Movies Review & Information System

Muhammad Nouman Zafar

A00314863

B.Eng. Software Engineering

Technological University of the Shannon: Midlands Midwest

Table of Contents

Introduction

My comprehensive movie review and information system is here to help film enthusiasts dive deep into the world of cinema. The purpose of this platform is to provide users with in-depth knowledge about their favourite films, as well as an plenty of related content and functions.

There is an extensive database of movies waiting to be explored when user enter my website. A wealth of information can be accessed with just a few clicks, including the title, genre, release date, description, overall rating, and user ratings. User can find the information they're looking for quickly using the intuitive user interface.

A dedicated page that serves as a treasure trove of movie-related content is available when user click on a specific movie. A comprehensive list of individuals involved in the making of the film can be found here. Each person is linked to a page where user can explore all the movies associated with them.

This platform goes beyond mere information display. User can get a glimpse of the movie by watching related clips, such as trailers. User can also post their thoughts and opinions in the comments section, creating a lively community of movie fans.

A Toolkit page offers a range of advanced features for those who want to take their experience to the next level. User can alter the data and contribute to the ever-growing database by adding new movies and people or linking people to specific movies.

This movie review and information system is ultimate destination for all things cinema, no matter what kind of movie user like.

The technical project documentation serves as a comprehensive guide to understanding the architecture, functionality, and implementation details of my movie review and information system application. From the initial conceptualization to the deployment phase, each aspect of development process is documented herein, offering insights into the design choices, challenges encountered, and solutions adapted during the phases like brain storming phase, planning, implementation.

Key components of the Movie Review and Information Application are divided into three different sections and each section serves a very important role to complete the application, and those three components are

- Front End
- Back End
- Storage

Research

The preliminary research for the Movie Review and Information Application focused on identifying existing movie review platforms, assessing user requirements, and assessing potential technologies and frameworks. Like TMBD, IMDB etc.

My initial plan was to integrate with existing movie databases to retrieve comprehensive movie information. The platforms provided me with APIs to access movie titles, genres, release dates, descriptions, and ratings. However, I encountered difficulties with the usage guidelines and restrictions associated with these APIs, prompting me to investigate other options. At the same time, I wanted to build my own API using backend services, if I had used the external APIs then there would be no need of backend or Database, so My final decision was to build everything myself because this way every component of the project is controlled by me.

After evaluating several options, including external APIs and open data sources, I decided to build my own database of movie information. Flexibility in data management and the desire to avoid potential restrictions imposed by third-party APIs were factors that influenced this decision. Even though I am built my own database but I ran that database on AWS EC2 console instance with the help of Ubuntu virtual machine. For data storage, I explored many options like relational and NoSQL databases. In the end I decided to use MySQL database which is easier to integrate with the back and front-end technologies.

I considered several options for both the front-end and back-end components of the application. The front end was evaluated for their ability to provide a dynamic and responsive user interface. It was ultimately chosen for its popularity, extensive community support, and robust ecosystem of libraries and tools. As front-end technology first, I wanted to use Angular but after the extensive research I realised that it is way harder to handle and there are so many things needs to be sorted before using Angular so eventually I decided to use React with the help of Node.js for my front-end user interface.

I evaluated different programming languages and frameworks for building APIs and managing data storage on the back end. While alternatives like Java, Python with Django and Node.js with Express were considered, I ultimately settled on Spring-boot due to its

robust features for both API development and data storage management with the help of Speedment. Overall, its combination of flexibility, performance, and community support made it the optimal choice for my project's backend development needs.

Technologies

Movie Review and Information application is divided into three different sections

- Front End
- Back End
- Storage

Front-End Technologies:

The technologies used to build the front-end user interface are:

- React
- React Router DOM
- Axios
- React Player
- HTML & CSS
- JavaScript
- Node.js
- Local Storage
- VS Code
- npm

React:

- React is a JavaScript library for building user interface. It allows developers to create reusable UI components and efficiently manage the state of the application.
- React's component-based architecture encourages a modular approach to building applications, making it easier to maintain and scale projects.
- JSX is used to write React components, allowing developers to write HTML-like syntax within JavaScript code.

React Router DOM:

- React Router DOM is library that provide routing capabilities for React applications.
- It allows developers to define different routes within the application based on URL, enabling navigation between different components or pages of the application.
- Features like dynamic route matching and nested routing make it powerful for building complex single-page applications.

Axios:

- Axios is a popular JavaScript library for making HTTP requests from he browser or from the backend of the whole application.
- It provides a simple and intuitive API for performing asynchronous operations like fetching data from server or posting data to an API endpoint.
- Axios supports features like request and response interception, automatic transformation of JSON data, and handling errors.

React Player:

- React Player is a React component specially designed for playing media files such as videos and audio.
- It abstracts away the complexities of implementing media playback in a web application, providing a simple interface for embedding media content.
- React Player supports various media sourced, including URLs, files, and streaming services like YouTube.

HTML & CSS:

- HTML (Hypertext Mark-up Language) is used for creating web pages and applications. It provides the structure and content on webpage.
- CSS (Cascading Style Sheets) is used to style HTML elements, controlling their appearance, layout, and presentation on the screen.

JavaScript:

- JavaScript is a programming language of the web, used for adding interactivity and dynamic behaviour to the web pages and applications.
- JavaScript features like arrow functions, restructuring assignment, template literals, and async/await are commonly used in React development for writing clean and concise code.

Node.js:

- Node.js is a JavaScript runtime environment that allows developers to run JavaScript code outside of a web browser.
- In this application Node.js is used in order to use its component npm

Local Storage:

- Local Storage is a web browser feature that allows web applications data to store locally on the user's device.
- It provides simple key-value storage mechanism and can store large amounts of data composed to cookies.
- In my Application Local Storage is used to store the information of a logged in user like email, likely for authentication purpose.

Visual Studio Code:

- VS Code is a free and open source code editor developed by Microsoft.
- It provides a rich set of features for coding, debugging, and version control which makes I a popular choice among the developers.
- VS Code offers built-in support for JavaScript, JSX, HTML, CSS and various other programming languages and file formats commonly used in web application development.

npm (Node package manager):

- npm is the default package manager for Node.js, the JavaScript runtime environment.
- It allows developers to install, share and manage dependencies for their projects.
- With npm, developers can easily integrate third-party libraries, frameworks, and tools into their applications.
- npm also provides a command-line interface for package installation, version management, and script execution.

Back End Technologies:

The technologies used to build the back-end are:

- Spring Boot
- Spring Web
- Spring Framework Annotations
- Spring Response Entity
- CORS
- Java Streams
- Java Lambda Expression
- Java Generics
- Java Collections Framework
- JPA
- Dependency Injection
- Speedment
- Maven
- Eclipse

Spring Boot: Spring boot is a popular Java framework for building web applications. It has tools for building RESTful APIs, handling HTTP requests, and handling dependencies.

Spring Web: Spring Web is a component of the Spring framework and offers tools for building web applications, including RESTful services.

Spring Framework Annotations: Annotations like @RestController, @RequestMapping, @GetMapping, @PostMapping, and @RequestBody are used to define REST endpoints and handle HTTP requests.

Spring Response Entity: This class is used to represent the entire HTTP response. It allows you to control the HTTP response status, headers, and body.

CORS: Spring Cross-Origin Resource Sharing @CrossOrigin annotation allows cross-origin requests from any origin. This is useful when the client-side code is served from a different domain.

Java Streams: Java Streams are used for processing collections of data in a functional style. They allow you to perform aggregate operations on collections, such as filtering, mapping, and reducing.

Java Lambda Expression: Lambda expressions are used in stream operations to concisely define inline functions.

Java Generics: Generics are used throughout the code to provide type safety and reusability.

Java Collections Framework: The List and Map interfaces from the Java Collections Framework are used to store and manipulate collections of data.

Java Persistence API: It is used for data persistence as the back-end application interacts with database e.g. (MovieManager, MoviePeopleManager etc.).

Dependency Injection: Spring boot's dependency injection is used in the constructor of the Controller class to inject instance of Manager classes.

Speedment: It is a java object relational mapping tool that allows developers to create java applications with high-performance database access layer. Speedment generates java code based on an existing database schema, providing type-safe APIs for interacting with the database.

Maven: Maven is a build automation tool primarily used for Java projects. It is primarily used for building Java projects. Managing a project's dependencies, compiling source code, and packaging the application into distributable forms like JAR files are all streamlined by it. Maven takes a declarative approach, where developers specify project configurations, dependencies, and build steps in an XML file called pom.xml (Project Object Model) The directory structure in Maven makes it easier for developers to work together and share code. Overall, Maven helps streamline the development process by automating repetitive tasks and enforcing best practices in Java project management. It also helps enforce best practices in Java project management.

Eclipse: Eclipse is an integrated development environment (IDE) primarily used for Java development, although it supports various programming languages through plugins. It provides developers with a comprehensive set of tools for writing, debugging, and deploying software applications.

Storage Technologies:

Technologies used to store the data are:

- MySQL Database
- AWS EC2
- Security Groups
- PuTTY

MySQL Database: MySQL is an open-source relational database management system. It's the core technology for storing and managing structured data.

AWS EC2: Elastic Compute Cloud provides scalable computing capacity in the cloud. You launch virtual servers, known as instances, which can run various operating systems and applications, including MySQL.

Security Groups: Security groups act as virtual firewalls for your EC2 instances to control inbound and outbound traffic. You can configure security groups to allow only necessary traffic to reach your MySQL server.

PuTTY: PuTTY is a popular open-source SSH client. It allows you to securely connect to your EC2 instance over SSH (Secure Shell) protocol. With PuTTY, you can remotely access the command-line interface of your EC2 instance, where you can execute commands to manage your MySQL server, such as querying databases, configuring settings, or running maintenance tasks.

Application Domain

The movie review and information system application domain encompass various aspects related to the world of cinema, catering to the needs and interests of movie enthusiasts, critics, and casual viewers alike.

The application domain is broken down into different sections like:

- Movie Information
- User Interaction
- Movie Search
- Cinematic content
- Industry Insights
- Recommendations
- Data Management
- Accessibility
- Community Building

Movie Information: The core of the domain revolves around providing comprehensive information about movies. This includes information such as title, genre, release date, synopsis, cast and crew, ratings, trailers, and related media content.

User Interaction: User interaction is facilitated by features like comments, evaluations, and evaluations in this application. Users can share their thoughts, gain insights, and discuss their favourite films. This creates a lively community of movie enthusiasts within the platform.

Movie Search: The application domain incorporates a robust movie search feature, allowing users to find specific movies easily. Users can search by title or keyword. This feature enhances usability and efficiency by enabling users to discover relevant movies quickly.

Cinematic content: Feature films, documentaries, short films, and series are covered by the domain. It caters to diverse preferences, covering various genres, languages, and cultural backgrounds.

Industry Insights: Information about filmmakers, actors, producers and other professionals involved in the movie-making process is provided by this application.

Recommendations: This application domain provides the recommendation based on users selected movie genre. It will display all the movies with the same genre.

Data Management: This application manages vast amounts of data related to movies, users, comments, ratings, and interactions. Database management systems are used to organize, retrieve and update information efficiently.

Accessibility: The domain emphasizes accessibility, ensuring that users can easily navigate the platform, search for specific movies, and access relevant content. User-friendly interfaces, intuitive navigation menus, and robust search functionalities enhance usability.

Community Building: A sense of community is built by providing features for social interaction, collaboration, and knowledge sharing through the usage of feature of posting comments for any selected movie.

By addressing these key aspects of movie watching experience, this application creates a comprehensive and engaging environment for users to explore, interact and appreciate the art of cinema.

Requirements

The purpose of the Movie Review and Information System is to provide users with a comprehensive platform to explore and interact with a vast database of movies, covering various genres, release dates, and ratings. The application aims to help users discover, analyse, and discuss their favourite films, while also providing insights into the movie-making process and fostering a sense of community among movie enthusiasts.

There are two type of requirements:

- Functional Requirements
    - Movie Exploration Functionality
    - Display Movie Information
    - User Interaction Features
    - Advance Functionality

- Non-Functional Requirements
    - Performance
    - Usability
    - Security

Functional Requirements

Movie Exploration Functionality:

- **Search by Title:** User should be able to search by movie title.
- **Browse by Category:** user should be able to browse movies by category such as popular, top-rated, recently released.

Display Movie Information:

- **List View:** Display search results or browse categories in a list format, showing relevant details such as title, genre, release date, and thumbnail image.
- **Detailed View:** Allow users to access detailed information about a selected movie, including synopsis, cast and crew, ratings, trailers, and related media content.

User Interaction Feature:

- **Comments Section:** Enable users to post comments and discuss their thoughts and opinions on specific movies.
- **Rating:** Allow users to rate movies and write reviews to share their feedback with the community.

Advanced Functionality:

- **Toolkit Page:** Offer advanced features for users who want to contribute to the database, such as adding new movies, linking individuals to specific movies, and updating movie information.
- **Registration:** Allow new users to register themselves, when they register their information stores in database and whenever they try to log in their provided details will match the details stored in database.

Non-Functional Requirements

Performance:

- **Response Time:** Ensure that movie information is retrieved and displayed promptly, with minimal latency.
- **Scalability:** Design the system to handle a large number of concurrent users and a growing database of movies without compromising performance.

Usability:

- **Intuitive Interface:** Design a user-friendly interface with intuitive navigation and clear organization of movie information.
- **Accessibility:** Ensure that the application is accessible to all users.

Security:

- **Data Privacy:** Implement measures to protect user data, including encryption of sensitive information and secure authentication mechanisms.
- **Secure Communication:** Use HTTPS protocol to encrypt data transmitted between the client and server, ensuring privacy and integrity.

Technology Stack

Frontend: Utilize React.js for building the frontend user interface, providing a responsive and interactive user experience.

Backend: Implement Spring Boot for the backend application, handling API requests, data retrieval, and database management.

Database: Utilize MySQL for storing and managing movie information, user data, and interaction records.

Constraints

Data Storage: Ensure that sufficient storage capacity is available to store the growing database of movies and user interactions.

AWS EC2: Utilize AWS EC2 for hosting the application and database, ensuring scalability and reliability of the infrastructure.

Assumptions

Internet Connectivity: It is assumed that users will have a stable internet connection to access the application and retrieve movie data.

Modern Devices: The application is designed to be compatible with modern web browsers and devices, including desktops, laptops, tablets, and smartphones.

Dependencies

Third-Party Libraries: Identify and manage dependencies on third-party libraries and frameworks used in the development of the application, including those managed via npm for frontend and Maven for backend.

Custom APIs: Acknowledge the dependency on custom APIs developed for fetching movie data, managing user interactions, and integrating additional features.

AWS Services: Recognize the dependency on AWS services such as EC2 for hosting the application and database management.

Architecture

There are three sections of the application

- Front end
- Back end
- Database

Front End Architecture

Description: The frontend of the Movie Review and Information System serves as the user interface layer, responsible for presenting movie information, facilitating user interactions, and providing a seamless browsing experience.

Technologies Used: React, React Router DOM, Axios, HTML, CSS, JavaScript, Node.js

Key Components:

- **UI Components:** Various UI components such as cards, lists, modals, and forms are utilized to create an intuitive and visually appealing interface for users to interact with.
- **Routing:** React Router DOM is employed for defining and managing routes within the application, enabling navigation between different pages or components based on URL changes.
- **API Integration:** Frontend components interact with the backend API endpoints using Axios to fetch movie data, submit user interactions, and retrieve updates.
- **State Management:** State management libraries like useState and useContext are utilized to manage application state, such as search queries, movie details, user authentication status, and session data.
- **Styling:** CSS is used to style UI components and create a cohesive design system that enhances the visual appeal and usability of the application.
- **Media Playback:** React Player component is employed for playing media files such as trailers, providing users with a seamless viewing experience directly within the application.
- **Local Storage:** HTML5 Local Storage is utilized to store user session data and preferences locally on the user's device, enhancing the efficiency of subsequent visits and interactions with the application.

Back End Architecture

Description: The backend of the Movie Review and Information System serves as the data processing and management layer, responsible for handling API requests, interacting with the database, and executing business logic.

Technologies Used: Spring Boot, Spring Web, Spring Data JPA, Speedment, Java, MySQL

Key Components:

- **RESTful API:** Spring Boot is utilized to create RESTful API endpoints that enable communication between the frontend and backend components of the application. Endpoints are defined using annotations such as @RestController, @RequestMapping, and @GetMapping.
- **Data Access Layer:** Spring Data JPA is employed to interact with the MySQL database, providing CRUD (Create, Read, Update, Delete) operations for managing movie information, user data, and interaction records. Speedment is used to generate type-safe Java APIs for interacting with the database entities, reducing boilerplate code and enhancing development productivity.
- **Business Logic:** Java components encapsulate the application's business logic, including authentication, authorization, data validation, and error handling. Dependency Injection is utilized to manage component dependencies and promote modularity and testability.
- **CORS Configurations:** Cross-Origin Resource Sharing (CORS) is configured to allow cross-origin requests from the frontend, enabling seamless communication between the client-side and server-side components of the application.
- **Security:** Spring Security is employed to implement authentication and authorization mechanisms, ensuring that only authenticated users can access protected resources and perform authorized actions within the application. Encryption techniques are utilized to secure sensitive data, such as user credentials and session tokens.

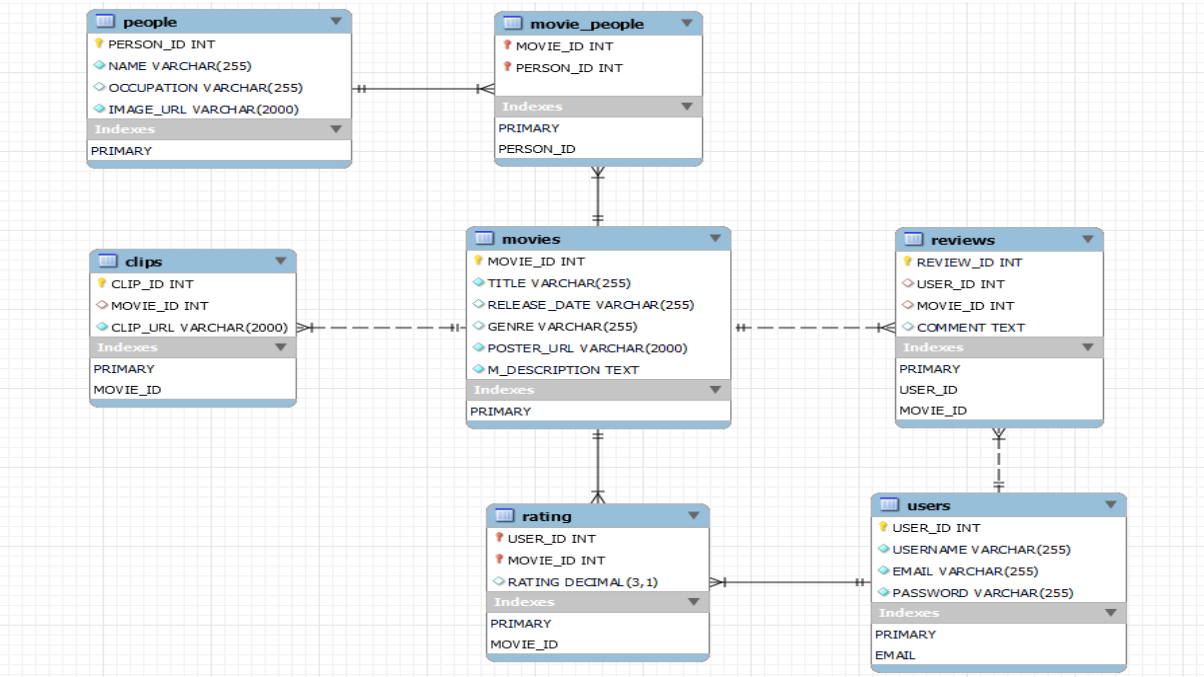
Database Architecture

Description: The storage layer of the Movie Review and Information System is responsible for storing and managing structured data related to movies, users, comments, ratings, and interactions.

Technologies Used: MySQL, AWS EC2, Security Groups, PuTTY

Key Components:

- **Relational Database:** MySQL is employed as the relational database management system (RDBMS) for storing structured data in tables with defined relationships and constraints. The database schema is designed to efficiently store and retrieve movie information, user profiles, comments, ratings, and other relevant data.
- **Cloud Hosting:** AWS EC2 is utilized to host the MySQL database instance, providing scalable computing capacity in the cloud. Security Groups are configured to control inbound and outbound traffic to the database, ensuring network security and compliance with data protection standards.
- **Secure Communication:** PuTTY is used as an SSH client to securely connect to the AWS EC2 instance over SSH protocol, enabling administrators to remotely access and manage the MySQL server. Encryption techniques are employed to safeguard data transmitted between the client and server, preventing unauthorized access and tampering.

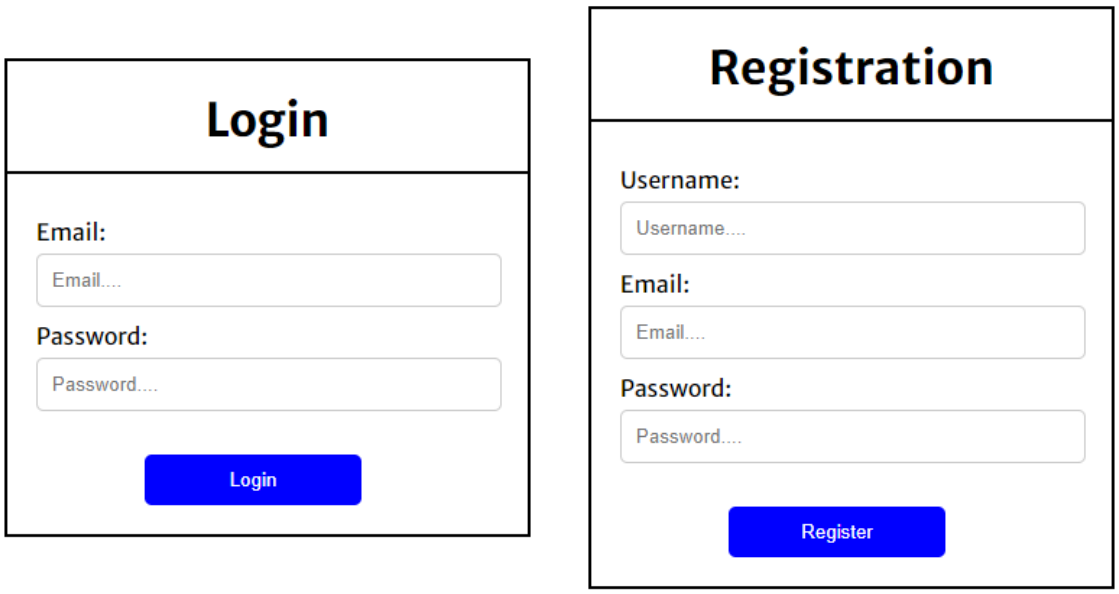


Design

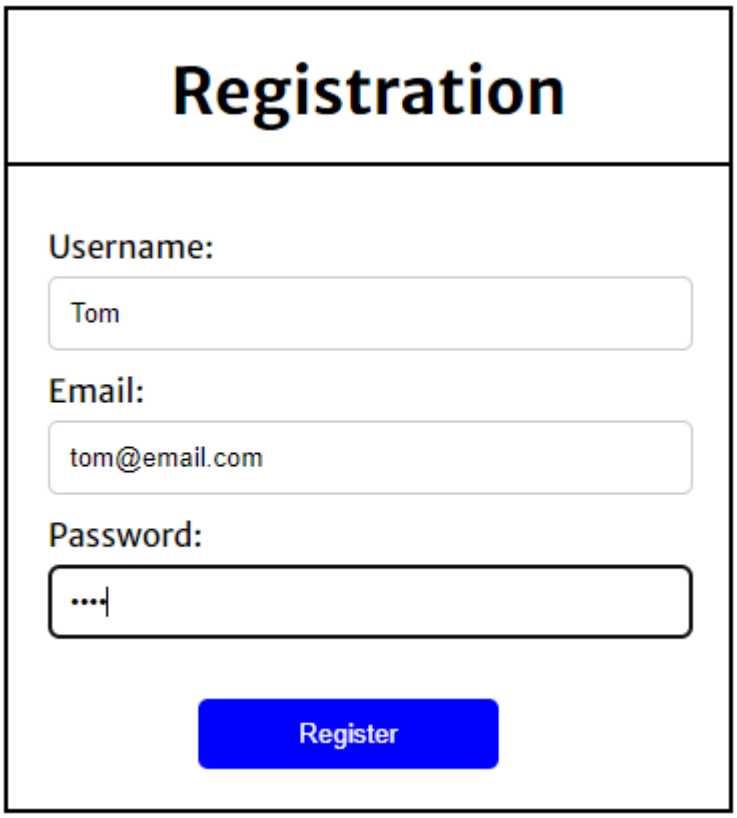
The Movie Review and Information System is designed to provide users with a comprehensive platform for exploring, interacting with, and discussing movies. The application follows a client-server architecture, with the frontend developed using React.js for a dynamic and interactive user interface, and the backend implemented using Spring Boot to handle API requests, data retrieval, and database management. MySQL is used as the relational database management system for storing and managing structured data related to movies, users, comments, ratings, and interactions.

Registration Page:

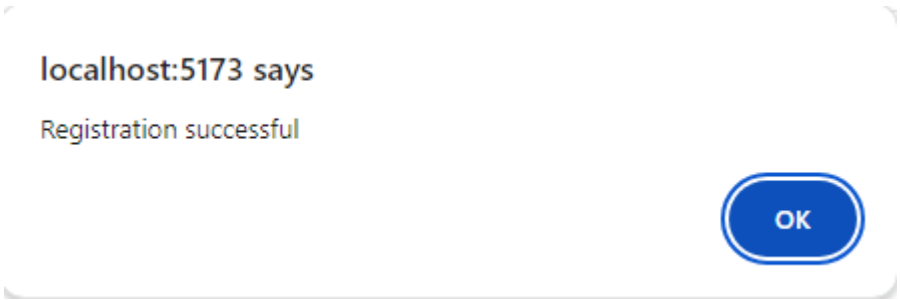
When the application starts Registration/Login page is displayed.



This page is used to register the new users as well as login form is provided for the existing users.



To register User input their details. Once they hit the Register button a notification is displayed telling the user if the registration is successful or not.

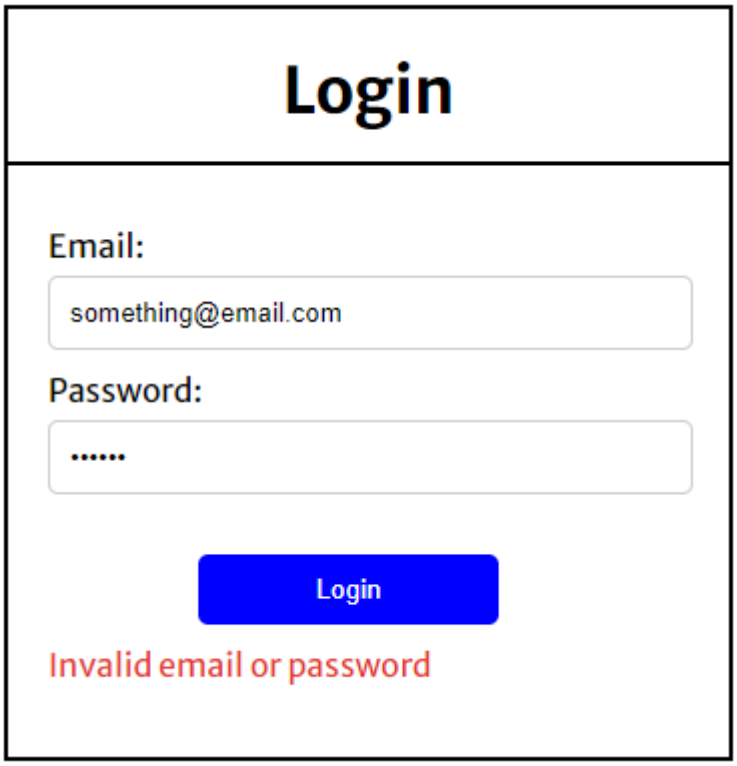


If the user is registered successfully, user ca uses the provided details to log into the system.

And User record is created in database.

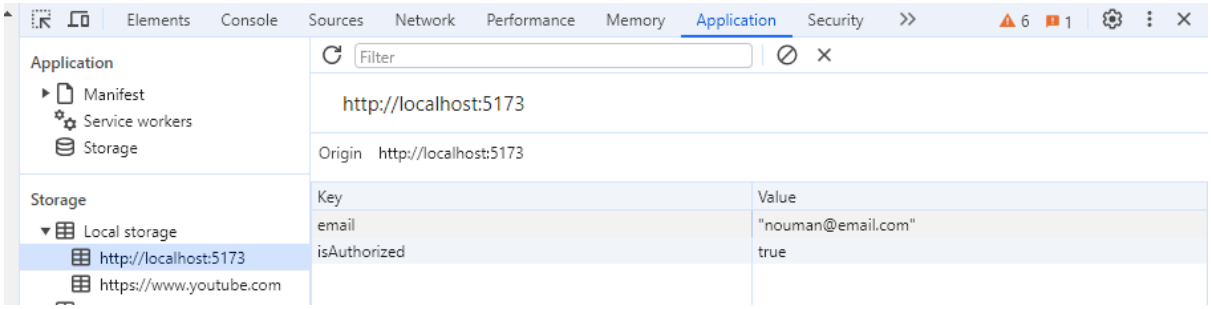
USER_ID	USERNAME	EMAIL	PASSWORD
1	Alice	alice@email.com	password123
2	Bob	bob@email.com	secret456
3	nouman	nouman@email.com	nouman1
4	Tom	tom@email.com	tom1

If the entered details aren’t correct the error message is displayed.

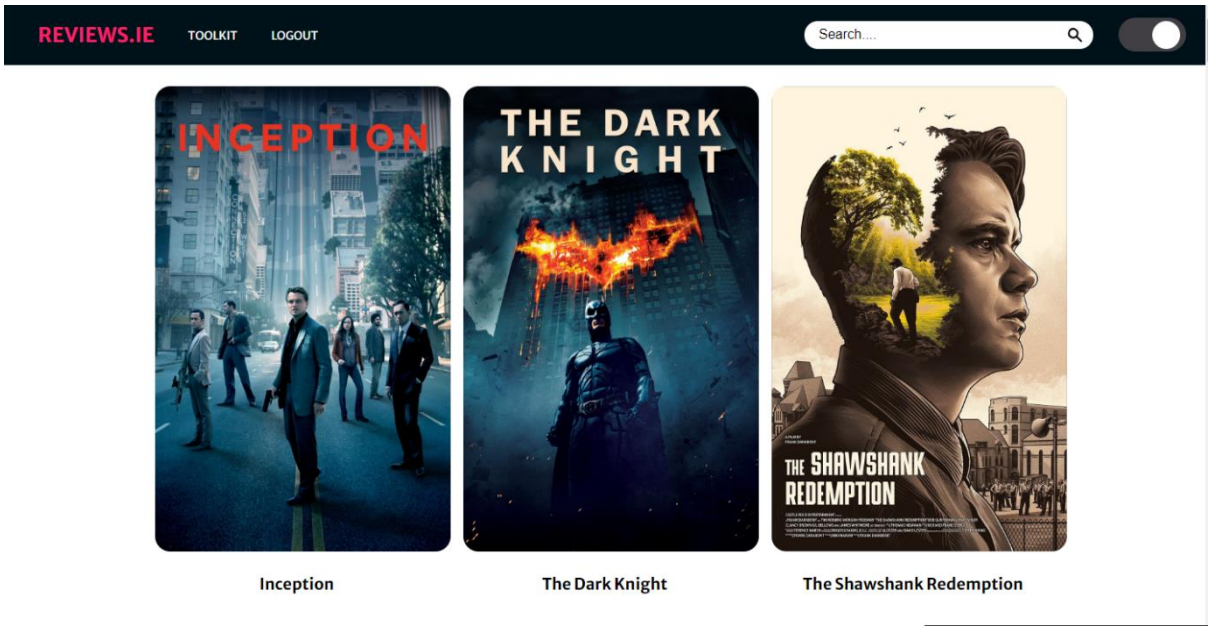


If the entered details are correct user will be navigated to the Home Page of the application.

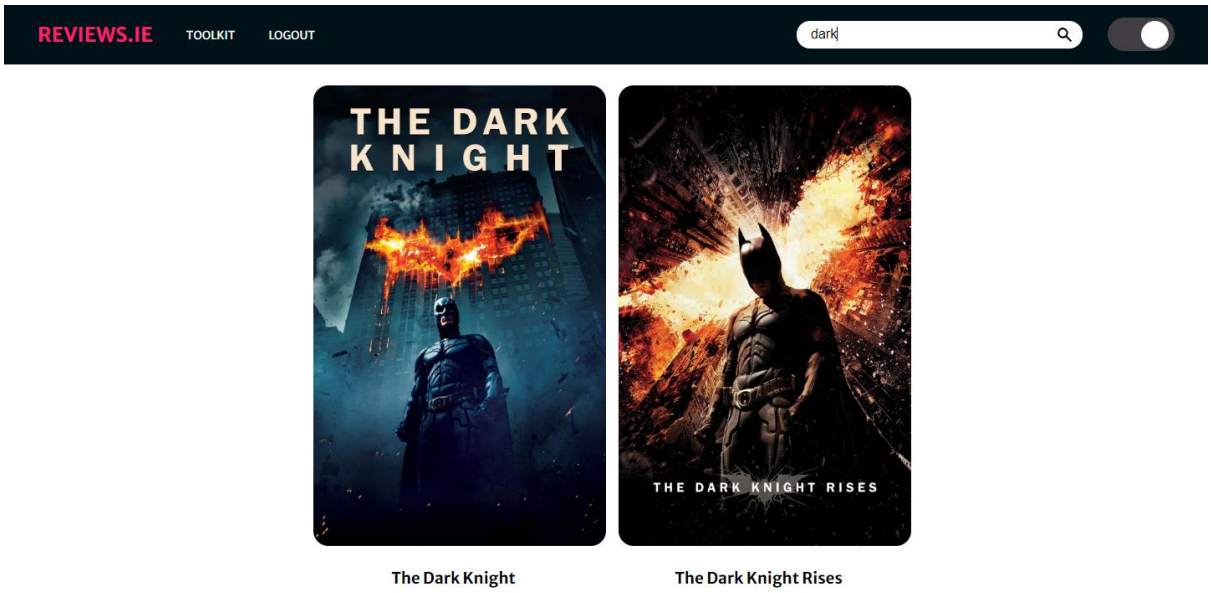
And their some details are stored in the local storage of the browser.



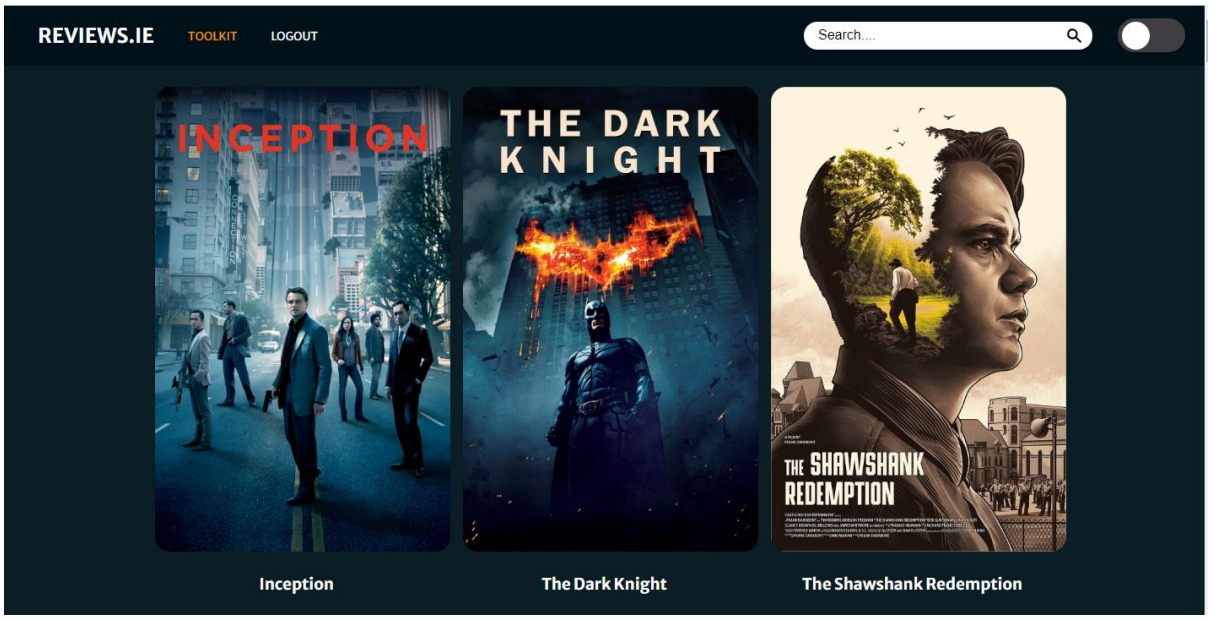
Home Page:



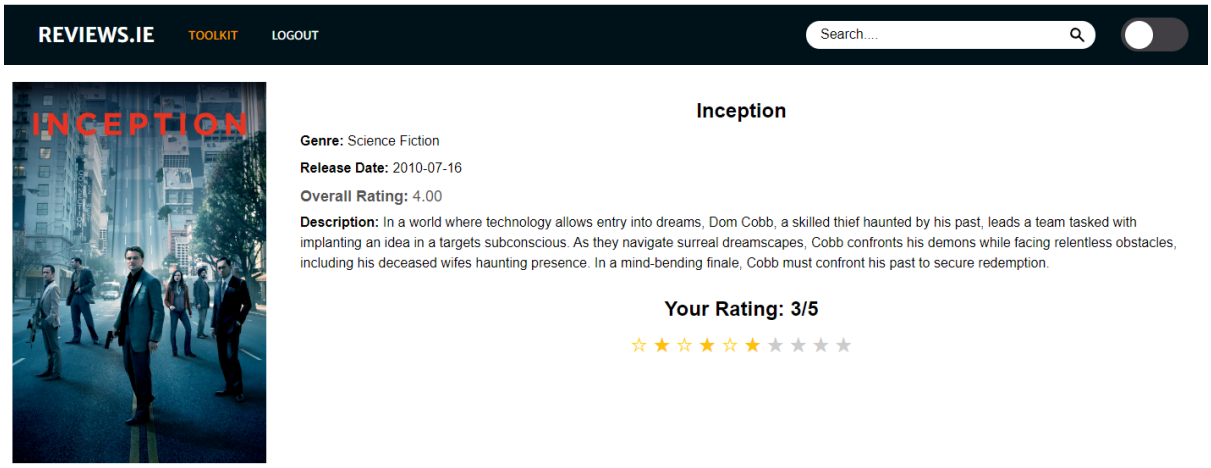
User can search any movie by typing the title as the characters are entered in the search bar with each character a get request is sent to backend and backend then talks to SQL Server and return the results e.g.



Dark mode of the application:



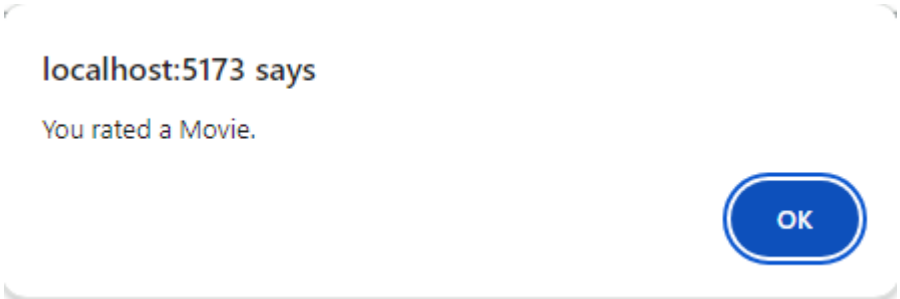
Once clicked on any movie, user will be navigated to a new page with all the related information about the selected movie.



If user give rating through clicking on the stars, if the record doesn't exist for the user a new record will be created, if it already exists the existing record will be updated.

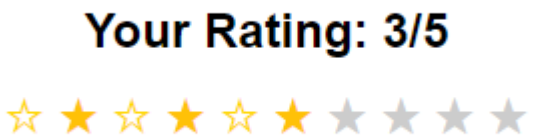
USER_ID	MOVIE_ID	RATING
1	101	5.0
1	104	2.0
1	107	4.5
2	101	4.0
2	103	3.0
2	105	1.0
2	106	5.0
3	101	3.0
3	106	3.5
3	133	3.0

As user rate a movie a notification is displayed to ensure user that their rating has been stored in database.



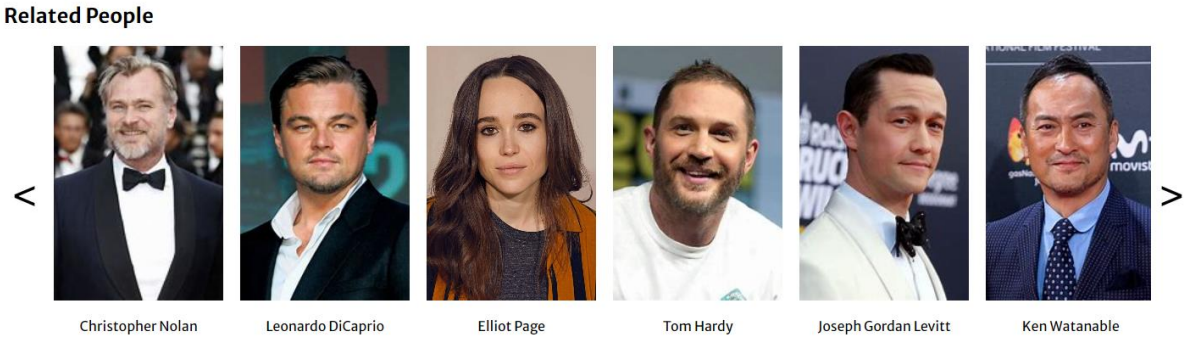
Overall Rating before user interaction.

And then user will be able to see their own rating, the glowing yellow stars indicate their own rating given to a specific movie.



With the user input of rating changes the overall rating.

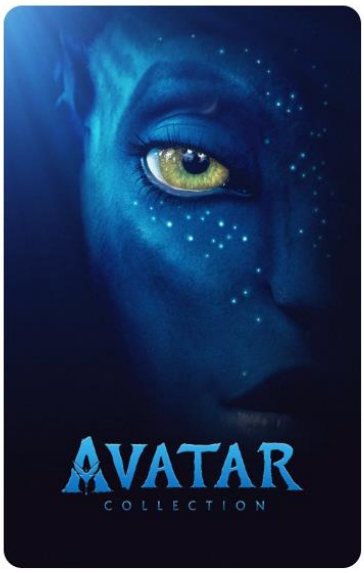
Related people to that selected movie are displayed in a sliding bar.



If user click on any of the related people it will navigate to another page and display all the movies they have worked in as in Actor/Director. E.g. if a user clicks on Elliot Page

Inception

The Lion King



Avatar

Next section on Details page is Related Movies, these movies are related to the selected movies as in these have the same genre as the one selected.

Related Movies

The Matrix

Interstellar

Jurassic Park

Back to the Future

When clicked on related movie, it will trigger and update the current page with the details of selected related movie.

The Matrix

Genre: Science Fiction

Release Date: 1999-03-31

Overall Rating: 4.25

Description: "The Matrix" is a groundbreaking science fiction film directed by the Wachowskis. Set in a dystopian future where humanity is unknowingly trapped in a simulated reality called the Matrix, the story follows computer hacker Neo, played by Keanu Reeves, as he discovers the truth about the world around him and joins a rebellion against the machines that control humanity. With its innovative visual effects, mind-bending plot twists, and philosophical themes exploring the nature of reality and free will, "The Matrix" has become a cultural phenomenon and a landmark in the sci-fi genre.

Your Rating: 3.5/5

☆ ★ ☆ ★ ☆ ★ ☆ ★ ★ ★

Next section on page is related Clips.

Related Clips

These clips are displayed using sliding scroll bar. These videos can be played.

Next section is Comment section where user can type and post any comment regarding the move.

Comments

Alice
Mind-bending plot!

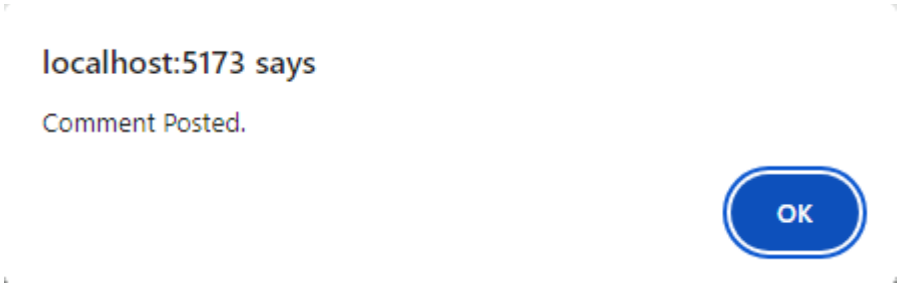
Alice
One of the best movies ever made!

Write your comment.....

Post

User can type and post the comment.

Once they post the comment a notification is displayed letting user know if it was posted successfully or not.



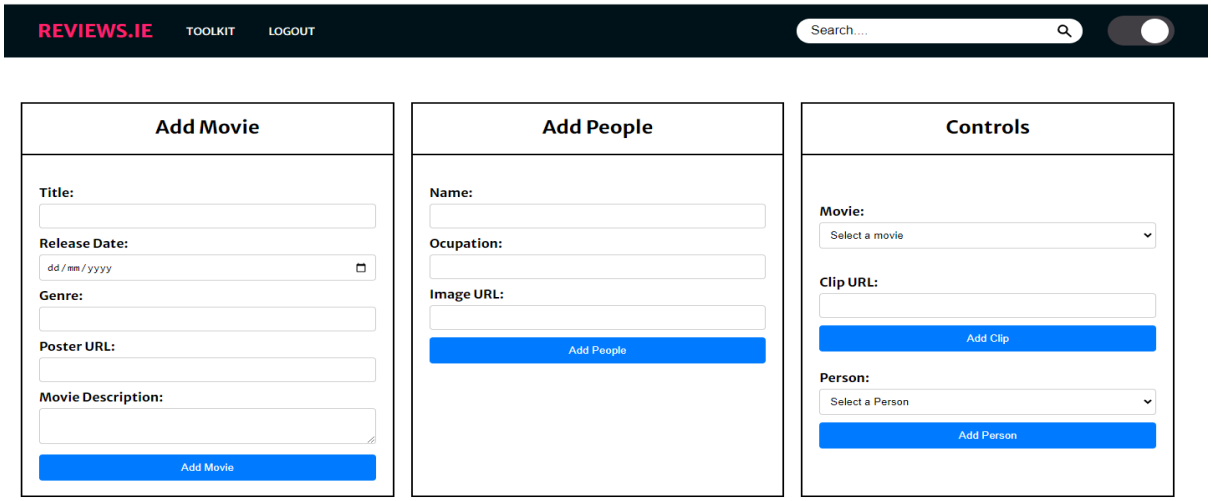
This newly added Comment can be seen in database.

REVIEW_ID	USER_ID	MOVIE_ID	COMMENT
3001	1	101	Mind-bending plot!
3002	2	102	Heath Ledger was phenomenal.
3003	1	101	One of the best movies ever made!
3004	3	101	I want to add comment.....
3005	3	101	best movie of the decade
3006	3	101	yeah
3007	3	101	timer for the scroll to appear
3008	3	101	now

As well as on the web page too.

If there are more than 10 comment sin the comment section it will create the Vertical scroll bar.

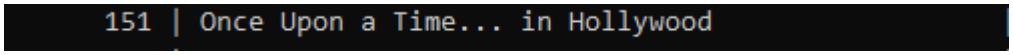
There is a Toolkit page where user can manipulate the data in database and perform some actions.



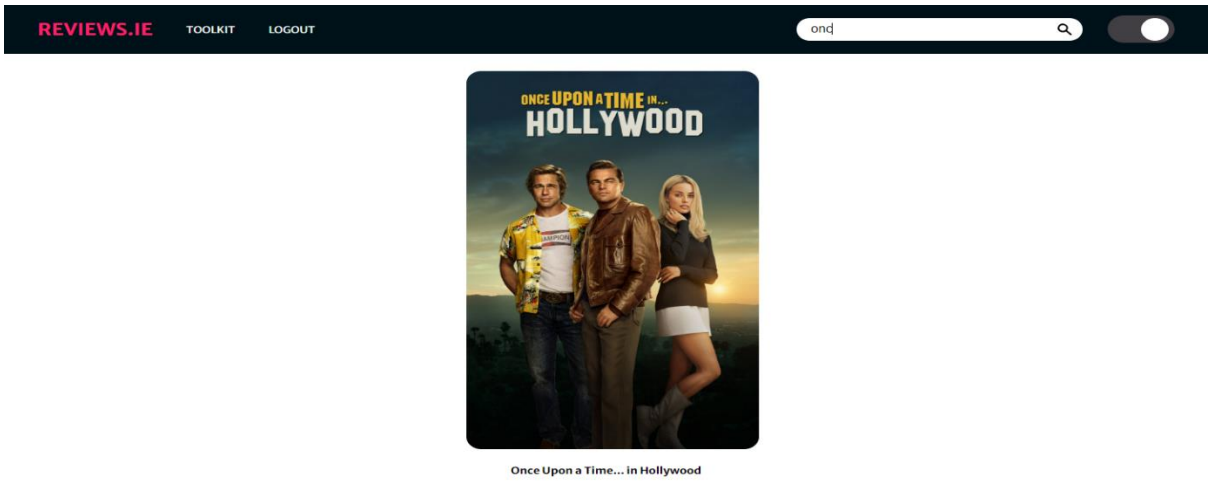
To add movie user, fill all the required section on the Add Movie Form.

As they hit the Add Movie button a notification is displayed to show if movie is added successfully or not.

This new movie can be seen in the database too.



And on the web page too.



New movies details are also added in the database.

Once Upon a Time... in Hollywood

Genre: comedy

Release Date: 2019-07-26

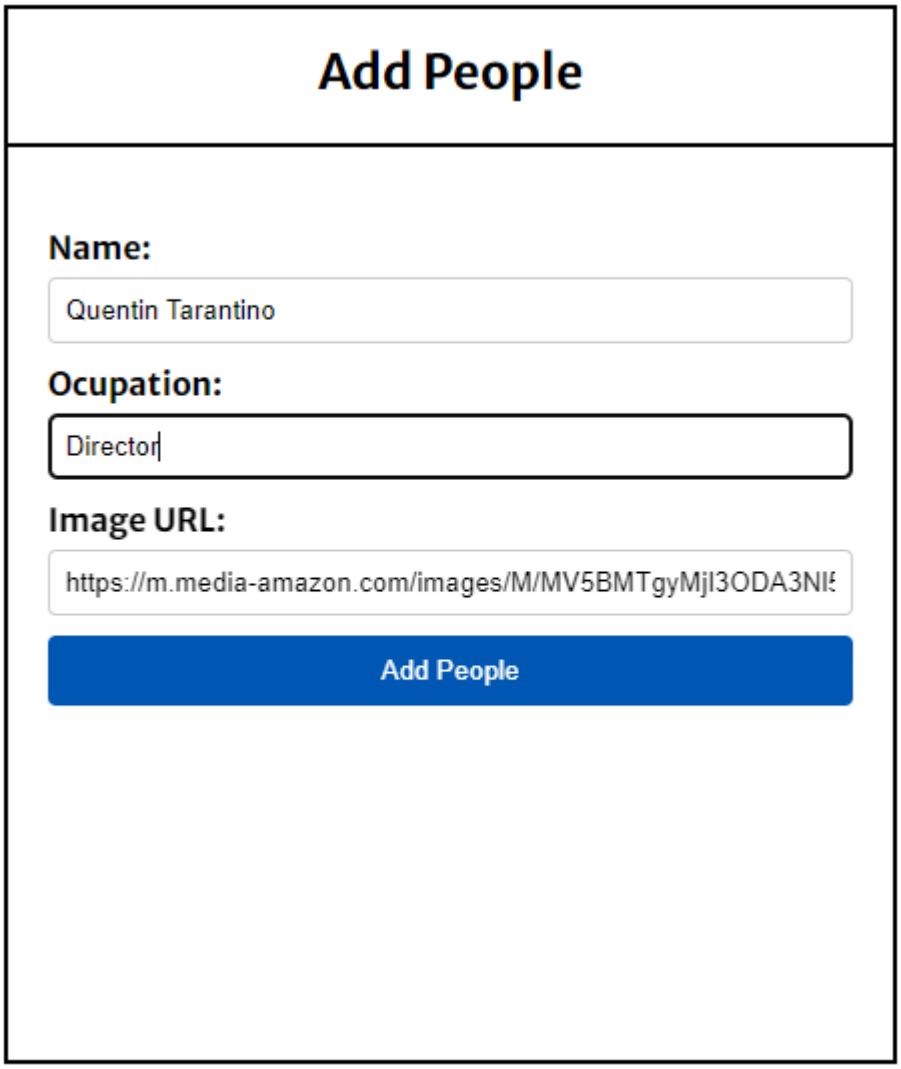
Overall Rating: 0.00

Description: Once Upon a Time... in Hollywood is a sprawling epic that transports viewers to the vibrant and turbulent world of 1969 Los Angeles. Directed by Quentin Tarantino, the film intricately weaves together the lives of fictional characters with real historical events and figures. At its core are Rick Dalton, a fading television star grappling with his career's decline, and his loyal stunt double, Cliff Booth. As they navigate the shifting landscape of Hollywood, encountering the likes of Sharon Tate and the looming presence of the Manson Family, the film delves into themes of fame, friendship, and nostalgia. Tarantino's meticulous attention to detail captures the essence of the late 1960s, from the neon-lit streets of Hollywood to the sun-soaked vistas of Los Angeles. Through a blend of comedy, drama, and historical fiction, "Once Upon a Time... in Hollywood" immerses viewers in a world of glamour, decadence, and uncertainty. With its sprawling narrative and ensemble cast led by Leonardo DiCaprio, Brad Pitt, and Margot Robbie, the film offers a rich and immersive cinematic experience that both entertains and provokes reflection on the allure and fragility of fame.

Your Rating: 0

★ ★ ★ ★ ★ ★ ★ ★ ★ ★

In similar way a new Person can be added too.

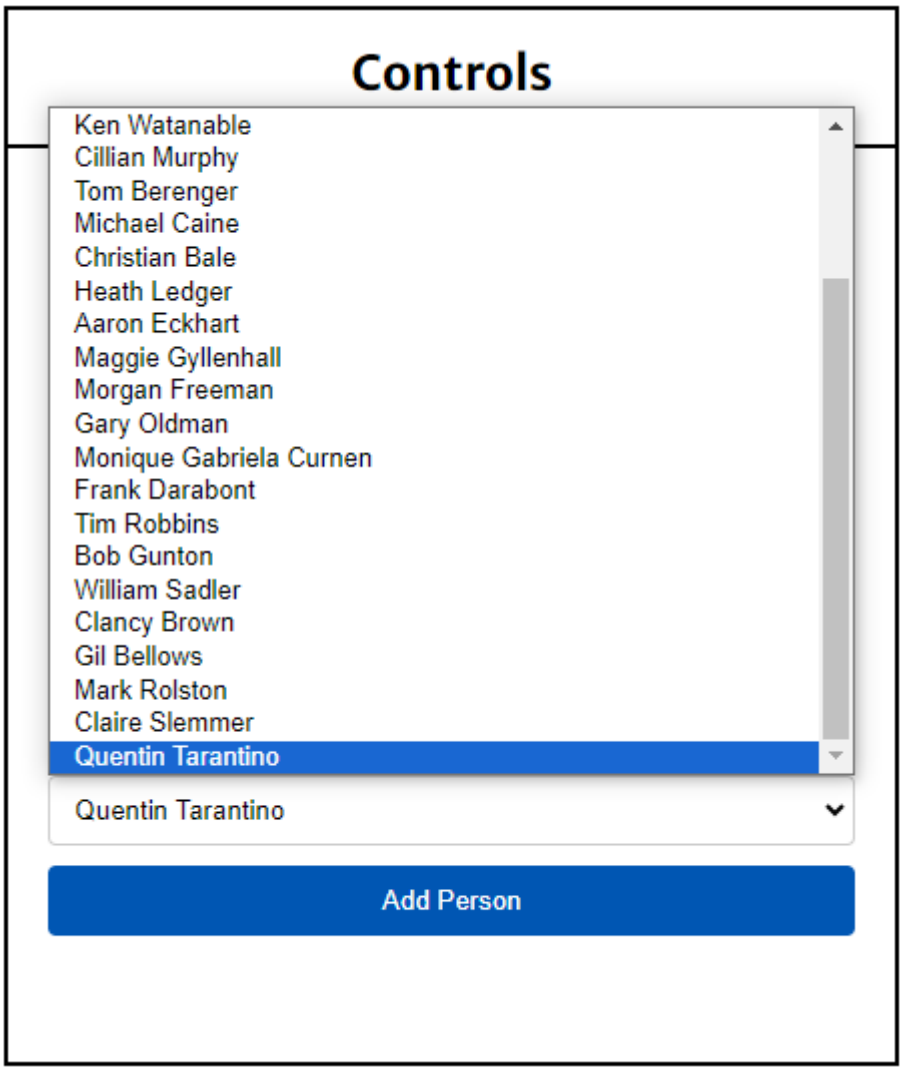


Notification:

Can be seen in database.

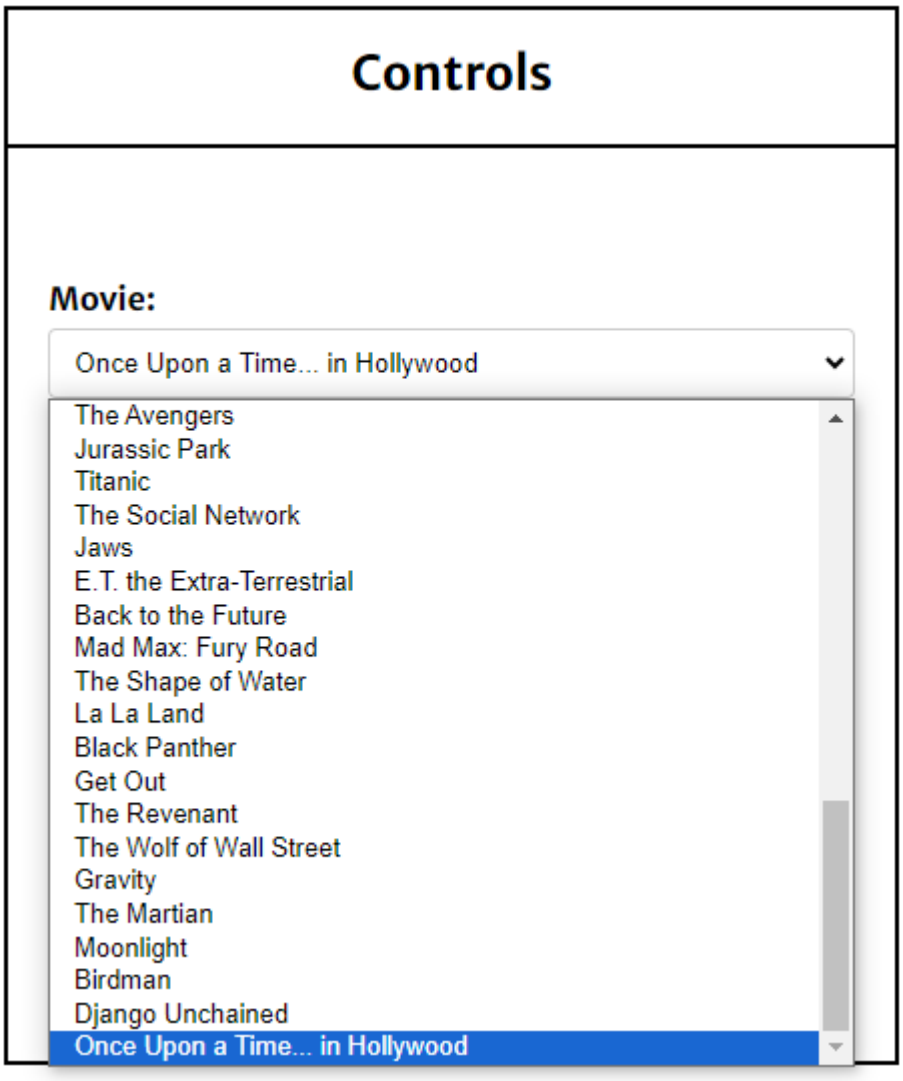
```
5025 | Quentin Tarantino      | Director   | https://m.media-amazon.com/images/M/MV5BMTgyMjI3ODA3Nl5BMl5BanBnXkFtZTcwNzY2MDYxOQ@@._V1_.jpg
```

And as soon its added in the database the web page is updated too with the new information of data.



If user try to link a person to movie but movies is not selected an error will be displayed.

These newly added person can be linked to existing movies.



If it's successful a user is notified.

Once it’s added it can be seen in the database as well as on the web page too.

Related People

Quentin Tarantino

In Similar way a clip can be linked to a specific movie.

Controls

Movie:

Once Upon a Time... in Hollywood

Clip URL:

https://www.youtube.com/watch?v=ELeMaP8EPAA&pp=ygUdb25

Add Clip

Person:

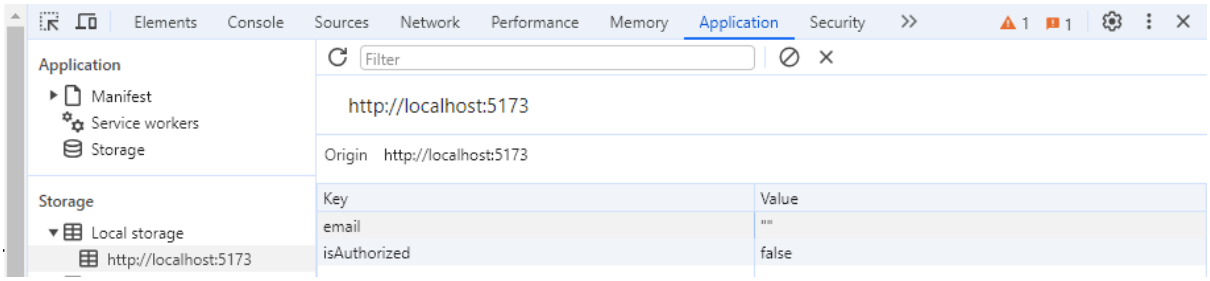
Select a Person

Add Person

User is notified.

And it can be seen on the details page of the movie.

Once the user is Logged off all the related info from the local storage is erased.



Implementation

The Movie Review and Information System was implemented following the design outlined in the architecture section. Here are the key steps and details of the implementation process:

- **Frontend Development:** The frontend of the application was developed using React.js along with other necessary libraries such as React Router DOM for routing, Axios for making HTTP requests to the backend, and React Player for media playback. The UI components were designed to provide an intuitive and visually appealing user interface for interacting with movie information.
- **Backend Development:** Spring Boot was chosen for the backend development due to its robust features for building RESTful APIs and handling data persistence. Spring Data JPA was used to interact with the MySQL database, and Speedment was integrated to generate type-safe Java APIs for database access. Security features such as authentication and authorization were implemented using Spring Security to ensure data privacy and secure communication.
- **Database Setup:** MySQL was used as the relational database management system for storing movie information, user data, comments, ratings, and interactions. The database schema was designed to efficiently organize and retrieve data, with appropriate relationships and constraints defined between tables. AWS EC2 was utilized to host the MySQL database instance, and security groups were configured to control inbound and outbound traffic to the database.
- **User Registration/Login:** A registration page was implemented to allow new users to register themselves by providing necessary details such as username, email, and password. Existing users could log in using their credentials, with user authentication handled securely using Spring Security.
- **Movie Exploration:** Users could explore movies by searching for titles or browsing categories such as popular, top-rated, or recently released. The frontend sent requests to the backend API endpoints, which retrieved relevant movie information from the database and returned it to the client for display.

- **User Interaction:** Features such as comments and ratings were implemented to enable user interaction with the application. Users could post comments, rate movies, and view their own ratings reflected in the overall rating of the movie. User input was stored in the database, and notifications were displayed to confirm successful interactions.
- **Advanced Functionality:** A Toolkit page was provided for users who wanted to contribute to the database by adding new movies, linking individuals to specific movies, or updating movie information. Forms were implemented to collect user input, and backend API endpoints were created to handle data manipulation operations.
- **Notification System:** A notification system was implemented to provide feedback to users on the outcome of their actions, such as successful registration, posting comments, or adding movies to the database. Notifications were displayed in real-time to enhance the user experience.
- **Testing and Debugging:** The application underwent rigorous testing and debugging to ensure that all features functioned as intended and that user interactions were handled smoothly. Unit tests were written to validate the functionality of backend APIs, and end-to-end testing was performed to simulate user interactions and verify system behaviour.

Overall, the implementation of the Movie Review and Information System involved collaboration between frontend and backend developers, database administrators, and system administrators to create a seamless and feature-rich platform for movie enthusiasts to explore, interact, and engage with their favourite films.

Testing

The following type of testing was included through out the making if this project

- Unit Testing
- Integration Testing
- End-to-end Testing

Unit Testing: Unit testing was conducted extensively for both frontend and backend components of the Movie Review and Information System. Frontend unit tests were written using Jest and React Testing Library, while backend unit tests were written using JUnit. Each individual function, method, and component was tested in isolation to ensure its correctness and functionality.

Integration Testing: Integration testing was performed to validate the interactions between frontend and backend components. This involved testing the integration points where different modules or layers of the application connect. Integration tests were written using tools like postman for frontend testing and Spring Boot's testing framework for backend testing.

End-To-End Testing: End-to-End testing scenarios were executed to simulate real user interactions and workflows. This included testing user registration, movie search, commenting, rating, and other key features of the application. End-to-End tests were automated using tools like Selenium and were performed on various browsers and devices to ensure cross-platform compatibility.

This comprehensive testing approach ensures that the Movie Review and Information System meets high standards of quality, reliability, security, and accessibility, providing users with a seamless and enjoyable experience.

Evaluation

Restating Objectives: The original objectives of the Movie Review and Information System project were to provide users with a comprehensive platform to explore and interact with a vast database of movies, covering various genres, release dates, and ratings. The application aimed to help users discover, analyse, and discuss their favourite films while also providing insights into the movie-making process and fostering a sense of community among movie enthusiasts.

Achievement of Objectives: The project has successfully achieved its objectives. Users are able to explore a wide range of movies through intuitive search and browsing functionalities. The application provides detailed information about each movie, including synopsis, cast, crew, ratings, and trailers, facilitating informed decision-making for users. The inclusion of user interaction features such as comments and ratings has fostered a lively community of movie enthusiasts within the platform.

Evaluation of Requirements: Both functional and non-functional requirements have been effectively addressed in the project. Functional requirements such as movie exploration functionality, display of movie information, user interaction features, and advanced functionality like the Toolkit page and registration have been implemented as per specifications. Non-functional requirements such as performance, usability, and security have also been given due consideration, ensuring a responsive, user-friendly, and secure experience for users.

Appropriateness of the Solution: The solution provided by the Movie Review and Information System is highly appropriate for its intended purpose. It fulfils the needs of movie enthusiasts by offering a rich and engaging platform for discovering, exploring, and discussing movies. The integration of user interaction features, advanced functionalities, and comprehensive movie information ensures that the application caters to a diverse audience with varied interests and preferences. Overall, the solution aligns well with the project objectives and effectively addresses the requirements of its target users.

Conclusion

In conclusion, the Movie Review and Information System represents a comprehensive and user-friendly platform for movie enthusiasts to explore, interact with, and appreciate the world of cinema. By restating the original objectives of the project and evaluating its outcomes, it's evident that the system has successfully achieved its goals.

The project effectively addresses both functional and non-functional requirements, providing users with intuitive functionalities for movie exploration, detailed information display, and rich user interaction features. The design and technology decisions, including the use of React.js for the frontend and Spring Boot for the backend, have proven to be appropriate, resulting in a robust and scalable architecture.

Looking back, the chosen solution stands as a testament to its appropriateness for the intended purpose. It offers a seamless experience for users, enabling them to discover, analyse, and discuss their favourite films while fostering a sense of community within the platform. Overall, the Movie Review and Information System serves as the ultimate destination for all things cinema, catering to the diverse interests and preferences of movie enthusiasts worldwide.

References

Database:

AWS Console: https://docs.aws.amazon.com/ec2/?icmpid=docs_homepage_featuredsvcs

MySQL: https://dev.mysql.com/doc/

SQL Server on Ubuntu: https://docs.rackspace.com/docs/install-mysql-server-on-the-ubuntu-operating-system

PuTTY Docs: https://documentation.help/PuTTY/

Frontend:

Angular Documentation: https://angular.io/docs

React docs: https://legacy.reactjs.org/docs/getting-started.html

Axios: https://axios-http.com/docs/intro

Node.js: https://nodejs.org/en/learn/getting-started/introduction-to-nodejs

Backend:

Maven: https://maven.apache.org/guides/index.html

Spring boot: https://spring.io/projects/spring-boot

Speedment: https://speedment.github.io/speedment-doc/

Appendix

You can check my code for all the parts on my GitHub Repository.

https://github.com/NoumanZafar/MoviesInfo_FinalYear