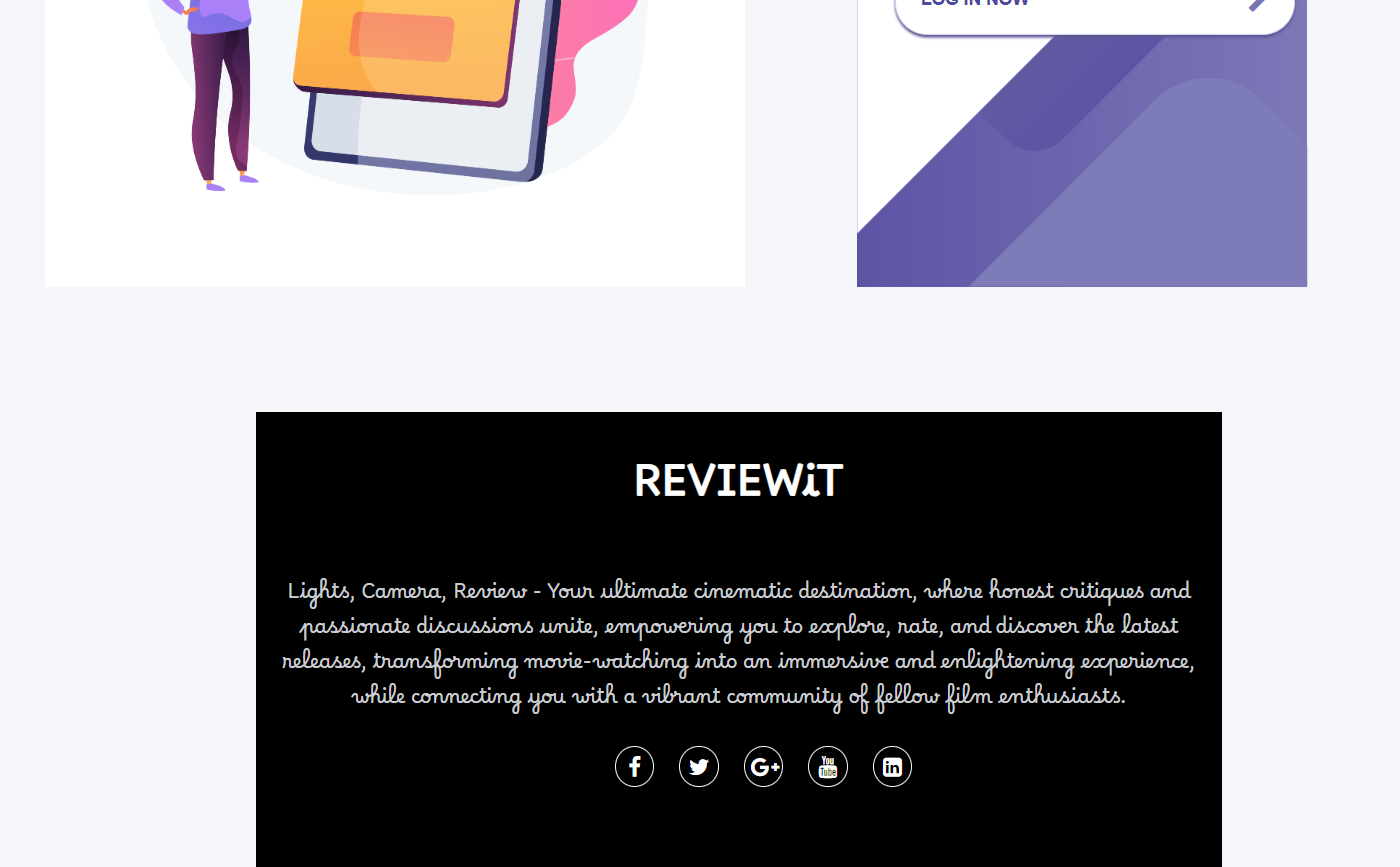
**Figure 3.2movies page**

figure 3.2It shows the page of list of movies in the home

****

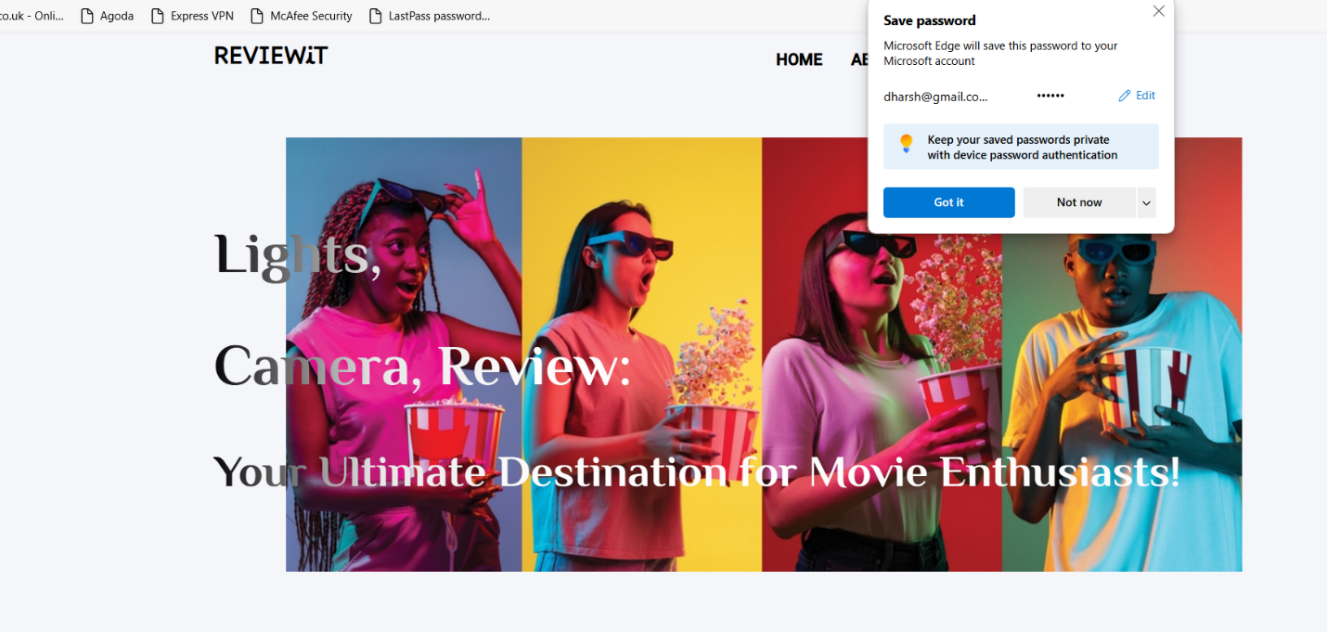
**Figure 3.3 REGISTRATION page**

figure 3.3this pageshows the regirstration

****

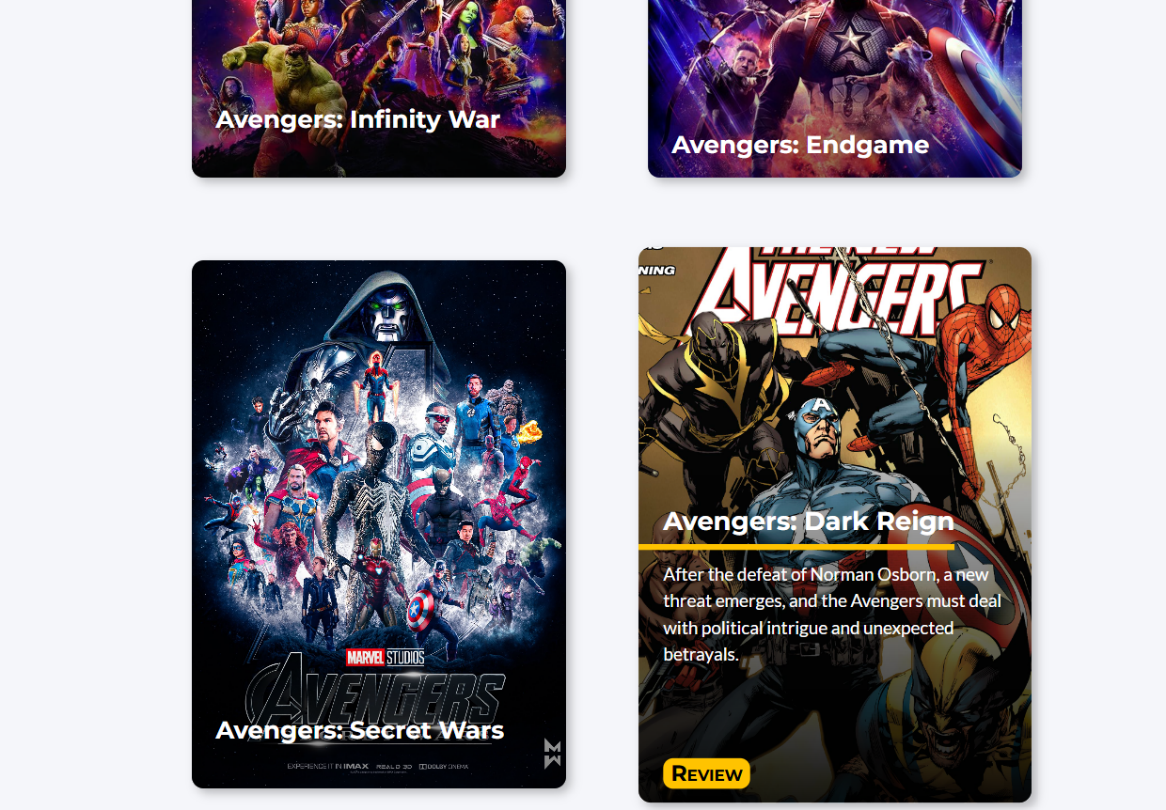
**Figure 3.4 About page**

figure 3.4 this page shows the about my page

****

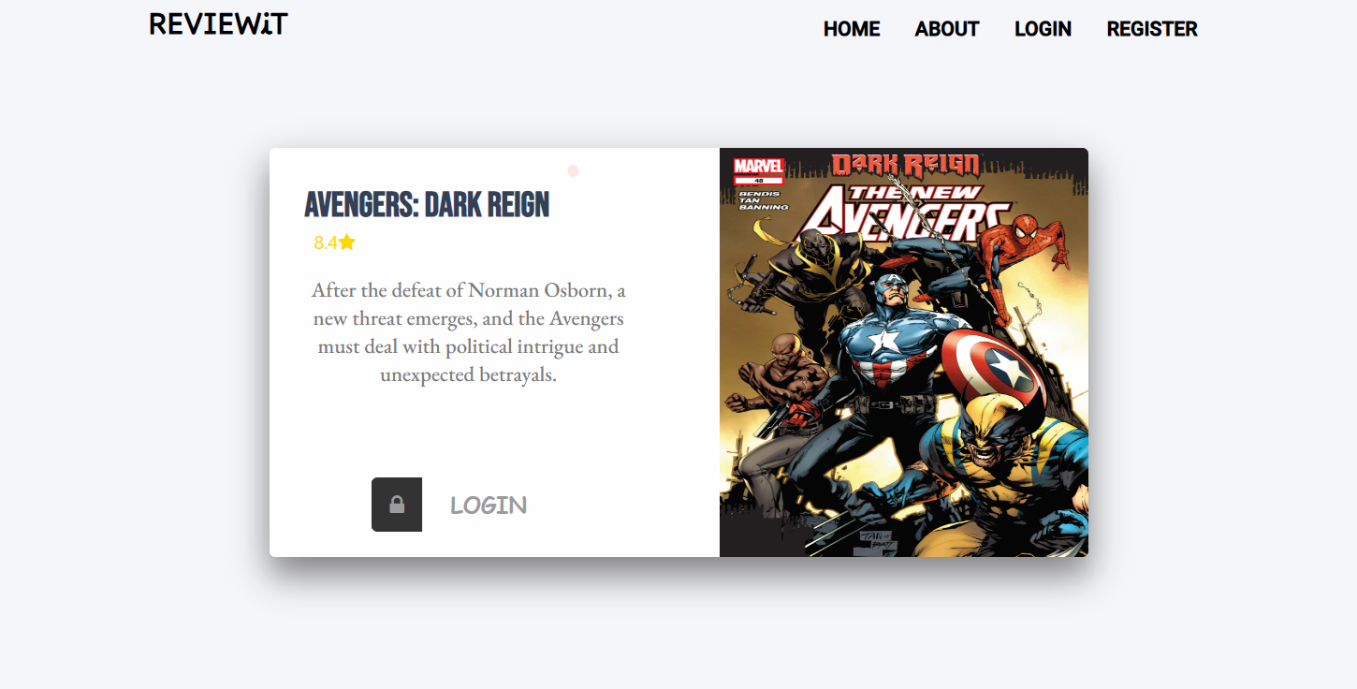
**Figure 3.5 login shows**

figure 3.5 this page shows the login successful

****

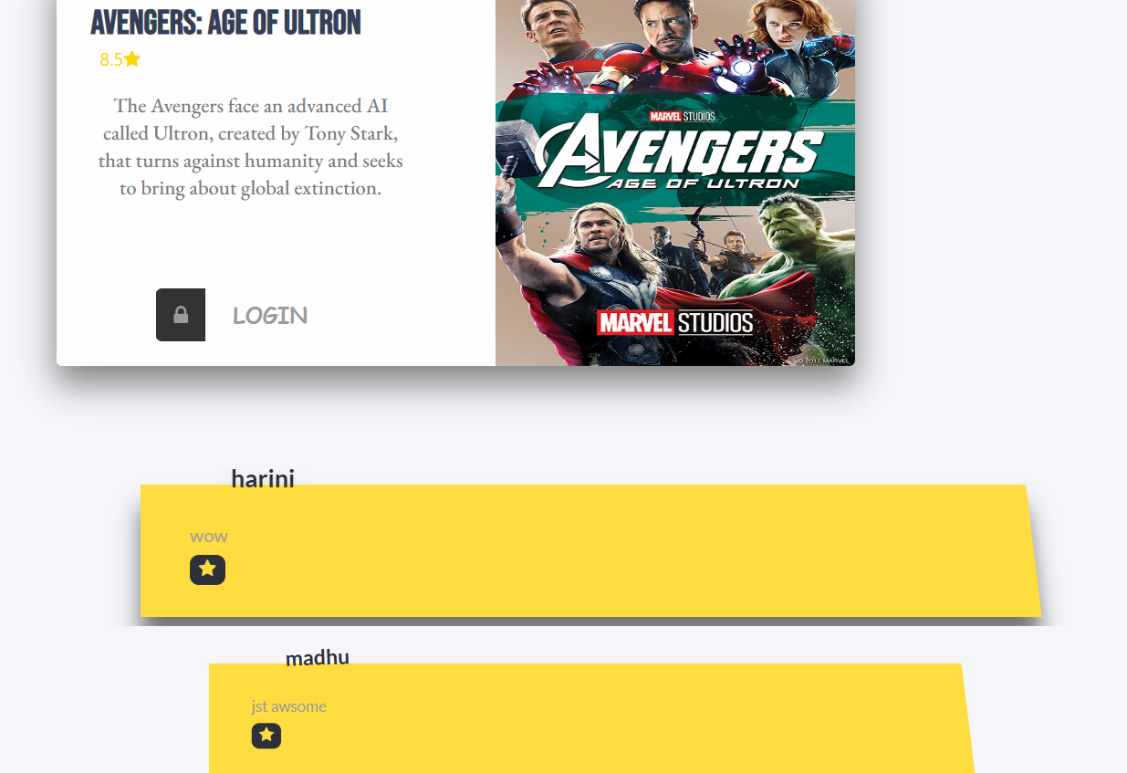
**Figure 3.6 To Review the movie**

Figure 3.6 It Shows to click and reviewit on this page

****

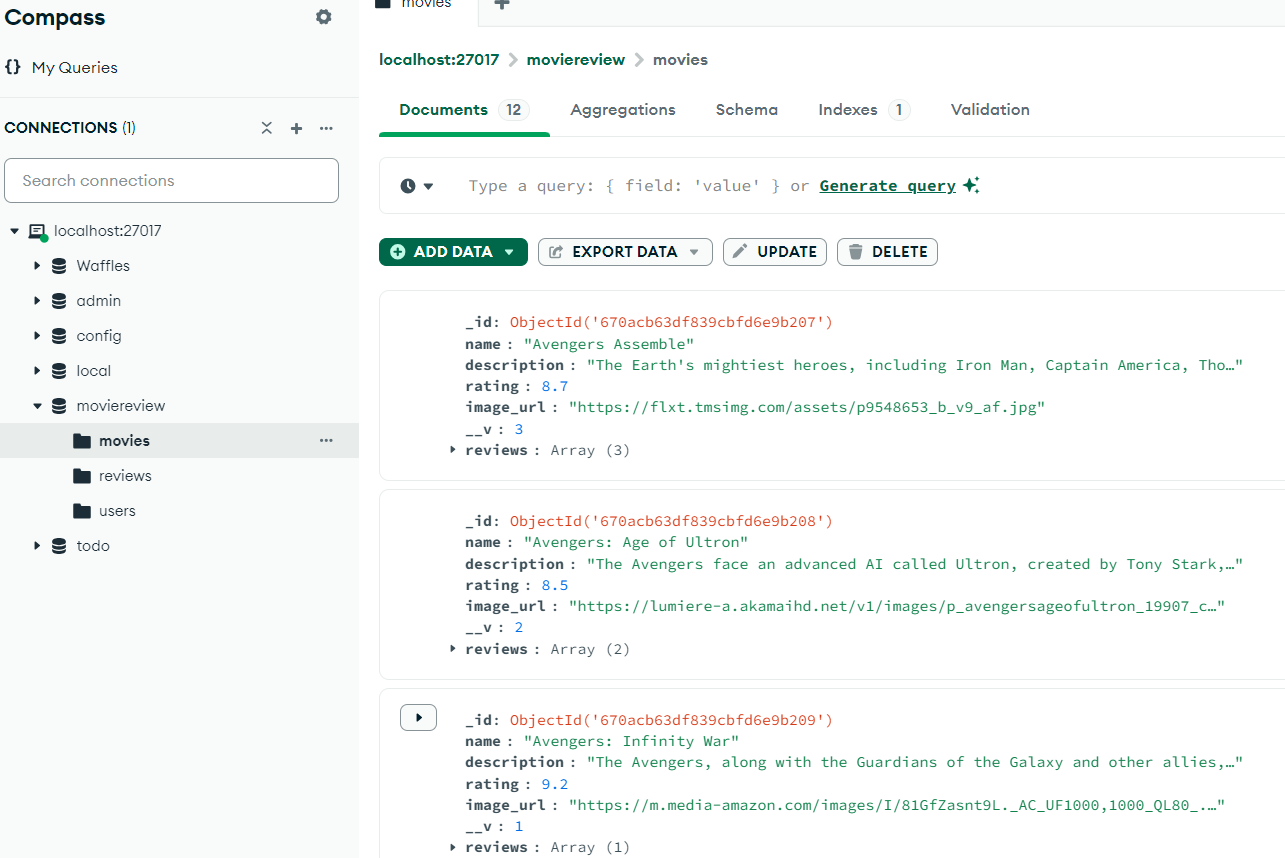
**Figure 3.7 review page**

figure 3.7 this page shows the description and review

****

**Figure 3.8 Place your movie review and ratings**

figure 3.8 It shows to give the star rating

****

**Figure 3.9:Storing data in MongoDB**

Figure 3.9 It shows the user ,movies ,description ,star rating

**CHAPTER:4**

**CONCLUSION**

The MERN-based Movie Review Rating System effectively fulfills its core objectives of allowing users to submit, rate, and review movies in real-time while providing a smooth and responsive user experience. With efficient data storage using MongoDB, a dynamic and intuitive React frontend, and fast backend performance through Node.js and Express, the system is well-optimized for both functionality and scalability. However, there are opportunities for further enhancements, including better verification for reviews, advanced security features, and additional social and personalization tools to increase user engagement. Overall, it provides a solid foundation for future improvements and expansion**.**

**REFERENCES:**

* 1. CinAmi GitHub
  2. Movie-MERN-APP GitHub
  3. https://youtu.be/Bd1EBSCu2os?si=6my55IWyLQWfSbtH