

Front-End UI/UX Mini Project

Project Submission Template

1. Title Page

- **Project Title:** e.g., *Responsive Portfolio Website for a Freelancer*
 - **Submitted By:**
 - **Team Members:** Neha Tresa Bobby
Tania Robby
 - **Roll Number(s):** 2460411
2461032
 - **College Email ID:** neha.tresa@btech.christuniversity.in
tania.robby@btech.christuniversity.in
 - **Course:** *UI/UX Design Fundamentals*
 - **Instructor Name:** Mr. Narendra Kumar
 - **Institution:** *CHRIST (DEEMED TO BE UNIVERSITY)*
 - **Date of Submission:** 26/09/2025
-

2. Abstract

This project focuses on designing and developing a responsive Movie Review Website where users can browse a list of popular movies, read existing user reviews and ratings, and submit their own comments and star ratings.

The core functionality is implemented using HTML5 for structure, CSS3 (with Bootstrap 5) for responsive styling and layout, and JavaScript/jQuery for dynamic features like movie listing, detail views, searching, and managing the image slideshow.

The design uses a dark, cinematic theme with a dynamic, blurred background to enhance the user experience. The outcome is a functional, visually engaging front-end prototype of a modern movie review platform.

3. Objectives

- Implement a Movie Listing showing posters, titles, and descriptions.
 - Develop a Detail View to display full movie information and user reviews.
 - Create a functional Review Submission Form for users to add ratings and comments (front-end only).
 - Enable Search and Filter functionality for the movie list using JavaScript/jQuery.
 - Ensure a fully Responsive Layout across all devices using the Bootstrap framework and custom CSS media queries.
 - Incorporate a dynamic Hero Slideshow using JavaScript to showcase featured movies.
-

4. Scope of the Project

- Focused primarily on front-end design and interactivity using client-side technologies.
 - Data is stored locally within the JavaScript file (script2.js) and is not integrated with a backend server or database.
 - Utilised jQuery for DOM manipulation and event handling, and Bootstrap 5 for foundational styling and responsive components (e.g., Navbar, Modal).
 - The project is viewable on desktop, tablet, and mobile viewports, thanks to responsive design.
-

5. Tools & Technologies Used

Tool/Technology	Purpose
HTML5	Markup and content structure (index2.html).
CSS3	Styling, theme, and custom layout management (style2.css).
Bootstrap 5	Responsive grid system, navbar, and modal components.
JavaScript (jQuery)	Dynamic content rendering, event handling (click, hover), search functionality, and slideshow logic (script2.js).
VS Code	Code editor
Chrome DevTools	Testing and debugging

6. HTML Structure Overview

The page utilizes a modern HTML5 structure:

- **<header>**: Contains the fixed, semi-transparent Bootstrap Navbar with the site brand and the movie search/login elements.
 - **<div id="dynamic-bg">**: A fixed, full-page element used for the blurred, dynamic background image.
 - **<div id="hero-banner">**: The top section, featuring the **#hero-slideshow** (to display rotating featured movie backgrounds) and the **slideshow-controls**.
 - **<main>**: The main container for the page content, including:
 - **Movie Listing**: A grid structure (**#movie-list**) to display clickable **movie-card** elements.
 - **Movie Detail Section**: A hidden container (**#movie-detail-section**) that shows when a card is clicked. It includes the movie poster (**#movie-poster-detail**), detailed description, and the **Reviews** section (**#user-reviews** and **#review-form**).
 - **Modal**: A Bootstrap **#loginModal** for a non-functional login/registration form.
-

7. CSS/Bootstrap

- **External File**: Styling is managed in **style2.css**.
- **Theming**: Uses a **dark theme** with a primary background color of **#0d1926** and a primary accent color of **#3b82f6** (Bootstrap's primary blue).
- **Dynamic Background**: The full-page **#dynamic-bg** is styled with **position: fixed** and **filter: brightness(0.4) blur(4px)** to create an immersive, cinematic backdrop for the content.

- **Movie Card Grid:** Uses **CSS Grid** (`display: grid`) for a responsive and flexible layout of movie cards.
- **Movie Detail Layout:** Uses **Flexbox** (`display: flex` on `.review-section-container`) to organize the reviews and the sticky movie poster on larger screens.
- **Transitions & Effects:** Extensive use of the `transition` property for smooth hover effects on navigation links, buttons, and movie cards (e.g., `transform: translateY(-8px)`).
- **Responsiveness:** Uses **@media queries** to adapt the movie list grid and the detail section layout for mobile and tablet views.

8. JavaScript/jQuery Overview

- **Primary Tool:** All dynamic functionality is driven by jQuery (a fast, small, and feature-rich JavaScript library).
- **File Location:** All code resides in the external file, `script2.js`.
- **Execution Trigger:** Code execution is encapsulated within `$(document).ready(function() { ... })`, ensuring that the script only runs after the entire HTML structure (DOM) is fully loaded.
- **Data Source:** Movie information, including title, descriptions, poster paths, and user reviews, is stored locally in a `const movies` array within the script, acting as a mock database.
- **Core Role:** Manages DOM manipulation, handles all user interactions (events), and controls the dynamic display features like the slideshow, search results, and movie detail views.

9. Key Features

Feature	Description
Dynamic Hero Slideshow	Automatically rotates through featured movie backgrounds in the jQuery's <code>setInterval</code> and indicator controls.
Movie Listing	Displays movies in a responsive grid layout. Hovering over a card changes the dynamic background image.
Detail View	Clicking a movie card transitions the user to a detailed view, complete with the poster, title, description, and existing user reviews.
Search Functionality	A working search bar (<code>#movie-search-nav</code>) uses jQuery's <code>keyup</code> event to instantly filter the displayed movie list by title.
Review Submission	A form to submit new user reviews (rating and comment), which are dynamically added to the movie's review array and displayed immediately.
Cinematic UI/UX	Dark theme, semi-transparent navbar, and a fixed, blurred background image that changes based on the viewed movie.

10. Challenges Faced & Solutions

Challenge	Solution
Synchronizing Backgrounds	Used two background elements: a sharp, rotating image for the hero (.slide-image) and a fixed, blurred one for the full-page effect (#dynamic-bg). jQuery handles updating both when a movie is clicked/hovered
Managing Slideshow State	Implemented stopSlideshow() on mouse hover/detail view and startSlideshow() on mouse leave/return to list, ensuring smooth control over the automatic rotation.
Responsive Grid Layout	Used grid-template-columns: repeat(auto-fill, minmax(300px, 1fr)) for a flexible listing, adjusted with @media queries to ensure a proper display on smaller screens.
Text Readability on Dark/Blurred Backgrounds	Applied strong text-shadow to the main movie title and description elements to ensure maximum contrast against the potentially light parts of the background images.

11. Outcome

A visually compelling and functionally robust front-end prototype for a movie review website was successfully completed. The project leverages Bootstrap 5 for responsiveness and jQuery for all dynamic features, including the image slideshow, instant movie search, and review display. The implementation successfully meets all specified requirements, providing a clean, dark-themed interface that offers a great user experience across different devices.

12. Future Enhancements

- Local Storage/IndexedDB: Implement persistent storage so that new reviews remain after a page refresh.
 - Genre Filtering: Extend the search and filter to include movie genres.
 - Light/Dark Mode: Add a theme toggler using CSS custom properties.
 - User Authentication: Integrate with a basic authentication system to track which user submitted which review.
-

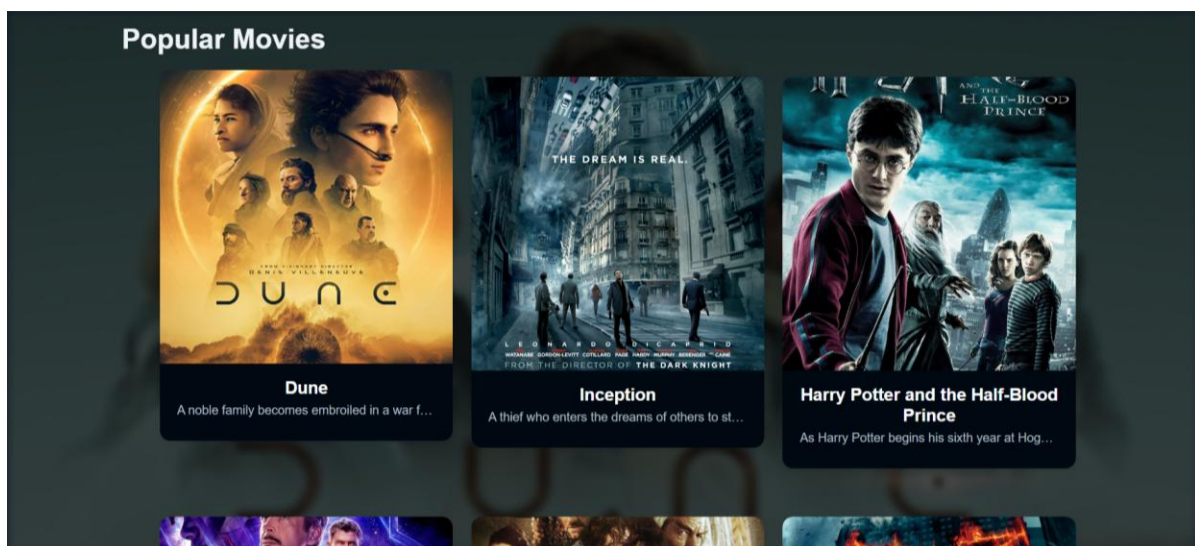
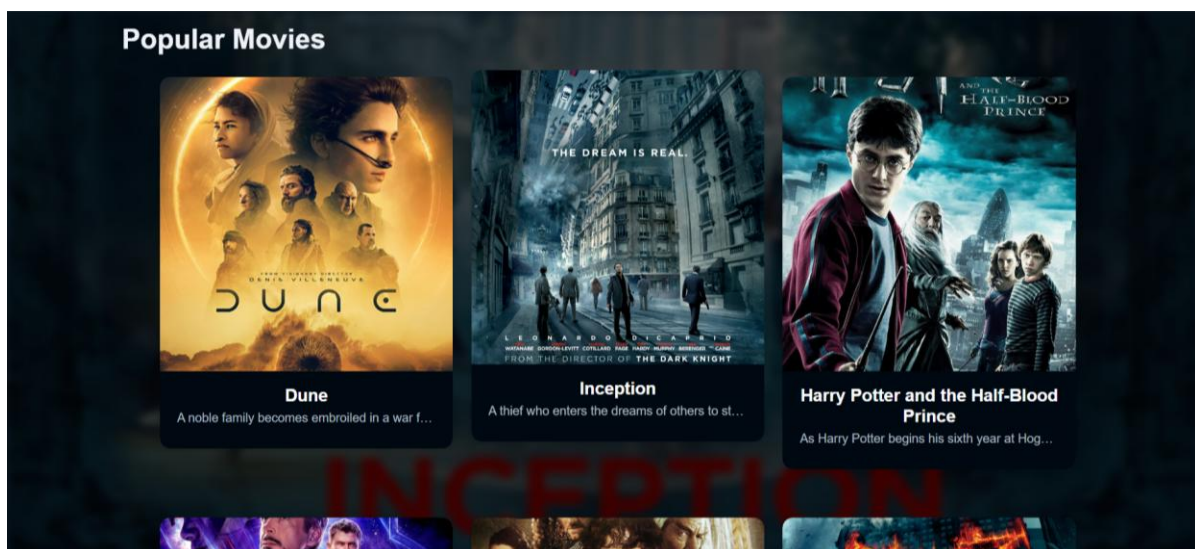
13. Sample Code – Code Snippets

1. Cinematic Dynamic Background – CSS, Javascript and jQuery

```
#dynamic-bg {  
  position: fixed;  
  top: 0;  
  left: 0;  
  width: 100%;  
  height: 100%;  
  background-size: cover;  
  background-position: center;  
  background-repeat: no-repeat;  
  filter: brightness(0.4) blur(4px);  
  transition: background-image 0.5s ease-in-out,  
              filter 0.5s ease-in-out;  
  z-index: -1;  
}
```

```
// Mouse Enter (Hover ON)  
$(document).on('mouseenter', '.movie-card', function() {  
  const bgUrl = $(this).data('bg-url');  
  if (bgUrl) {  
    $dynamicBg.css('background-image', `url('${bgUrl}')`);  
  }  
});  
  
// Mouse Leave (Hover OFF)  
$(document).on('mouseleave', '.movie-card', function() {  
  $dynamicBg.css('filter', 'brightness(0.3) blur(5px)');  
});
```

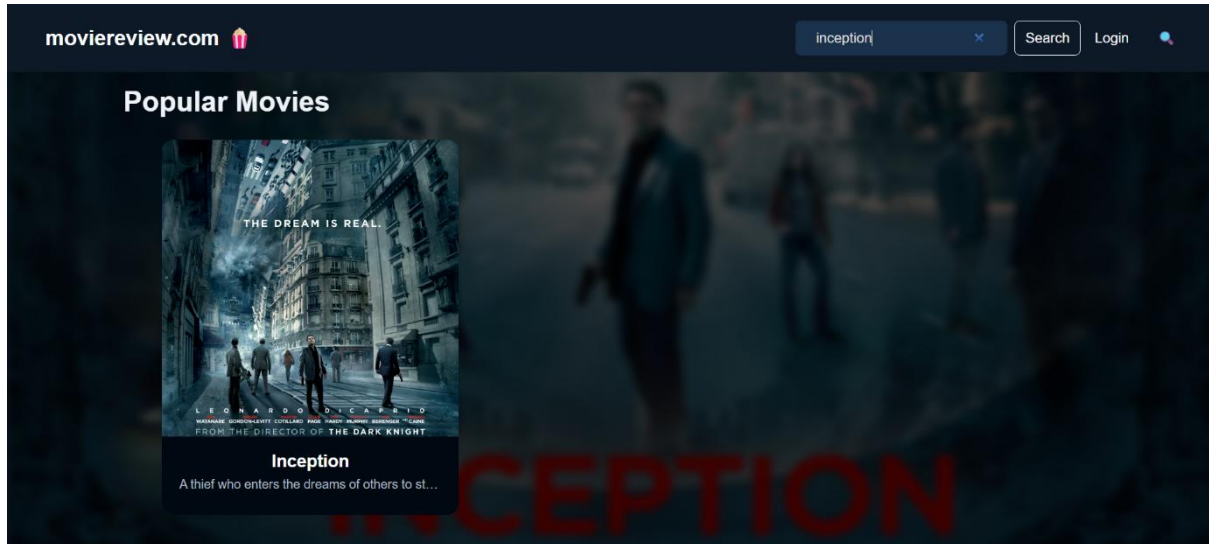
Output:



2. Instant Movie Search Filter – JavaScript and jQuery

```
$("#movie-search-nav").on("keyup", function() {  
  const searchText = $(this).val().toLowerCase();  
  const filteredMovies = movies.filter(movie =>  
    movie.title.toLowerCase().includes(searchText)  
  );  
  renderMovies(filteredMovies);  
});
```

Output:



3. Dynamic Hero Section Slideshow - JavaScript

```
// Slideshow variables  
let currentSlideIndex = 0;  
let slideshowInterval;  
const slideshowImages = movies.slice(0, 15); // Use first 15 movies for slideshow  
  
// Initialize slideshow  
function initializeSlideshow() {  
  const $slideshow = $('#hero-slideshow');  
  const $indicators = $('.slide-indicators');  
  
  // Clear existing content  
  $slideshow.empty();  
  $indicators.empty();  
  
  // Create slide images  
  slideshowImages.forEach((movie, index) => {  
    const slideElement = `  
      <div class="slide ${index === 0 ? 'active' : ''}" data-index="${index}">  
          
      </div>  
    `;  
    $slideshow.append(slideElement);  
  
    // Create indicators  
    const indicator = `    $indicators.append(indicator);  
  });  
};
```



```

// Start automatic slideshow
function startSlideshow() {
  slideshowInterval = setInterval(() => {
    nextSlide();
  }, 5000); // Change slide every 5 seconds
}

// Stop slideshow
function stopSlideshow() {
  clearInterval(slideshowInterval);
}

// Go to next slide
function nextSlide() {
  currentSlideIndex = (currentSlideIndex + 1) % slideshowImages.length;
  updateSlide();
}

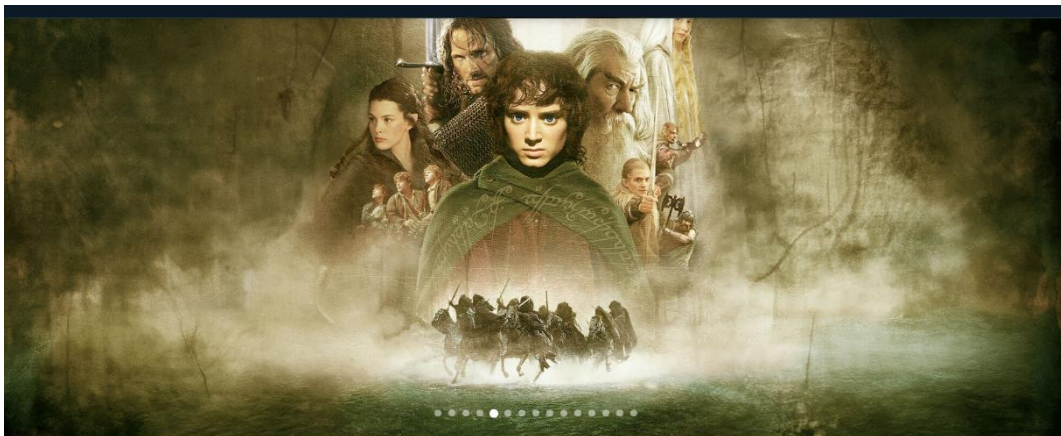
// Go to previous slide
function prevSlide() {
  currentSlideIndex = currentSlideIndex === 0 ? slideshowImages.length - 1 : currentSlideIndex - 1;
  updateSlide();
}

// Update active slide
function updateSlide() {
  $('.slide').removeClass('active');
  $('.indicator').removeClass('active');

  $(`.slide[data-index="${currentSlideIndex}"]`).addClass('active');
  $(`.indicator[data-slide="${currentSlideIndex}"]`).addClass('active');
}

```

Output:

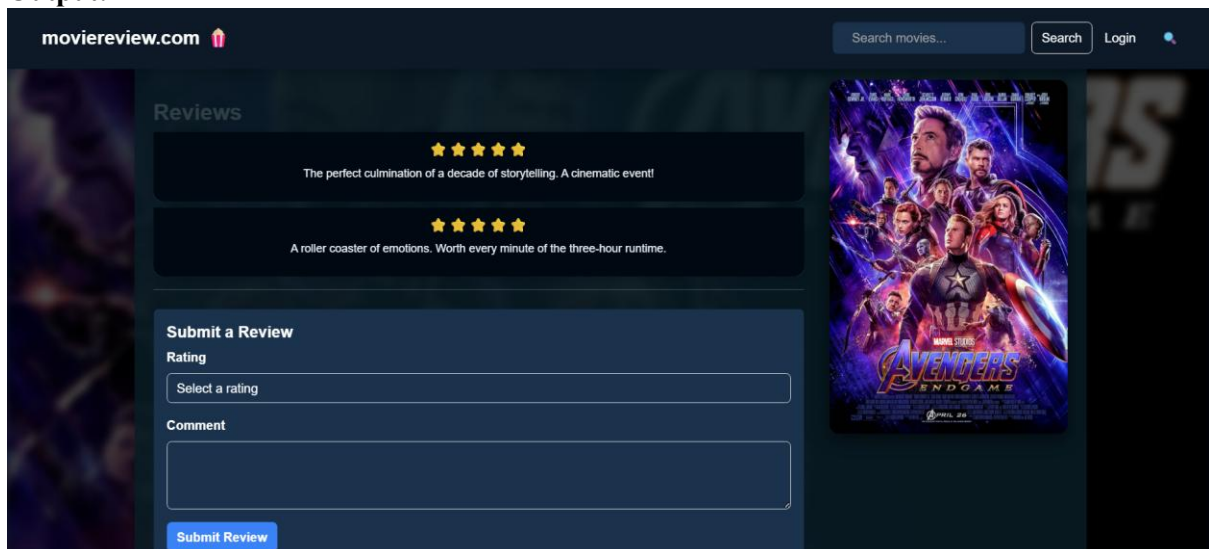


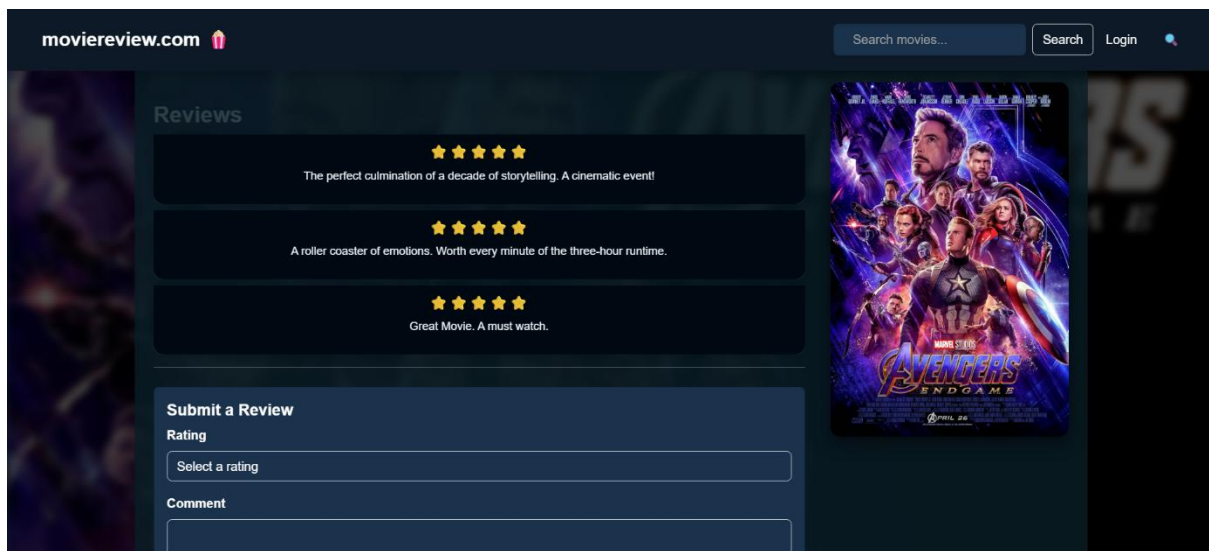
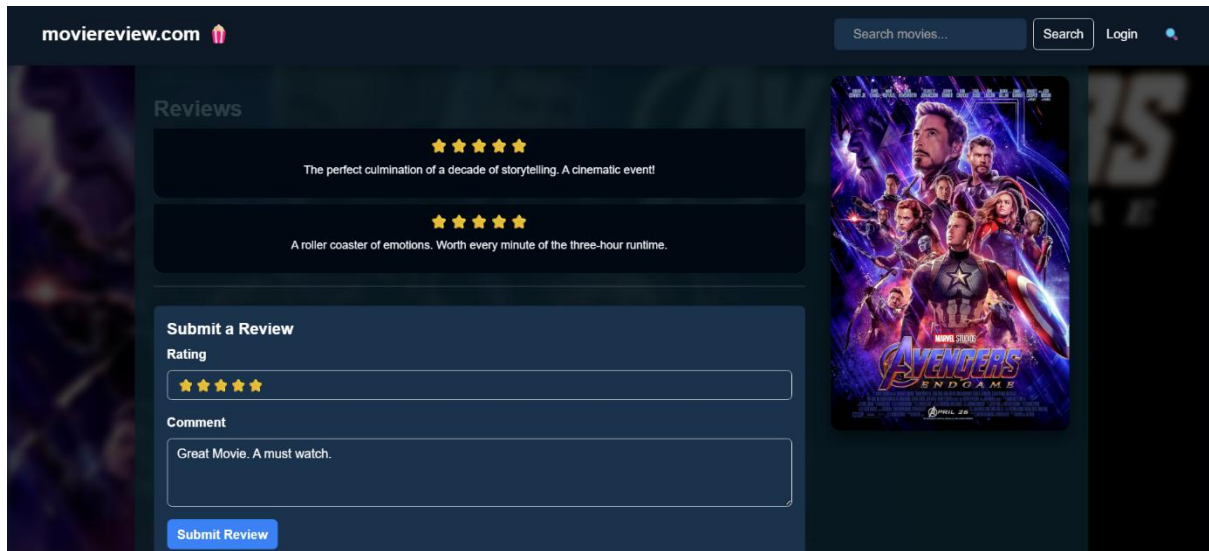
4. In-Memory Review Display and Submission – JavaScript and jQuery

```
function displayMovieReviews(movie) {
  $("#user-reviews").empty();
  if (movie.reviews.length === 0) {
    $("#user-reviews").append('<p style="color: #f0f4f8;">No reviews yet. Be the first to add one! 📄</p>');
  } else {
    movie.reviews.forEach(review => {
      const reviewElement = `
        <div class="card mb-2">
          <div class="card-body">
            <h6 class="card-subtitle mb-2 text-muted">${'★'.repeat(review.rating)}</h6>
            <p class="card-text">${review.comment}</p>
          </div>
        </div>
      `;
      $("#user-reviews").append(reviewElement);
    });
  }
}
```

```
$("#review-form").on("submit", function(e) {
  e.preventDefault();
  const movieId = $(this).data("movie-id");
  const movie = movies.find(m => m.id === movieId);
  if (movie) {
    const newReview = {
      rating: $("#review-rating").val(),
      comment: $("#review-comment").val()
    };
    movie.reviews.push(newReview);
    displayMovieReviews(movie);
    $("#review-form")[0].reset();
  }
});
```

Output:



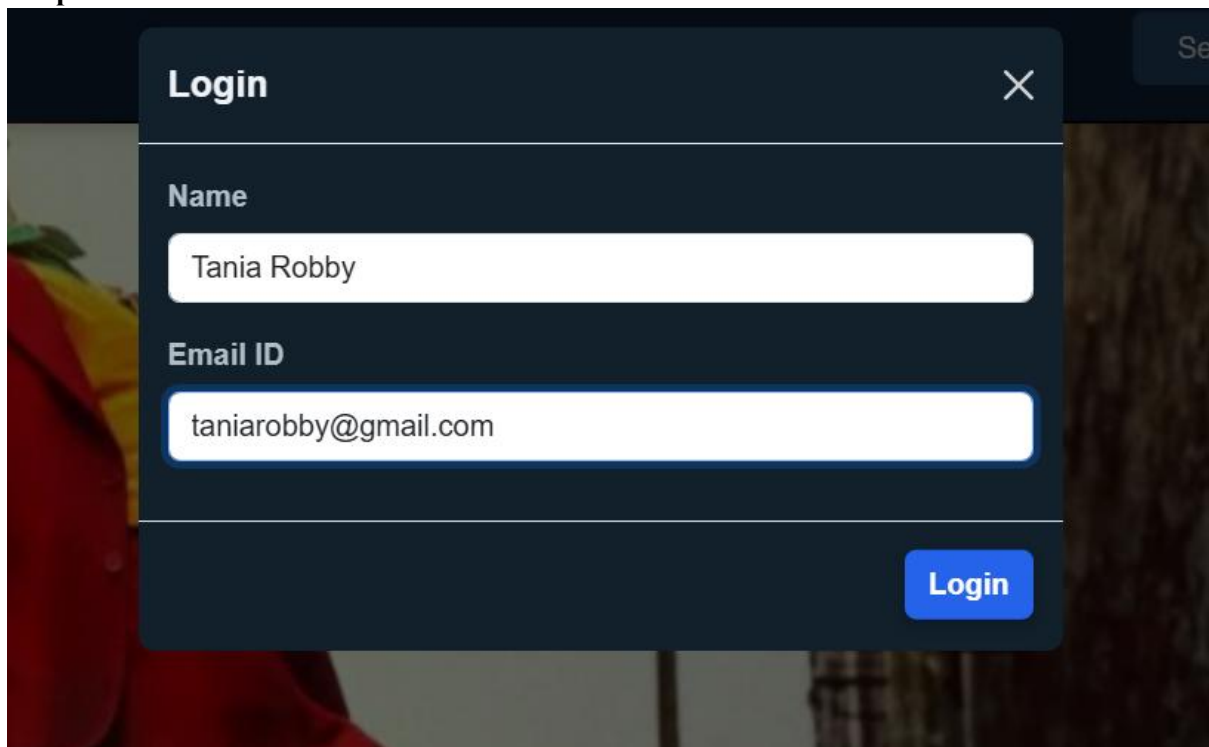


5. Responsive Login Page – HTML, Bootstrap and JavaScript

```
<!-- Login Modal -->
<div class="modal fade" id="loginModal" tabindex="-1" aria-labelledby="loginModalLabel" aria-hidden="true">
  <div class="modal-dialog">
    <div class="modal-content">
      <form id="login-form">
        <div class="modal-header">
          <h5 class="modal-title" id="loginModalLabel">Login</h5>
          <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
        </div>
        <div class="modal-body">
          <div class="mb-3">
            <label for="login-name" class="form-label">Name</label>
            <input type="text" class="form-control" id="login-name" required>
          </div>
          <div class="mb-3">
            <label for="login-email" class="form-label">Email ID</label>
            <input type="email" class="form-control" id="login-email" required>
          </div>
        </div>
        <div class="modal-footer">
          <button type="submit" class="btn btn-primary">Login</button>
        </div>
      </form>
    </div>
  </div>
</div>
```

```
// Handle login form submission
$('#login-form').on('submit', function(e) {
  e.preventDefault();
  const name = $('#login-name').val();
  const email = $('#login-email').val();
  // You can add your login logic here
  $('#loginModal').modal('hide');
  alert(`Welcome, ${name}!`);
});
});
```

Output:



A screenshot of a dark-themed 'Login' modal window. The modal has a title bar with 'Login' and a close button (X). It contains two input fields: 'Name' with the value 'Tania Robby' and 'Email ID' with the value 'taniarobby@gmail.com'. A blue 'Login' button is located at the bottom right of the modal.

