

ANGULAR PROJECT

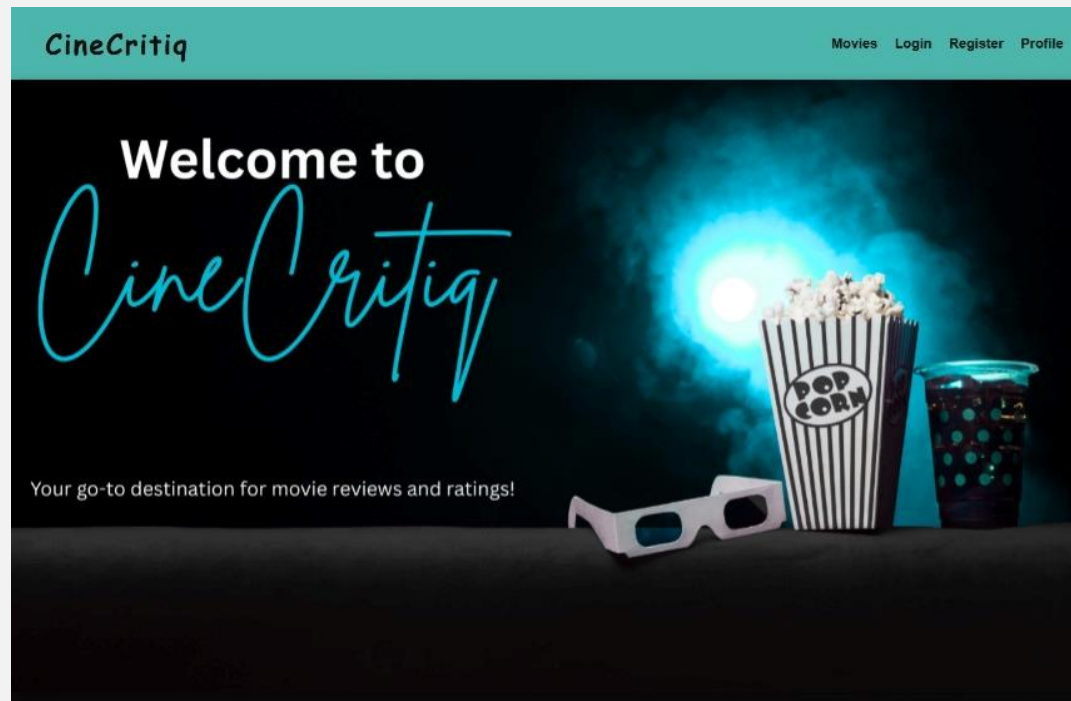
CINECRITIQ-MOVIE REVIEW WEBSITE

DONE BY:
K.Kalaivani

CINECRITIQ

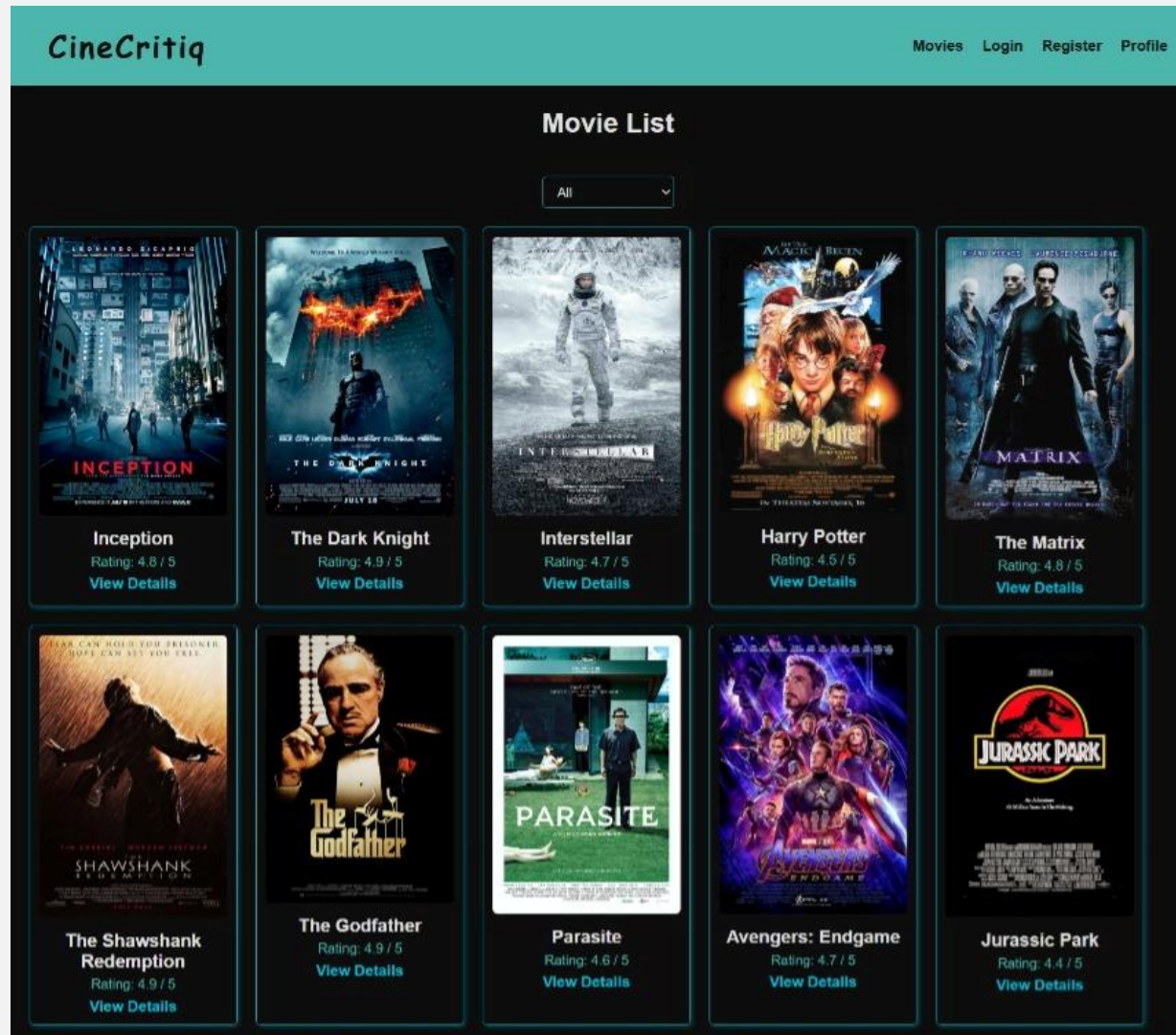
- **CineCritiq** is a dynamic movie review website designed using Angular. It provides users with a platform to share their thoughts on their favorite movies.
- The platform offers a **login** feature for returning users and a **registration** option for new users to create an account.
- Once logged in, users can access their **profile page**, where they can view and manage all the reviews they have added.
- The application efficiently manages data using services, APIs, and local storage, offering a secure and dynamic user experience.

HOME PAGE



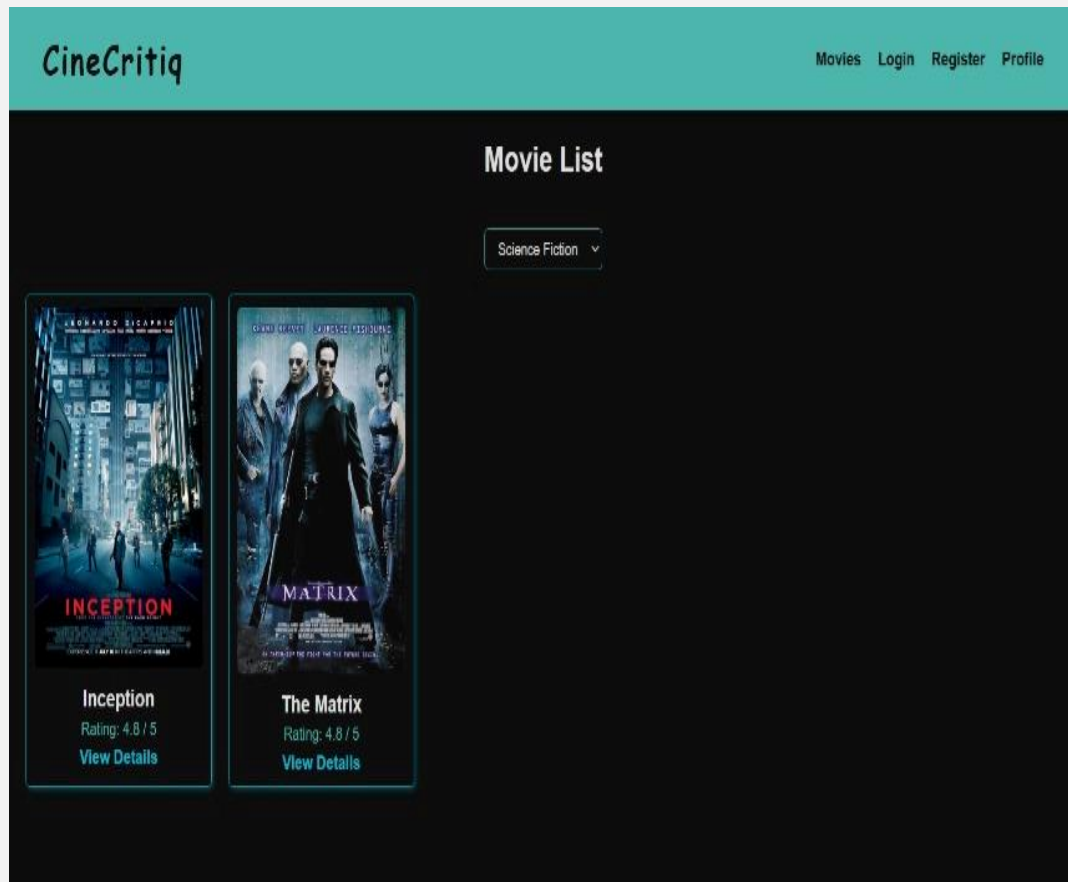
- The **Home Page** of **CineCritiq** serves as the first point of interaction for users, featuring a visually appealing banner image with a welcoming message.
- It sets the tone for an engaging movie review experience.
- The header includes the **CineCritiq** title, which acts as a navigational link—clicking on it will always bring users back to this **Home Page**.
- The design ensures easy access to essential features like **Movies**, **Login**, **Register**, and **Profile**, making navigation smooth and intuitive for users.

MOVIES PAGE



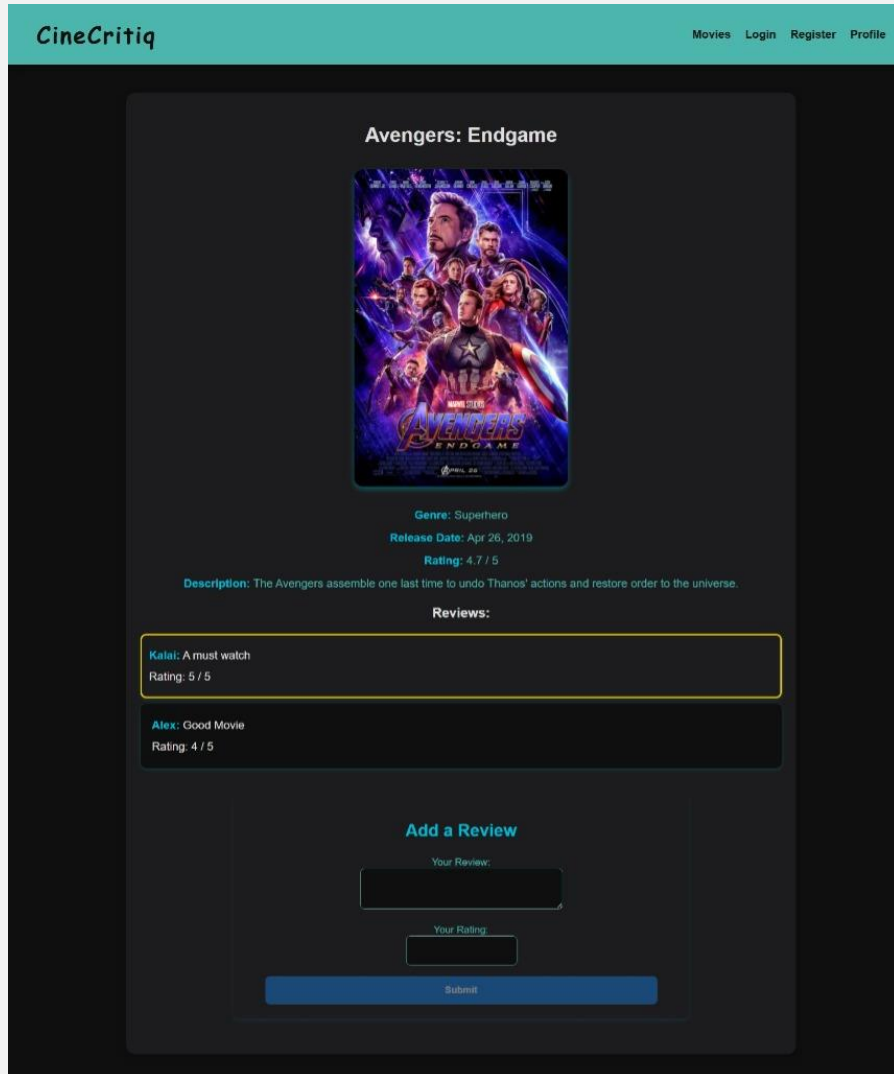
- The **Movies Page** in **CineCritiq** presents a visually structured list of movies displayed as **card components**.
- Each card features the **movie title**, **rating**, and a **"View Details"** button.
- The ratings are dynamically **fetch**ed from **locally hosted API**.
- Clicking on **"View Details"** navigates users to a **dedicated movie details page**, where they can explore in-depth information about the selected film.

GENRE FILTER PIPE



- The **Movies Page** includes a **genre filter** implemented using a **custom pipe**.
- A **dropdown menu** allows users to select a specific genre, dynamically updating the displayed movies to match the chosen category.
- Upon selecting a genre, only movies belonging to that genre are shown, providing a personalized browsing experience.

MOVIE DETAILS PAGE

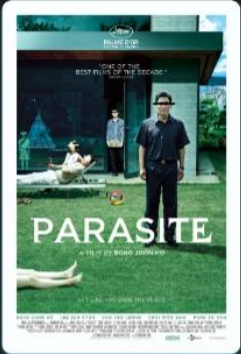


- The movie details page displays comprehensive information about a selected movie, including its **title**, **image**, **genre**, **release date**, **rating**, and **description**, all of which are dynamically fetched from API.
- The release date is formatted using Angular's **inbuilt date pipe** to ensure a clean and readable format.
- Below the movie details, the **user reviews** submitted by different viewers are listed, where a custom **ngClass directive** is used to highlight reviews with a perfect **5/5 rating** to visually distinguish top-rated feedback.
- The page also includes an interactive section that allows users to **add their own reviews and ratings**.

ADD REVIEWS

CineCritiq [Movies](#) [Login](#) [Register](#) [Profile](#)

Parasite



Genre: Thriller
Release Date: Nov 8, 2019
Rating: 4.6 / 5

Description: A poor family schemes to become employed by a wealthy family and infiltrates their household.

Reviews:
No reviews available for this movie.

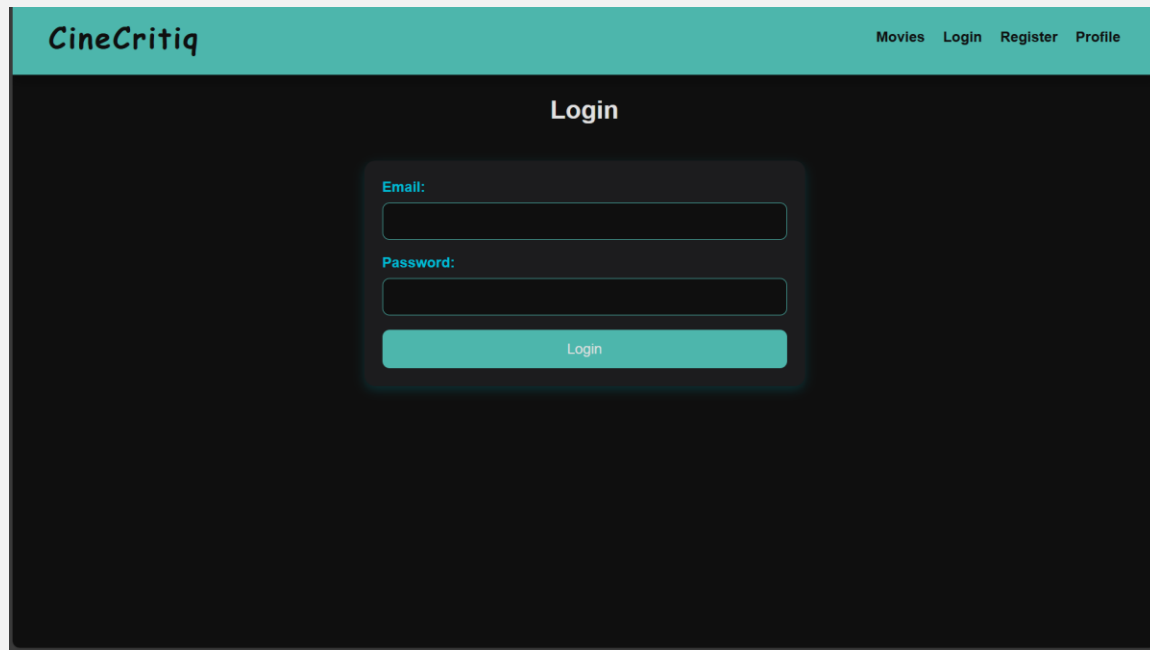
Add a Review

Your Review:

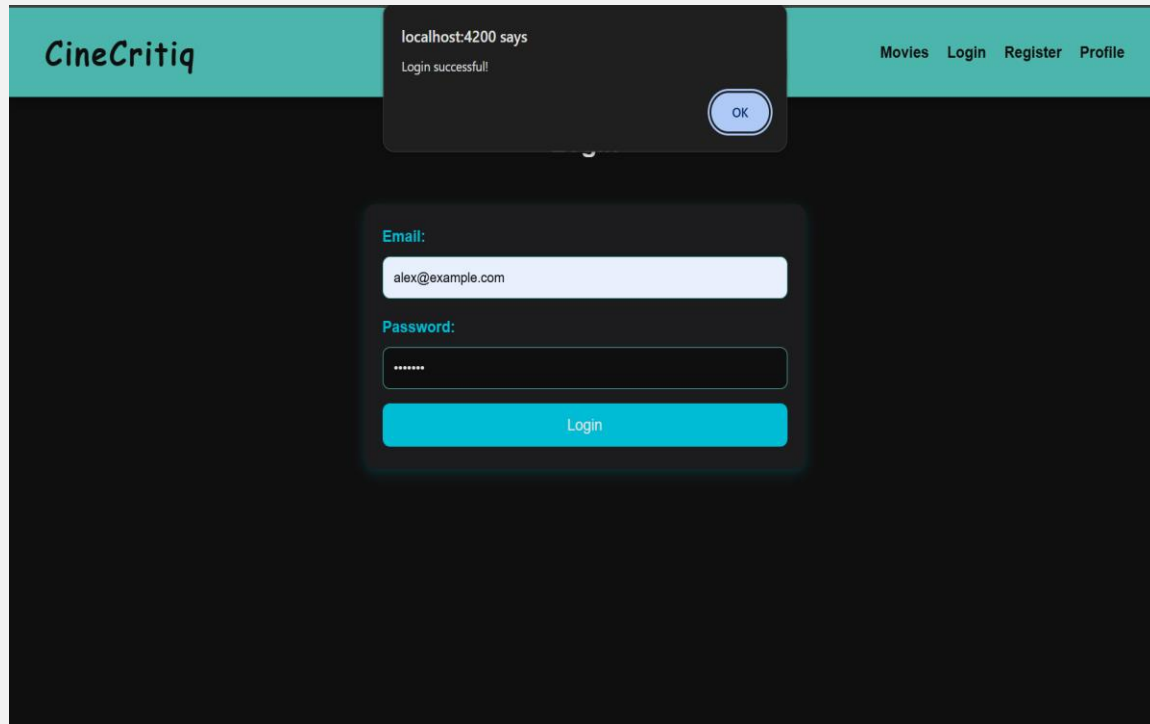
Your Rating:

Submit

- The **Add a Review** section leverages a **template-driven form** with ngModel for seamless two-way data binding.
- This form allows users to submit their feedback by entering a review and assigning a rating between 1 and 5.
- The review input is handled by a **textarea**, while the rating is captured through an **input field**.



A screenshot of the CineCritiq application's login page. The page has a dark teal header with the 'CineCritiq' logo on the left and navigation links 'Movies', 'Login', 'Register', and 'Profile' on the right. The main content area is dark gray and features a 'Login' heading. Below the heading is a form with two input fields: 'Email:' and 'Password:'. The 'Email' field is empty, and the 'Password' field is also empty. A teal 'Login' button is positioned below the password field.



A screenshot of the CineCritiq application's login page after a successful login. The page layout is identical to the previous screenshot, but with additional elements. A dark gray alert box is visible in the upper left, displaying the text 'localhost:4200 says' and 'Login successful!' with an 'OK' button. The 'Email' input field now contains the text 'alex@example.com', and the 'Password' field is filled with eight asterisks. The 'Login' button remains teal.

LOGIN PAGE

- The **Login Page** is designed with a **template-driven form** using **ngModel** for two-way data binding, allowing users to input their credentials (email and password).
- Upon clicking the **Login** button, the form validates the inputs and processes the login request.
- If the credentials are correct, an alert message stating "**Login successful!**" is displayed.

CineCritiq

Movies Login Register Profile

Register

Name:

Email:

Password:

Register

REGISTER PAGE

- The **Register Page** also uses a **template-driven form** with **ngModel** to collect user details.
- Upon successful registration, an alert **"Registration successful! You can now log in."** is shown, and the user is redirected to the login page.
- If the email already exists, an alert **"User already exists. Please log in or use a different email."** is displayed, preventing duplicate registrations.

CineCritiq

localhost:4200 says
Registration successful! You can now log in.

Movies Login Register Profile

OK

Name:

Enzo

Email:

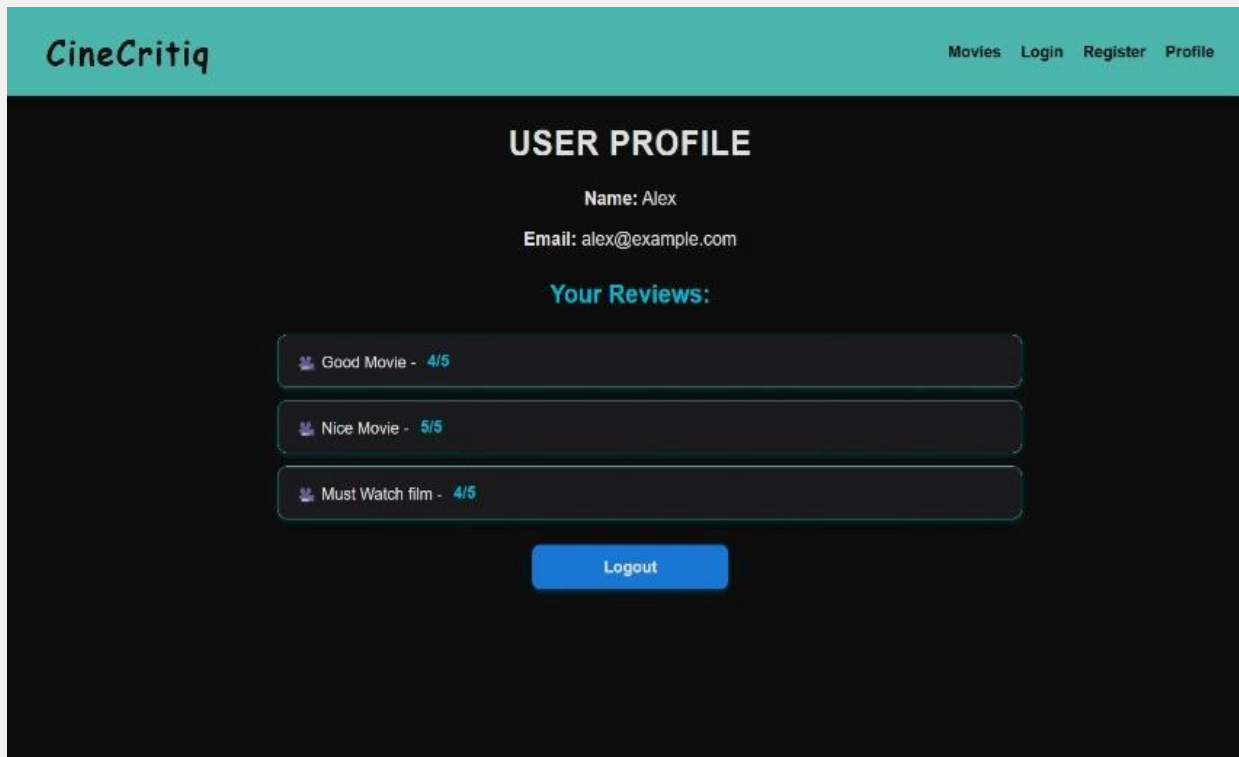
enzo@example.com

Password:

....

Register

PROFILE PAGE



- The **User Profile Page** displays the logged-in user's details, including their **name, email, and reviews**.
- It features a **Logout** button that logs the user out and redirects them to the login page.
- The page uses a **route guard** to ensure that only authenticated users can access it.
- If a user tries to access the profile page without logging in, they are automatically redirected to the **login page**.

CONCLUSION

CineCritiq project satisfies the conditions as follows:

- Components:** Separate components for home,login, register, profile, and reviews.
- Parent-Child Components:** Review acts as a child component.Movie Details acts as parent component.
- Directives:** *ngIf, *ngFor (structural), and attribute directives(ngClass) for styling.
- @Input, @Output:** Passing review data between parent and child.
- Routing and Routing Parameters:** Navigation with route parameters for dynamic content.
- Local Storage:** Storing user authentication and review data.
- Routing Guard:** Restricts profile access to logged-in users.
- Services and Observables:** API calls and data handling using services(movie service,user service) with observables.
- API Call:** Fetching movie and review data dynamically.
- Form Handling:** Login, register, and review forms using ngModel.
- Inbuilt Pipes:** Using inbuilt date pipe to display the release date.
- Custom Pipes:** Custom pipe like genre filter pipe to categorize movies based on genre.