ONLINE MOVIE TICKET BOOKING USINF MEAN STACK

ABSTRACT:

The Movie Booking Management System is a full-stack web application designed to digitalize and simplify the process of browsing movies and booking tickets. It allows users to view available movie listings, check seat availability, and make reservations in real-time through a user-friendly interface. On the backend, the application is built using Node.js, Express.js, and MongoDB, ensuring efficient data handling and seamless communication between the client and server. The frontend, developed with HTML, CSS, and JavaScript, provides an interactive and responsive experience for users. Security is maintained using modern practices like JWT-based authentication and password encryption with Bcrypt. The system's modular architecture, comprising models, controllers, and routes, ensures scalability and maintainability. This project demonstrates practical implementation of CRUD operations, RESTful APIs, and full-stack web development principles, making it a valuable tool for learning and real-world use in the cinema and entertainment industry.

INTRODUCTION:

The Movie Booking Management System is a web-based application developed to streamline and digitize the movie ticket booking process. With the increasing demand for online services in the entertainment sector, this system aims to provide users with a convenient platform to browse movies, check seat availability, and book tickets from the comfort of their homes. It eliminates the need for manual ticketing, reduces waiting times, and enhances the overall user experience.

Technically, the project uses a full-stack architecture. The backend is developed using Node.js and Express.js, which handle routing, server logic, and database operations. MongoDB is used as the database to store movies, users, and booking information in a structured and scalable format. On the frontend, HTML, CSS, and JavaScript are used to create a responsive and interactive user interface, allowing users to interact with the system efficiently.

Security is a major consideration in this system. User credentials are protected using bcrypt hashing, and access to protected routes is managed using JWT (JSON Web Tokens) for authentication. These measures ensure that user data remains secure and that only authorized users can perform specific operations, particularly administrative tasks.

In summary, the Movie Booking Management System serves as a real-world example of how modern web technologies can be applied to automate and enhance everyday services. It combines efficient backend operations with an intuitive frontend to deliver a complete and functional solution for both moviegoers and cinema management teams. This project not only solves a practical problem but also showcases core concepts of web development, database management, and system security.

DESIGN:

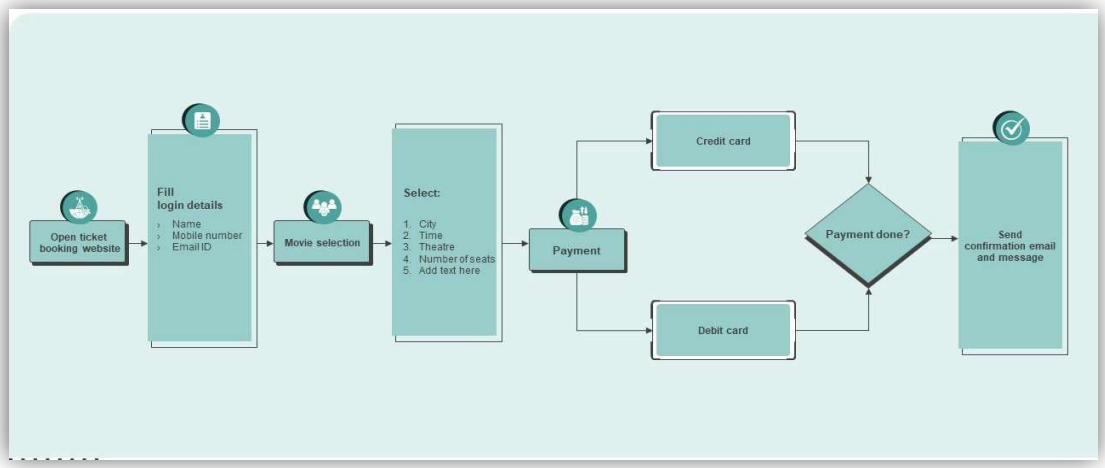
The design of the Movie Booking Management System follows the MVC (Model-View-Controller) architecture to ensure clear separation of concerns, scalability, and maintainability. This architecture breaks the system down into three core components: the **Model** (data and business logic), the **View** (frontend interface), and the **Controller** (request handling and communication between model and view).

1. Frontend Design (View)

The frontend is designed using HTML, CSS, and JavaScript to provide a responsive and user-friendly interface. Users can navigate through movie listings, view details, select seats, and book tickets. JavaScript dynamically interacts with the backend using fetch() API calls to retrieve movie data and update the UI in real time. The UI design focuses on simplicity, accessibility, and clarity, especially in the booking and movie browsing screens.

2. Backend Design (Controller & Server Logic)

The backend is developed using Node.js with the Express.js framework. It handles all client requests, routes them through defined endpoints, and returns responses. Controllers are responsible for implementing business logic such as user registration, login authentication, movie creation, and seat booking. Middleware is used to protect routes (e.g., admin-only access), and API routes are cleanly separated by function.



3. Database Design (Model)

The database is built with MongoDB, a NoSQL document-based database, and accessed through Mongoose for schema modeling. There are two primary collections:

- **Users**: Stores user credentials, roles (admin/user), and basic profile info.
- **Bookings**: Records which user booked which movie, at what time, and which seats.

4. Authentication & Security Design

Security is handled via JWT-based authentication, which securely identifies users across sessions. Passwords are hashed using bcrypt before being stored in the database. Protected routes and role-based access ensure that only admins can perform sensitive actions like adding or managing movies.

This modular and layered design ensures that each component is independently manageable and can be upgraded or scaled as needed. It provides a solid foundation for further enhancement, such as integrating payment gateways, adding movie posters, or expanding to a full MEAN-stack with Angular.

FLOW OF THE DATA IN THE SYSTEM:

The data flow in the Movie Booking Management System describes how information moves through the application — from user input on the frontend, to processing in the backend, and storage in the database. It follows a typical client-server architecture and adheres to RESTful API communication between the frontend and backend.

1. User Registration & Login

- Input: The user submits a registration or login form from the frontend.
- Process:
    - The frontend sends a POST request to the backend (/api/register or /api/login).
    - Backend receives the data, validates it, and hashes the password using bcrypt (for registration).
    - On successful login, the backend generates a JWT token.
- Output: User gets a response (e.g., success message or error), and on login, the JWT token is stored for future authenticated requests.

2. Movie Retrieval

- Input: When the homepage loads or the user navigates to the movie section, a request is made to fetch movies.
- Process:
    - Frontend sends a GET request to the backend (/api/movies).
    - Backend fetches movie data from the MongoDB database via the Mongoose model.
- Output: A list of movie objects (title, genre, showtime, etc.) is sent back and displayed on the frontend.

3. Booking a Movie

- Input: The user selects a movie, chooses seats, and clicks "Book."
- Process:
    - Frontend sends a POST request to /api/bookings with movie ID, user ID, and selected seat numbers.
    - Backend checks seat availability, reserves the selected seats, and stores the booking in the database.
- Output: Backend responds with a booking confirmation, and the booking data is stored in the bookings collection.

4. Viewing Booked Seats

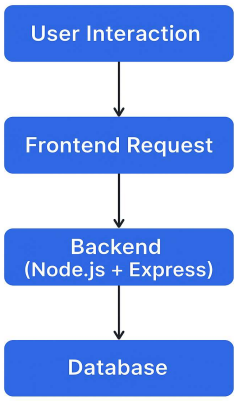
- Input: When a user views a movie’s seat layout.
- Process:
    - A GET request is made to /api/bookings/:movieId to fetch booked seats.

    - Backend retrieves booking data from the database for the selected movie.
- Output: Frontend displays booked and available seats to prevent double booking.

Summary of Data Flow

- Frontend ↔ Backend: Communicates via HTTP (REST APIs).
- Backend ↔ Database: Uses Mongoose to interact with MongoDB.
- User Authentication: JWT token flows with each secured request.
- Booking Flow: Real-time seat update ensures no duplication or clash.

FLOW CHART:



TECHNOLOGIES USED:

Here are the technologies used in your MEAN stack-based movie booking project

Frontend

- HTML5 – Structure of the website
- CSS3 – Styling and layout
- JavaScript – Dynamic interactions, API calls

Backend

- Node.js – Server-side JavaScript runtime
- Express.js – Web framework to build APIs and route requests

Database

- MongoDB – NoSQL database for storing movies, users, bookings
- Mongoose – ODM (Object Data Modeling) library for MongoDB

Authentication (if used)

- JWT (JSON Web Tokens) – Secure login/authentication
- bcrypt.js – Password hashing
- Development Tools
- VS Code – Code editor
- Git + GitHub – Version control
- Nodemon – Auto-reload server during development

ABOUT EACH MODULE:

- **app.js – Main Server Setup**
  - This is the entry point of your application.
  - Initializes Express server and connects to MongoDB.
  - Loads middleware like body parsers or CORS.
  - It would typically also define or import route handlers (e.g., for users, movies, bookings).
- **createAdmin.js – Admin Initialization**
  - Script to create an initial admin user in the database.
  - Useful when you deploy the system for the first time.
  - Ensures there's at least one admin account to manage movies and users.
- **.env – Environment Configuration**
  - Holds sensitive config values like MongoDB URI, server port, etc.
  - Keeps your credentials safe and allows environment-specific configuration (dev, prod).
- **models/User.js – User Model**
  - Defines the schema for storing user details.
  - Includes fields like:
    - name
    - email
    - password
    - role (e.g., admin or user)

- Used by authentication and authorization logic to distinguish between users and admins.

🔹 **models/Movie.js – Movie Model**

- Schema for movie entries shown to users.
- Fields include:
  - name: Name of the movie
  - language: Language (e.g., English, Hindi)
  - cast: List of main actors
  - type: Genre (e.g., action, comedy)
  - poster: Image URL or path
  - trailer: YouTube/video link
- Managed by the admin to create/update/delete movie data.

🔹 **assets/ – Static Resources**

- Contains images such as logos or banners (e.g., logo.jpg, hello.jpg).
- Used in the frontend or dashboard for UI enhancement.

🔹 **node _modules/ – Installed Dependencies**

- Includes all external libraries (like express, mongoose, dotenv, etc.)
- Automatically generated when running npm install.

CODE:

Home.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Movie Listing</title>
  <style>
    body {
      font-family: Arial, sans-serif;
```

```
    background-color: black;
    margin: 0;
    padding: 0;
    font-size: 20px;
}
header {
    display: flex;
    justify-content: space-between;
    align-items: center;
    background: blueviolet;
    padding: 20px 20px;
    color: white;
}
.logo img {
    height: 120px;
    width: 120px;
    border-radius: 50%;
}
nav ul {
    list-style: none;
    display: flex;
    gap: 15px;
    padding: 0;
}
nav ul li a {
    color: white;
    text-decoration: none;
    font-size: 20px;
}
#logoutBtn {
    background-color: red;
    color: white;
```

```
    padding: 10px 10px;
    border: none;
    border-radius: 5px;
    cursor: pointer;
    font-size: 20px;
}
#logoutBtn:hover {
    background-color: darkred;
}
.movie-list-container {
    text-align: left;
    padding: 20px;
    background-color: white;
    margin: 20px;
    border-radius: 10px;
    box-shadow: 0px 4px 8px rgba(0, 0, 0, 0.1);
}
.movies-row {
    display: flex;
    overflow-x: auto;
    padding: 10px;
    gap: 20px;
    scrollbar-width: none;
}
.movies-row::-webkit-scrollbar {
    display: none;
}
/* Main container centers everything */
.main-container {
    display: flex;
    flex-direction: column;
    align-items: center;  /* Centers content horizontally */
```

```
    padding: 20px;
}
/* Centered Filter Section */
.filters {
    background: blueviolet;
    padding: 15px;
    width: 400px;
    border-radius: 20px;
    box-shadow: 0px 8px 16px rgba(0, 0, 0, 0.1);
    text-align: center;
    border: 5px solid transparent; /* Default transparent border */
    transition: border-color 0.3s ease, box-shadow 0.3s ease;
}
.filters:hover {
    border-color: blueviolet; /* Dark purple border on hover */
    box-shadow: 0px 4px 12px rgba(123, 31, 162, 0.4); /* Slightly stronger shadow */
}
.filters label {
    font-size: 20px;
    font-weight: bold;
    display: block;
    margin-top: 8px;
}
.filters select {
    width: 90%;
    padding: 6px;
    border-radius: 5px;
    border: 1px solid #ccc;
    font-size: 20px;
}
.filters select:hover {
    border-color: blueviolet; /* Dark purple border on hover */
```

```
    box-shadow: 0px 8px 16px rgba(123, 31, 162, 0.4); /* Slightly stronger shadow */
}
/* Movies Grid */
.moviesContainer {
    display: flex;
    flex-wrap: wrap;
    gap: 15px;
    width: 80%;
    justify-content: center;
}
.movie-card {
    background: #fff;
    padding: 10px;
    border-radius: 10px;
    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
    transition: transform 0.2s;
    width: 250px;
    text-align: center;
}
.movie-card:hover {
    background-color: #e0c3fc; /* Light purple */
    transform: scale(1.05);
    transition: background-color 0.3s ease, transform 0.3s ease;
}
.movie-card img {
    width: 100%;
    height: 350px;
    object-fit: cover;
    border-radius: 10px;
}
.rating {
    background: blueviolet;
```

```
        color: white;
        font-size: 14px;
        padding: 5px;
        border-radius: 5px;
        display: inline-block;
        margin-top: 5px;
    }
    #movieList {
        display: flex;
        flex-wrap: wrap;
        gap: 20px;
        justify-content: center;
    }
    .movie-card {
        width: 250px;
        border: 1px solid #ddd;
        border-radius: 8px;
        padding: 10px;
        box-shadow: 2px 2px 5px rgba(0, 0, 0, 0.1);
        text-align: center;
    }
    .movie-card img {
        width: 100%;
        border-radius: 8px;
    }
    .movie-card h3 {
        margin: 10px 0 5px;
    }
    .movie-card p {
        font-size: 14px;
    }
</style>
```

```
</head>
<body>
  <header>
    <div class="logo">
      <img src="assets/logo2.jpg" alt="MovieZone Logo">
    </div>
    <nav>
      <ul>
        <li><a href="home1.html">Home</a></li>
        <li><a href="login.html">Login</a></li>
        <li><a href="signup.html">Signup</a></li>
        <li><button id="logoutBtn">Logout</button></li>
      </ul>
    </nav>
  </header><br>
  <div class="main-container">
    <div class="filters">
      <h2>Filter Movies</h2>
      <label for="language">Language:</label>
      <select id="language" onchange="filterMovies()">
        <option value="all">All</option>
        <option value="Telugu">Telugu</option>
        <option value="English">English</option>
        <option value="Hindi">Hindi</option>
      </select>
      <label for="genre">Genre:</label>
      <select id="genre" onchange="filterMovies()">
        <option value="all">All</option>
        <option value="Action">Action</option>
        <option value="Drama">Drama</option>
        <option value="Romantic">Romantic</option>
        <option value="Thriller">Thriller</option>
```

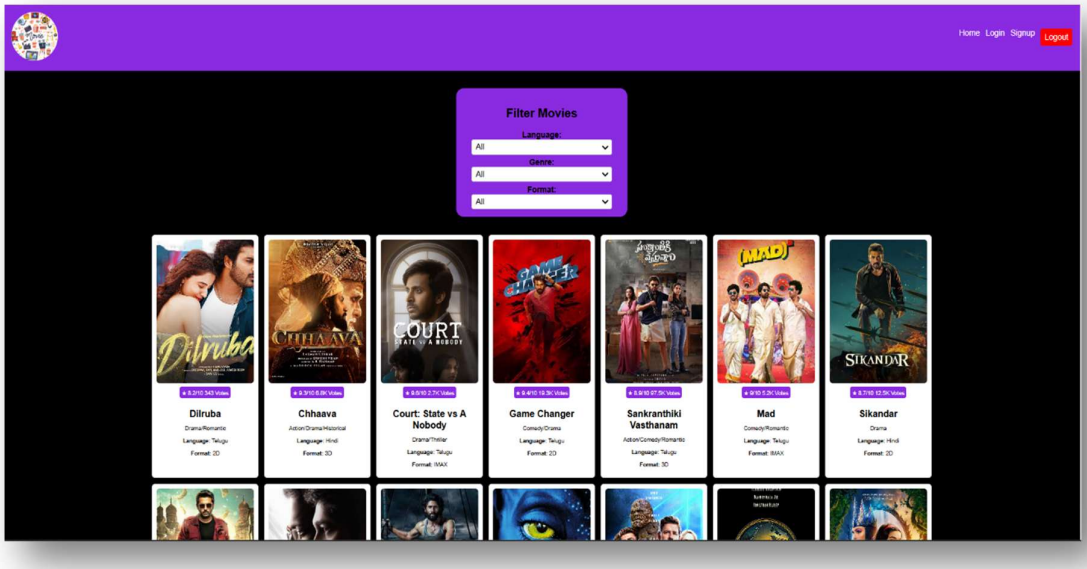
```
        </select>
        <label for="format">Format:</label>
        <select id="format" onchange="filterMovies()">
          <option value="all">All</option>
          <option value="2D">2D</option>
          <option value="3D">3D</option>
          <option value="IMAX">IMAX</option>
        </select>
      </div><br><br>
      <div id="moviesContainer" class="moviesContainer"></div>
      <h2>Movie List</h2>
      <div id="movieList"></div>
    </div>
    <script>
      document.getElementById("logoutBtn").addEventListener("click", function () {
        localStorage.removeItem("username"); // Remove user session
        localStorage.removeItem("authToken"); // Remove authentication token if used
        window.location.href = "login.html"; // Redirect to login page
      });
      const movies = [
            { title: "Dilruba", image: "assets/dilruba.jpg", rating: 8.2, votes: "343", genre: "Drama/Romantic", language: "Telugu", format: "2D", link: "dilruba.html" },
          { title: "Chhaava ", image: "assets/chhaava.jpg", rating: 9.3, votes: "6.8K", genre: "Action/Drama/Historical", language: "Hindi", format: "3D", link: "chhaava.html" },
        { title: "Court: State vs A Nobody", image: "assets/court.jpg", rating: 9.6, votes: "2.7K", genre: "Drama/Thriller", language: "Telugu", format: "IMAX", link: "first.html" },
        { title: "Game Changer", image: "assets/gamechanger.jpg", rating: 9.4, votes: "19.3K", genre: "Comedy/Drama", language: "Telugu", format: "2D", link: "game_changer.html" },
        { title: "Sankranthiki Vasthanam", image: "assets/sankranthiki.jpg", rating: 8.9, votes: "97.5K", genre: "Action/Comedy/Romantic", language: "Telugu", format: "3D", link: "sankranthi.html" },
            { title: "Mad", image: "assets/mad.jpg", rating: 9.0, votes: "5.2K", genre: "Comedy/Romantic", language: "Telugu", format: "IMAX", link: "mad.html" },
          { title: "Sikandar", image: "assets/sikandra.jpg", rating: 8.7, votes: "12.5K", genre: "Drama", language: "Hindi", format: "2D", link: "Sikandar.html" },
```

```
        { title: "Robinhood", image: "assets/robinhood.jpg", rating: 8.5, votes: "4.3K", genre: "Sci-Fi/Thriller", language: "Telugu", format: "3D", link: "robinhood.html" },
        { title: "Ramam Raghavam", image: "assets/ramam.jpg", rating: 9.1, votes: "8.6K", genre: "Adventure/Action", language: "Telugu", format: "IMAX", link: "#" },
        { title: "Thandel", image: "assets/thandel.jpg", rating: 8.8, votes: "6.9K", genre: "Drama", language: "Telugu", format: "2D", link: "thandel.html" },
        { title: "avatar", image: "assets/avatar.jpg", rating: 9.8, votes: "12.9K", genre: "Action", language: "English", format: "2D", link: "#" },
        { title: "fantastic", image: "assets/fantastic.jpg", rating: 8.8, votes: "6.9K", genre: "Thriller", language: "English", format: "2D", link: "#" },
        { title: "jurassic", image: "assets/jurassic.jpg", rating: 8.8, votes: "6.9K", genre: "Action", language: "English", format: "2D", link: "#" },
        { title: "snowwhite", image: "assets/snowwhite.jpg", rating: 8.8, votes: "6.9K", genre: "Drama", language: "English", format: "2D", link: "#" },
        { title: "thunder", image: "assets/thunder.jpg", rating: 8.8, votes: "6.9K", genre: "Thriller", language: "English", format: "2D", link: "#" }
    ];
    const moviesContainer = document.getElementById("moviesContainer");
    function displayMovies(filteredMovies) {
        moviesContainer.innerHTML = "";
        filteredMovies.forEach(movie => {
            const movieCard = document.createElement("div");
            movieCard.classList.add("movie-card");
            movieCard.innerHTML = `
                <img src="${movie.image}" alt="${movie.title}">
                <p class="rating">&#9733; ${movie.rating}/10 ${movie.votes} Votes</p>
                <h3>${movie.title}</h3>
                <p>${movie.genre}</p>
                <p><strong>Language:</strong> ${movie.language}</p>
                <p><strong>Format:</strong> ${movie.format}</p>
            `;
            if (movie.link) {
                movieCard.addEventListener("click", () => {
                    window.location.href = movie.link;
                });
```

```
        }
        moviesContainer.appendChild(movieCard);
    });
}
function filterMovies() {
    const selectedLanguage = document.getElementById("language").value;
    const selectedGenre = document.getElementById("genre").value;
    const selectedFormat = document.getElementById("format").value;
    const filteredMovies = movies.filter(movie =>
        (selectedLanguage === "all" || movie.language === selectedLanguage) &&
        (selectedGenre === "all" || movie.genre.includes(selectedGenre)) &&
        (selectedFormat === "all" || movie.format === selectedFormat)
    );
    displayMovies(filteredMovies);
}
// Initial display of all movies
displayMovies(movies);
async function fetchMovies() {
    const res = await fetch('http://localhost:5000/movies');
    const movies = await res.json();
    const movieList = document.getElementById('movieList');
    movieList.innerHTML = ''; // Clear previous content
    movies.forEach(movie => {
        const div = document.createElement('div');
        div.classList.add('movie-card');
        div.innerHTML = `
            <img src="${movie.image}" alt="${movie.title}">
            <h3>${movie.title}</h3>
            <p><strong>Year:</strong> ${movie.releaseYear}</p>
            <p><strong>Rating:</strong> ${movie.rating}/10</p>
            <p>${movie.description}</p>
        `;
```

```
            movieList.appendChild(div);
        });
    }
    fetchMovies();
    async function fetchMovies() {
  const res = await fetch("http://localhost:5000/movies");
  const movies = await res.json();
  const moviesContainer = document.getElementById("moviesContainer");
  moviesContainer.innerHTML = "";
  movies.forEach(movie => {
    const div = document.createElement("div");
    div.classList.add("movie-card");
    div.innerHTML = `
      <img src="${movie.image}" alt="${movie.title}">
      <p class="rating">&#9733; ${movie.rating}/10 ${movie.votes} Votes</p>
      <h3>${movie.title}</h3>
      <p>${movie.genre}</p>
      <p><strong>Language:</strong> ${movie.language}</p>
      <p><strong>Format:</strong> ${movie.format}</p>
    `;
    moviesContainer.appendChild(div);
  });
}
fetchMovies(); // Load movies on page load
  </script>
</body>
</html>
```

Output:



LOGIN.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Login</title>
    <style>
        * {
            margin: 0;
            padding: 0;
            box-sizing: border-box;
            font-family: Arial, sans-serif;
        }
        body {
            display: flex;
            justify-content: center;
            align-items: center;
            height: 100vh;
            background: blueviolet;
```

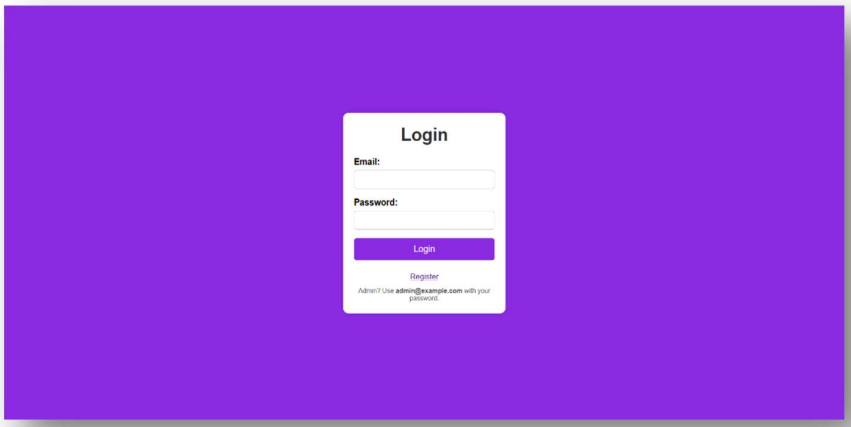
```
}
.login-container {
    background: white;
    padding: 20px;
    border-radius: 10px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
    width: 300px;
    text-align: center;
}
h1 {
    margin-bottom: 20px;
    color: #333;
}
.input-group {
    margin-bottom: 15px;
    text-align: left;
}
label {
    display: block;
    font-weight: bold;
    margin-bottom: 5px;
}
input {
    width: 100%;
    padding: 8px;
    border: 1px solid #ccc;
    border-radius: 5px;
}
button {
    width: 100%;
    padding: 10px;
    background: blueviolet;
```

```
            color: white;
            border: none;
            border-radius: 5px;
            cursor: pointer;
            font-size: 16px;
        }
        button:hover {
            background: #218838;
        }
        .register-link {
            margin-top: 10px;
            font-size: 14px;
        }
        p {
            font-size: 12px;
            color: #555;
            margin-top: 10px;
        }
    </style>
</head>
<body>
    <div class="login-container">
        <h1>Login</h1>
        <form id="loginForm">
            <div class="input-group">
                <label for="email">Email:</label>
                <input type="email" id="email" name="email" required>
            </div>
            <div class="input-group">
                <label for="password">Password:</label>
                <input type="password" id="password" name="password" required>
            </div>
```

```
        <button type="submit">Login</button>
    </form>
    <br>
    <a class="register-link" href="register.html">Register</a>
    <p>Admin? Use <strong>admin@example.com</strong> with your password.</p>
</div>
<script>
    document.getElementById("loginForm").addEventListener("submit", async function (event) {
    event.preventDefault(); // Prevent default form submission
    let email = document.getElementById("email").value.trim();
    let password = document.getElementById("password").value.trim();
    if (!email || !password) {
        alert("❌ Please enter both email and password!");
        return;
    }
    try {
        let response = await fetch("http://localhost:3000/login", {  // Adjust URL if needed
            method: "POST",
            headers: {
                "Content-Type": "application/json"
            },
            body: JSON.stringify({ email, password })  // Send email & password
        });
        let result = await response.json();
        if (response.ok) {
            alert("✅ Login Successful!");
            localStorage.setItem("token", result.token);  // Store token for session handling
            window.location.href = "home1.html";  // Redirect after login
        } else {
            alert("❌ " + (result.error || "Invalid credentials!"));
        }
```

```
    } catch (error) {
        console.error("❌ Error during login:", error);
        alert("❌ Server error! Try again later.");
    }
});
    </script>
</body>
</html>
```

Output:



REGISTER.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Signup Page</title>
    <style>
        * {
            margin: 0;
            padding: 0;
```

```
    box-sizing: border-box;
    font-family: Arial, sans-serif;
}

body {
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
    background: blueviolet;
}
.signup-container {
    background: white;
    padding: 20px;
    border-radius: 10px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
    width: 300px;
    text-align: center;
}
h2 {
    margin-bottom: 20px;
}
.input-group {
    margin-bottom: 15px;
    text-align: left;
}
label {
    display: block;
    margin-bottom: 5px;
    font-weight: bold;
}
input {
```

```
            width: 100%;
            padding: 8px;
            border: 1px solid #ccc;
            border-radius: 5px;
        }
        button {
            width: 100%;
            padding: 10px;
            background: blueviolet;
            color: white;
            border: none;
            border-radius: 5px;
            cursor: pointer;
            font-size: 16px;
        }
        button:hover {
            background: #218838;
        }
        .error {
            color: red;
            font-size: 12px;
            display: block;
            height: 15px;
        }
        .login-link {
            margin-top: 10px;
            font-size: 14px;
        }
    </style>
</head>
<body>
    <div class="signup-container">
```

```
    <h2>Sign Up</h2>
    <form id="signupForm">
      <div class="input-group">
        <label for="name">Full Name</label>
        <input type="text" id="name" placeholder="Enter your name" required>
        <small class="error" id="nameError"></small>
      </div>
      <div class="input-group">
        <label for="email">Email</label>
        <input type="email" id="email" placeholder="Enter your email" required>
        <small class="error" id="emailError"></small>
      </div>
      <div class="input-group">
        <label for="password">Password</label>
        <input type="password" id="password" placeholder="Enter password" required>
        <small class="error" id="passwordError"></small>
      </div>
      <div class="input-group">
        <label for="confirmPassword">Confirm Password</label>
          <input type="password" id="confirmPassword" placeholder="Confirm password"
required>
        <small class="error" id="confirmPasswordError"></small>
      </div>
      <a href="login.html" style="text-decoration: none;">
        <button type="submit">Sign Up</button>
      </a>
      <p class="login-link">Already have an account? <a href="login.html">Login</a></p>
    </form>
  </div>
  <script>
          document.getElementById("signupForm").addEventListener("submit",    async
function(event) {
      event.preventDefault(); // Prevent default form submission
```

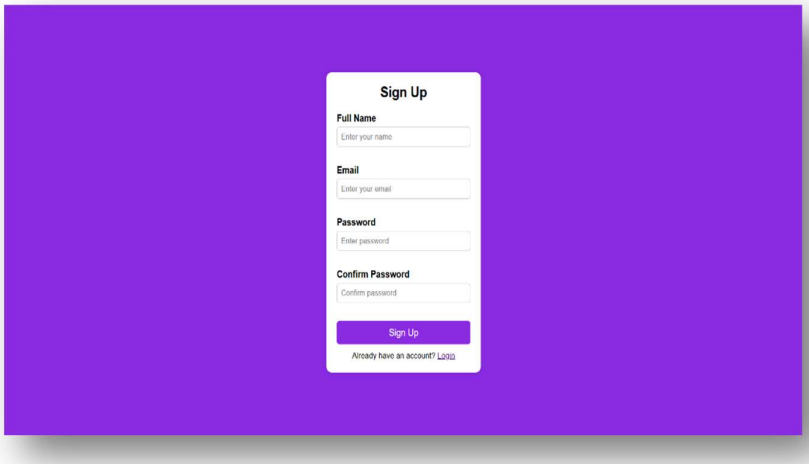
```
let name = document.getElementById("name").value.trim();
let email = document.getElementById("email").value.trim();
let password = document.getElementById("password").value.trim();
let confirmPassword = document.getElementById("confirmPassword").value.trim();
let nameError = document.getElementById("nameError");
let emailError = document.getElementById("emailError");
let passwordError = document.getElementById("passwordError");
let confirmPasswordError = document.getElementById("confirmPasswordError");
// Reset error messages
nameError.textContent = "";
emailError.textContent = "";
passwordError.textContent = "";
confirmPasswordError.textContent = "";
let valid = true;
// Name validation
if (name.length < 3) {
    nameError.textContent = "Name must be at least 3 characters.";
    valid = false;
}
// Email validation using regex
let emailRegex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;
if (!emailRegex.test(email)) {
    emailError.textContent = "Enter a valid email.";
    valid = false;
}
// Password validation
if (password.length < 6) {
    passwordError.textContent = "Password must be at least 6 characters.";
    valid = false;
}
// Confirm Password validation
if (password !== confirmPassword) {
```

```
            confirmPasswordError.textContent = "Passwords do not match.";
            valid = false;
        }
        if (valid) {
            try {
                let response = await fetch("http://localhost:3000/register", {
                    method: "POST",
                    headers: {
                        "Content-Type": "application/json"
                    },
                    body: JSON.stringify({ name, email, password })
                });
                let result;
                try {
                    result = await response.json();
                } catch (jsonError) {
                    throw new Error("Invalid JSON response from server.");
                }
                if (response.ok) {
                    alert("✅ Registration Successful! Redirecting to Login...");
                    window.location.href = "login.html"; // Redirect to login page
                } else {
                    alert("❌ Error: " + (result.error || "Unknown error"));
                }
            } catch (error) {
                console.error("❌ Error:", error);
                alert("❌ Something went wrong! Check console for details.");
            }
        }
    });
</script>
```

</body>

</html>

Output



FIRST.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Movie Page</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            background-color: #121212;
            color: #fff;
            margin: 0;
            padding: 0;
            text-align: center;
            font-size: 20px;
        }
        header {
            display: flex;
```

```
    justify-content: space-between;
    align-items: center;
    background: blueviolet;
    padding: 40px 20px;
    color: white;
}
.logo img {
    height: 50px;
}
nav ul {
    list-style: none;
    display: flex;
    gap: 15px;
    padding: 0;
}
nav ul li a {
    color: white;
    text-decoration: none;
    font-size: 20px;
}
.container {
    display: flex;
    margin: 20px;
    background: #1e1e1e;
    padding: 20px;
    border-radius: 10px;
}
.movie-poster img {
    width: 400px;
    border-radius: 10px;
    margin-left: 100px;
}
```

```
.movie-poster img:hover {
    transform: scale(1.2);
}
.movie-details {
    margin-left: 300px;
    text-align: left;
    font-size: 20px;
}
.rating, .movie-info {
    margin: 10px 0;
    font-size: 20px;
}
.tag {
    background: blueviolet;
    padding: 5px 10px;
    border-radius: 5px;
    margin-right: 5px;
}
.book-btn, .rate-btn, .submit-review-btn {
    background: blueviolet;
    color: #fff;
    border: none;
    padding: 10px 15px;
    cursor: pointer;
    border-radius: 5px;
    font-size: 18px;
    margin-top: 10px;
    transition: background 0.3s ease;
}
.book-btn:hover, .rate-btn:hover, .submit-review-btn:hover {
    background: darkviolet;
}
```

```
.section {
    width: 80%;
    margin: 20px auto;
    padding: 20px;
    color: white;
    background: rgb(194, 134, 250);
    border-radius: 10px;
    text-align: left;
    font-size: 20px;
}
.cast, .crew {
    display: flex;
    overflow-x: auto;
    gap: 40px;
    padding: 10px 0;
    font-size: 20px;
    transition: background 0.3s ease-in-out, transform 0.3s ease;
}
/* Hover Effect */
.section:hover,.cast:hover, .crew:hover {
    background: darkviolet; /* Light glow effect */
    transform: scale(1.02); /* Slight zoom effect */
    border-radius: 10px;
}
.person {
    text-align: center;
    min-width: 120px;
    font-size: 20px;
}
.person img {
    width: 200px;
    height: 200px;
```

```
    border-radius: 50%;
}
.book-btn {
    position: absolute;
    right: 20px;
    top: 20px;
    font-size: 20px;
}
.review-section {
    width: 80%;
    margin: 20px auto;
    padding: 20px;
    background: #f9f9f9;
    color: #000;
    border-radius: 10px;
}
.review {
    background: #f1f1f1;
    padding: 15px;
    border-radius: 30px;
    margin: 30px 0;
    display: flex;
    flex-direction: column;
    border-color: blueviolet;
}
.reviewText:hover{
    transform: scale(1.05);
    border-color: darkviolet;
}
#reviewText {
    width: 95%;
    padding: 10px;
```

```
            font-size: 20px;
            border: 5px solid blueviolet;
            border-radius: 5px;
            margin-top: 10px;
            resize: vertical;
        }
        .trailer-btn {
            background: blueviolet;
            color: white;
            border: none;
            padding: 12px 20px;
            font-size: 18px;
            font-weight: bold;
            cursor: pointer;
            border-radius: 5px;
            transition: background 0.3s ease, transform 0.2s ease;
            display: flex;
            align-items: center;
            gap: 8px;
        }
        .trailer-btn:hover {
            background: darkviolet;
            transform: scale(1.05);
        }
    </style>
</head>
<body>
    <header>
        <div class="logo">
            <img src="assets/logo.png" alt="MovieZone Logo">
        </div>
        <nav>
```

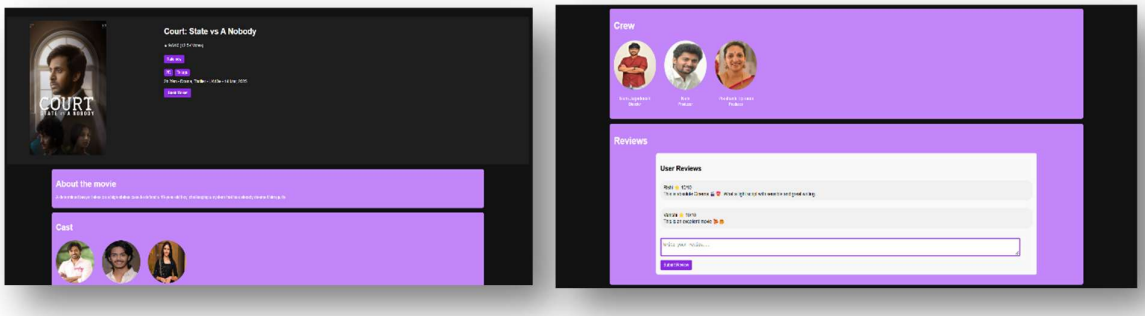
```
        <ul>
            <li><a href="home1.html">Home</a></li>
            <li><a href="login.html">Login</a></li>
            <li><a href="signup.html">Signup</a></li>
        </ul>
    </nav>
</header><br>
<div class="container">
    <div class="movie-poster">
        <img src="assets/court.jpg" alt="Court: State vs A Nobody"><br><br>
    </div>
    <div class="movie-details">
        <h1>Court: State vs A Nobody</h1>
        <div class="rating">
            <span class="star">★</span> <span>9.6/10 (12.5K Votes)</span><br><br>
            <button class="rate-btn">Rate now</button><br>
        </div>
        <div class="movie-info">
            <br>
            <span class="tag">2D</span>
            <span class="tag">Telugu</span><br>
            <p>2h 29m • Drama, Thriller • UA13+ • 14 Mar, 2025</p>
            <a href="second.html">
                <button class="trailer-btn">Book Ticket</button>
            </a>
        </div>
    </div>
</div>
<div class="section">
    <h1>About the movie</h1>
        <p>A determined lawyer takes on a high-stakes case to defend a 19-year-old boy, challenging a system that has already deemed him guilty.</p>
```

```
</div>
<div class="section">
  <h1>Cast</h1>
  <div class="cast">
    <div class="person">
      <img src="assets/priyadarshi.jpg" alt="Priyadarshi Pulikonda">
      <p>Priyadarshi Pulikonda<br><small>Actor</small></p>
    </div>
    <div class="person">
      <img src="assets/harsh.jpg" alt="Harsh Roshan">
      <p>Harsh Roshan<br><small>Actor</small></p>
    </div>
    <div class="person">
      <img src="assets/sridevi.jpg" alt="Sridevi Apalla">
      <p>Sridevi Apalla<br><small>Actor</small></p>
    </div>
  </div>
</div>
<div class="section">
  <h1>Crew</h1>
  <div class="crew">
    <div class="person">
      <img src="assets/ram.jpg" alt="Ram Jagadeesh">
      <p>Ram Jagadeesh<br><small>Director</small></p>
    </div>
    <div class="person">
      <img src="assets/nani.jpg" alt="Nani">
      <p>Nani<br><small>Producer</small></p>
    </div>
    <div class="person">
      <img src="assets/prashanti.jpg" alt="Prashanti Tipirneni">
      <p>Prashanti Tipirneni<br><small>Producer</small></p>
```

```
        </div>
    </div>
</div>
<div class="section">
    <h1>Reviews</h1>
    <div class="review-section">
        <h2>User Reviews</h2>
        <div class="review">
            <div class="user">Rishi <span class="rating">⭐ 10/10</span></div>
            <div class="content">This is absolute Cinema 🎬😍. What a tight script with sensible and great writing...</div>
        </div>
        <div class="review">
            <div class="user">Vamshi <span class="rating">⭐ 10/10</span></div>
            <div class="content">This is an excellent movie 🥰🤗 </div>
        </div>
        <textarea id="reviewText" placeholder="Write your review..."></textarea><br>
        <button class="submit-review-btn" onclick="addReview()">Submit Review</button>
    </div>
</div>
<script>
    function addReview() {
        let reviewText = document.getElementById("reviewText").value;
        if (reviewText.trim() === "") {
            alert("Please enter a review!");
            return;
        }
        let reviewSection = document.querySelector(".review-section");
        let newReview = document.createElement("div");
        newReview.classList.add("review");
        newReview.innerHTML = `
            <div class="user">Person <span class="rating">⭐ 9/10</span></div>
```

```
            <div class="content">${reviewText}</div>
        `;
        reviewSection.appendChild(newReview);
        document.getElementById("reviewText").value = "";
    }
    document.querySelector(".rate-btn").addEventListener("click", function() {
        alert("Rating feature coming soon!");
    });
    document.querySelectorAll(".book-btn").forEach(button => {
        button.addEventListener("click", function() {
            alert("Redirecting to ticket booking...");
        });
    });
</script>
</body>
</html>
```

Output



Second.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Movie Ticket Booking</title>
    <style>
```

```
body {
    font-family: Arial, sans-serif;
    background-color: white;
    margin: 0;
    padding: 0;
}
/* Header */
header {
    display: flex;
    justify-content: space-between;
    align-items: center;
    background: blueviolet;
    padding: 40px 20px;
    color: white;
}
.logo img {
    height: 50px;
}
nav ul {
    list-style: none;
    display: flex;
    gap: 15px;
    padding: 0;
}
nav ul li a {
    color: white;
    text-decoration: none;
    font-size: 20px;
}
/* Main Container */
.container {
    width: 90%;
```

```
    margin: 20px auto;
    background: rgb(221, 159, 247);
    padding: 20px;
    border-radius: 10px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
    position: relative;
}
/* Movie Info */
.movie-title {
    font-size: 28px;
    font-weight: bold;
}
.tags span {
    background: white;
    padding: 5px 10px;
    border-radius: 5px;
    margin-right: 5px;
}
/* Date Selection */
.date-picker {
    display: flex;
    gap: 10px;
    overflow-x: auto;
    padding: 10px 0;
}
.date {
    text-align: center;
    padding: 10px;
    background: #f2f2f2;
    border-radius: 5px;
    cursor: pointer;
    width: 70px;
```

```
    font-weight: bold;
}
.date:hover, .date.selected {
    background: darkviolet;
    color: white;
}
/* Showtimes */
.showtimes {
    display: flex;
    gap: 15px;
    flex-wrap: wrap;
}
.showtime {
    padding: 10px 15px;
    background: blueviolet;
    border-radius: 5px;
    cursor: pointer;
    font-weight: bold;
    color: white;
}
.showtime:hover {
    background: rgb(95, 2, 135);
    color: white;
}
/* Theater Listing */
.theater {
    background: white;
    padding: 15px;
    margin-top: 10px;
    border-radius: 5px;
    box-shadow: 0 0 5px rgba(0, 0, 0, 0.1);
}
```

```
.theater-name {
    font-size: 18px;
    font-weight: bold;
}
.info {
    font-size: 12px;
    color: gray;
}
/* Calendar Picker */
.calendar {
    margin: 20px 0;
}
input[type="date"] {
    padding: 10px;
    font-size: 16px;
    border-radius: 5px;
    border: 1px solid #ccc;
}
.selected-date {
    margin-top: 10px;
    font-weight: bold;
    color: #E50914;
}
.sidebar {
    display: flex;
    flex-direction: column;
    align-items: flex-end;
    gap: 10px;
    position: absolute;
    top: 40px;
    right: 50px;
}
```

```
.search-container {
    display: flex;
    align-items: center;
    gap: 10px;
}
.search-bar {
    padding: 10px;
    border: 1px solid #ccc;
    border-radius: 5px;
    width: 200px;
}
.dropdown {
    position: relative;
    display: inline-block;
    right: 50px;
}
.dropdown-content {
    display: none;
    position: absolute;
    background-color: white;
    min-width: 150px;
    box-shadow: 0px 4px 6px rgba(0, 0, 0, 0.1);
    padding: 10px;
    border-radius: 5px;
}
.dropdown:hover .dropdown-content {
    display: block;
}
.dropdown-content label {
    display: block;
    margin: 5px 0;
}
```

```
        button {
            background-color: blueviolet;
            color: white;
            padding: 8px 15px;
            border: none;
            cursor: pointer;
            border-radius: 5px;
        }
    </style>
</head>
<body>
    <header>
        <div class="logo">
            <img src="assets/logo.png" alt="MovieZone Logo">
        </div>
        <nav>
            <ul>
                <li><a href="home1.html">Home</a></li>
                <li><a href="login.html">Login</a></li>
                <li><a href="signup.html">Signup</a></li>
            </ul>
        </nav>
    </header><br>
    <div class="container">
        <div class="movie-title">Court: State vs A Nobody (Telugu)</div><br><br>
        <div class="tags">
            <span>Drama</span>
            <span>Thriller</span>
            <span>Telugu - 2D</span>
        </div><br>
        <!-- Calendar Picker -->
        <div class="calendar">
```

```
    <label for="movieDate">📅 Select a date:</label>
    <input type="date" id="movieDate">
    <p class="selected-date"></p>
</div>

<!-- Theaters Listing -->
<h3>Theaters Available</h3>
<div id="theater-list">
    <div class="theater" data-time="morning evening">
        <div class="theater-name">🎭 Cine Square Dolby Atmos A/C: Guntur</div>
        <div class="info">Non-cancellable</div>
        <div class="showtimes">
            <a href="fourth.html">
                <button class="showtime-btn">06:45 PM</button>
            </a>
        </div>
    </div>
    <div class="theater" data-time="afternoon night">
        <div class="theater-name">🎭 Cine Prime Screen 1 4K Atmos: Guntur</div>
        <div class="info">Cancellation Available</div>
        <div class="showtimes">
            <a href="fourth.html">
                <button class="showtime-btn">12:00 PM</button>
            </a>
            <a href="fourth.html">
                <button class="showtime-btn">1:45 PM</button>
            </a>
        </div>
    </div>
    <div class="theater" data-time="morning evening">
        <div class="theater-name">🎭 Harihar Cinemas: Guntur</div>
```

```
    <div class="info">Cancellation Available</div>
    <div class="showtimes">
      <a href="fourth.html">
        <button class="showtime-btn">07:00 AM</button>
      </a>
      <a href="fourth.html">
        <button class="showtime-btn">10:45 AM</button>
      </a>
      <a href="fourth.html">
        <button class="showtime-btn">6:00 PM</button>
      </a>
      <a href="fourth.html">
        <button class="showtime-btn">8:45 PM</button>
      </a>
    </div>
  </div>
  <div class="theater" data-time="afternoon night">
    <div class="theater-name">🎭 Siva Cinemas Dolby Atoms: Guntur</div>
    <div class="info">Cancellation Available</div>
    <div class="showtimes">
      <a href="fourth.html">
        <button class="showtime-btn">2:00 PM</button>
      </a>
      <a href="fourth.html">
        <button class="showtime-btn">4:45 PM</button>
      </a>
    </div>
  </div>
  <div class="theater" data-time="morning evening">
    <div class="theater-name">🎭 Hollywood Theatre: Guntur</div>
    <div class="info">Cancellation Available</div>
    <div class="showtimes">
```

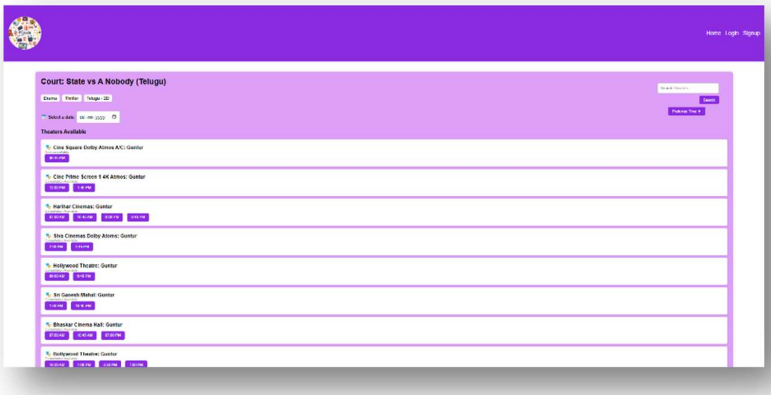
```
        <a href="fourth.html">
            <button class="showtime-btn">09:00 AM</button>
        </a>
        <a href="fourth.html">
            <button class="showtime-btn">6:45 PM</button>
        </a>
    </div>
</div>
<div class="theater" data-time="afternoon night">
    <div class="theater-name">🎭 Sri Ganesh Mahal: Guntur</div>
    <div class="info">Cancellation Available</div>
    <div class="showtimes">
        <a href="fourth.html">
            <button class="showtime-btn">1:00 PM</button>
        </a>
        <a href="fourth.html">
            <button class="showtime-btn">10:45 PM</button>
        </a>
    </div>
</div>
<div class="theater" data-time="morning evening">
    <div class="theater-name">🎭 Bhaskar Cinema Hall: Guntur</div>
    <div class="info">Cancellation Available</div>
    <div class="showtimes">
        <a href="fourth.html">
            <button class="showtime-btn">07:00 AM</button>
        </a>
        <a href="fourth.html">
            <button class="showtime-btn">10:45 AM</button>
        </a>
        <a href="fourth.html">
            <button class="showtime-btn">07:00 PM</button>
```

```
            </a>
          </div>
        </div>
        <div class="theater" data-time="morning afternoon evening">
          <div class="theater-name">🎭 Bollywood Theatre: Guntur</div>
          <div class="info">Cancellation Available</div>
          <div class="showtimes">
            <a href="fourth.html">
              <button class="showtime-btn">10:30 AM</button>
            </a>
            <a href="fourth.html">
              <button class="showtime-btn">1:00 PM</button>
            </a>
            <a href="fourth.html">
              <button class="showtime-btn">3:30 PM</button>
            </a>
            <a href="fourth.html">
              <button class="showtime-btn">7:00 PM</button>
            </a>
          </div>
        </div>
      </div>
      <div class="sidebar">
        <input type="text" class="search-bar" placeholder="Search Theaters...">
        <button onclick="searchTheaters()">Search</button>
        <div class="dropdown">
          <button>Preferred Time ▼</button>
          <div class="dropdown-content">
            <label><input type="checkbox" value="morning"> Morning (12:00 AM - 11:59 AM)</label>
            <label><input type="checkbox" value="afternoon"> Afternoon (12:00 PM - 3:59 PM)</label>
```

```
                <label><input type="checkbox" value="evening"> Evening (4:00 PM - 6:59 PM)</label>
                    <label><input type="checkbox" value="night"> Night (7:00 PM - 11:59 PM)</label>
            </div>
        </div>
    </div>
</div>
<script>
    // Date Picker Functionality
    document.getElementById("movieDate").addEventListener("change", function() {
        let selectedDate = this.value;
       document.querySelector(".selected-date").innerText = "Selected Date: " + selectedDate;
    });
    // Date Selection Highlight
    document.querySelectorAll(".date").forEach(date => {
        date.addEventListener("click", function() {
            document.querySelectorAll(".date").forEach(d => d.classList.remove("selected"));
            this.classList.add("selected");
        });
    });
    function searchTheaters() {
const searchQuery = document.querySelector('.search-bar').value.toLowerCase();
const checkboxes = document.querySelectorAll('.dropdown-content input:checked');
const selectedTimes = Array.from(checkboxes).map(cb => cb.value);
const theaters = document.querySelectorAll('.theater');
theaters.forEach(theater => {
    const theaterName = theater.querySelector('.theater-name').textContent.toLowerCase();
    const theaterTimes = theater.getAttribute('data-time').split(" "); // Split times into an array
    // Check if search query matches theater name
    const matchesSearch = !searchQuery || theaterName.includes(searchQuery);
    // Check if any selected time matches theater's available times
```

```
        const matchesTime = selectedTimes.length === 0 || selectedTimes.some(time => theaterTimes.includes(time));

        // Show or hide theater

        theater.style.display = matchesSearch && matchesTime ? "" : "none";

    });

}

    </script>

</body>

</html>
```

Output



FOURTH.html

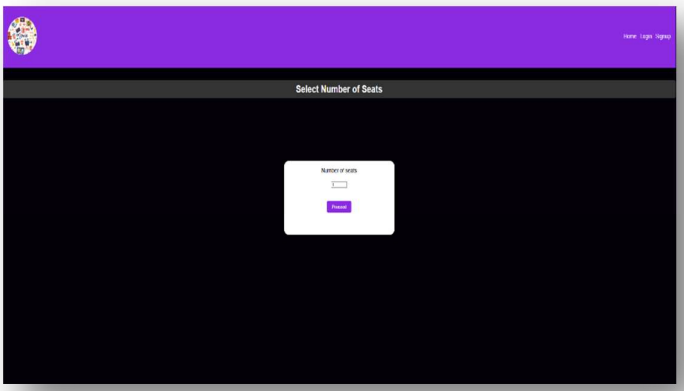
```
<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Select Number of Seats</title>

    <style>

        body {

            font-family: Arial, sans-serif;

            background: rgb(4, 0, 7);

            text-align: center;

        }
```

```
header {
    display: flex;
    justify-content: space-between;
    align-items: center;
    background: blueviolet;
    padding: 40px 20px;
    color: white;
}
.logo img {
    height: 50px;
}
nav ul {
    list-style: none;
    display: flex;
    gap: 15px;
    padding: 0;
}
nav ul li a {
    color: white;
    text-decoration: none;
    font-size: 20px;
}
h1 {
    color: white;
    background-color: #333;
    padding: 10px;
    border-radius: 5px;
}
.container {
    margin: auto;
    margin-top: 200px;
    width: 15%;
```

```
            background: white;
            padding: 20px;
            border-radius: 20px;
            box-shadow: 0px 4px 10px rgba(0, 0, 0, 0.2);
            text-align: center;
            font-size: 20px;
            height: 200px;
        }
        .proceed-btn {
            margin-top: 20px;
            padding: 10px 20px;
            background: blueviolet;
            color: white;
            border: none;
            font-size: 16px;
            cursor: pointer;
            border-radius: 5px;
        }
    </style>
</head>
<body>
    <header>
        <div class="logo">
            <img src="assets/logo.png" alt="MovieZone Logo">
        </div>
        <nav>
            <ul>
                <li><a href="home1.html">Home</a></li>
                <li><a href="login.html">Login</a></li>
                <li><a href="signup.html">Signup</a></li>
            </ul>
        </nav>
```

```
    </header><br>
    <h1>Select Number of Seats</h1>
    <div class="container">
        <label for="seatCount">Number of seats</label><br><br>
        <input type="number" id="seatCount" min="1" max="10" value="1"><br><br>
        <button class="proceed-btn" onclick="proceedToSeatSelection()">Proceed</button>
    </div>
    <script>
        function proceedToSeatSelection() {
            let seatCount = document.getElementById('seatCount').value;
            if (seatCount < 1 || seatCount > 10) {
                alert("Please enter a valid number of seats (1-10).");
                return;
            }
            localStorage.setItem('selectedSeatsCount', seatCount);
            window.location.href = 'fifth.html';
        }
    </script>
</body>
</html>
```

output



FIFTH.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Seat Layout</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            background: white;
            text-align: center;
        }
        header {
            display: flex;
            justify-content: space-between;
            align-items: center;
            background: blueviolet;
            padding: 40px 20px;
            color: white;
        }
        .logo img {
            height: 50px;
        }
        nav ul {
            list-style: none;
            display: flex;
            gap: 15px;
            padding: 0;
        }
        nav ul li a {
            color: white;
```

```
    text-decoration: none;
    font-size: 20px;
}
h1 {
    color: black;
}
.seat-wrapper {
    display: flex;
    justify-content: space-between; /* Places rows at left and right */
    width: 80%;
    margin: auto;
}

.row {
    display: flex;
    flex-wrap: wrap;
    gap: 15px;
    width: 45%; /* Each half takes up 45% width */
}
.container {
    margin: auto;
    margin-top: 140px;
    width: 50%;
    background: blueviolet;
    padding: 20px;
    border-radius: 20px;
    box-shadow: 0px 4px 10px rgba(0, 0, 0, 0.2);
    text-align: center;
    font-size: 18px;
    height: 100%;
}
.screen {
```

```
    width: 80%;
    margin: 20px auto;
    padding: 10px;
    background: lightgray;
    text-align: center;
    font-weight: bold;
}
.seat-container {
    display: flex;
    flex-direction: column;
    align-items: center;
    gap: 10px;
}
.row {
    display: flex;
    justify-content: center;
    gap: 10px;
}
.seat {
    width: 30px;
    height: 30px;
    border-radius: 5px;
    text-align: center;
    line-height: 30px;
    font-size: 14px;
    cursor: pointer;
    background: #eee;
    border: 1px solid #bbb;
}
.seat.sold {
    background: #999;
    cursor: not-allowed;
```

```
}
.seat.selected {
    background: green;
    color: white;
}
.legend {
    display: flex;
    justify-content: center;
    gap: 20px;
    margin-top: 20px;
}
.legend div {
    display: flex;
    align-items: center;
    gap: 5px;
}
.legend span {
    width: 20px;
    height: 20px;
    border-radius: 5px;
    display: inline-block;
}
.legend .available-seat { background: white; border: 1px solid #bbb; }
.legend .selected-seat { background: green; }
.legend .sold-seat { background: #999; }
.total-price {
    margin-top: 20px;
    font-size: 18px;
    font-weight: bold;
}
.confirm-btn {
    margin-top: 20px;
```

```
            padding: 10px 20px;
            background: #ce46ff;
            color: white;
            border: none;
            font-size: 16px;
            cursor: pointer;
            border-radius: 5px;
        }
    </style>
</head>
<body>
    <header>
        <div class="logo">
            <img src="assets/logo.png" alt="MovieZone Logo">
        </div>
        <nav>
            <ul>
                <li><a href="home1.html">Home</a></li>
                <li><a href="login.html">Login</a></li>
                <li><a href="signup.html">Signup</a></li>
            </ul>
        </nav>
    </header><br>
    <br><br><br>
    <h1>SEAT LAYOUT</h1>
    <div class="container">
        <div class="seat-container">
            <!-- Row A -->
            <div class="seat-wrapper">
                <!-- Left Side -->
                <div class="row">
                    <div class="seat available">1</div>
```

```
        <div class="seat available">2</div>
        <div class="seat available">3</div>
        <div class="seat available">4</div>
        <div class="seat available">5</div>
        <div class="seat available">6</div>
        <div class="seat available">7</div>
        <div class="seat available">8</div>
    </div>
    <!-- Right Side -->
    <div class="row">
        <div class="seat available">10</div>
        <div class="seat available">11</div>
        <div class="seat available">12</div>
        <div class="seat available">13</div>
        <div class="seat available">14</div>
        <div class="seat available">15</div>
        <div class="seat available">16</div>
        <div class="seat available">17</div>
    </div>
</div>
<!-- Row B -->
<div class="seat-wrapper">
    <!-- Left Side -->
    <div class="row">
        <div class="seat available">18</div>
        <div class="seat available">19</div>
        <div class="seat available">20</div>
        <div class="seat available">21</div>
        <div class="seat available">22</div>
        <div class="seat available">23</div>
        <div class="seat available">24</div>
        <div class="seat available">25</div>
```

```
    </div>
    <!-- Right Side -->
    <div class="row">
        <div class="seat available">26</div>
        <div class="seat available">27</div>
        <div class="seat available">28</div>
        <div class="seat available">29</div>
        <div class="seat available">30</div>
        <div class="seat available">31</div>
        <div class="seat available">32</div>
        <div class="seat available">33</div>
    </div>
</div>
<div class="seat-wrapper">
    <!-- Left Side -->
    <div class="row">
        <div class="seat available">34</div>
        <div class="seat available">35</div>
        <div class="seat available">36</div>
        <div class="seat available">37</div>
        <div class="seat available">38</div>
        <div class="seat available">39</div>
        <div class="seat available">40</div>
        <div class="seat available">41</div>
    </div>
    <!-- Right Side -->
    <div class="row">
        <div class="seat available">42</div>
        <div class="seat available">43</div>
        <div class="seat available">44</div>
        <div class="seat available">45</div>
        <div class="seat available">46</div>
```

```
            <div class="seat available">47</div>
            <div class="seat available">48</div>
            <div class="seat available">49</div>
        </div>
    </div>
    <div class="seat-wrapper">
        <!-- Left Side -->
        <div class="row">
            <div class="seat available">50</div>
            <div class="seat available">51</div>
            <div class="seat available">52</div>
            <div class="seat available">53</div>
            <div class="seat available">54</div>
            <div class="seat available">55</div>
            <div class="seat available">56</div>
            <div class="seat available">57</div>
        </div>
        <!-- Right Side -->
        <div class="row">
            <div class="seat available">58</div>
            <div class="seat available">59</div>
            <div class="seat available">60</div>
            <div class="seat available">61</div>
            <div class="seat available">62</div>
            <div class="seat available">63</div>
            <div class="seat available">64</div>
            <div class="seat available">65</div>
        </div>
    </div>
    <div class="seat-wrapper">
        <!-- Left Side -->
        <div class="row">
```

```
            <div class="seat available">66</div>
            <div class="seat available">67</div>
            <div class="seat available">68</div>
            <div class="seat available">69</div>
            <div class="seat available">70</div>
            <div class="seat available">71</div>
            <div class="seat available">72</div>
            <div class="seat available">73</div>
        </div>
        <!-- Right Side -->
        <div class="row">
            <div class="seat available">74</div>
            <div class="seat available">75</div>
            <div class="seat available">76</div>
            <div class="seat available">77</div>
            <div class="seat available">78</div>
            <div class="seat available">79</div>
            <div class="seat available">80</div>
            <div class="seat available">81</div>
        </div>
    </div>
    <div class="seat-wrapper">
        <!-- Left Side -->
        <div class="row">
            <div class="seat available">82</div>
            <div class="seat available">83</div>
            <div class="seat available">84</div>
            <div class="seat available">85</div>
            <div class="seat available">86</div>
            <div class="seat available">87</div>
            <div class="seat available">88</div>
            <div class="seat available">89</div>
```

```
    </div>
    <!-- Right Side -->
    <div class="row">
        <div class="seat available">90</div>
        <div class="seat available">91</div>
        <div class="seat available">92</div>
        <div class="seat available">93</div>
        <div class="seat available">94</div>
        <div class="seat available">95</div>
        <div class="seat available">96</div>
        <div class="seat available">97</div>
    </div>
</div>
<div class="seat-wrapper">
    <!-- Left Side -->
    <div class="row">
        <div class="seat available">98</div>
        <div class="seat available">99</div>
        <div class="seat available">100</div>
        <div class="seat available">101</div>
        <div class="seat available">102</div>
        <div class="seat available">103</div>
        <div class="seat available">104</div>
        <div class="seat available">105</div>
    </div>
    <!-- Right Side -->
    <div class="row">
        <div class="seat available">106</div>
        <div class="seat available">107</div>
        <div class="seat available">108</div>
        <div class="seat available">109</div>
        <div class="seat available">110</div>
```

```
        <div class="seat available">111</div>
        <div class="seat available">112</div>
        <div class="seat available">113</div>
    </div>
</div>
<div class="seat-wrapper">
    <!-- Left Side -->
    <div class="row">
        <div class="seat available">114</div>
        <div class="seat available">115</div>
        <div class="seat available">116</div>
        <div class="seat available">117</div>
        <div class="seat available">118</div>
        <div class="seat available">119</div>
        <div class="seat available">120</div>
        <div class="seat available">121</div>

    </div>
    <!-- Right Side -->
    <div class="row">
        <div class="seat available">122</div>
        <div class="seat available">123</div>
        <div class="seat available">124</div>
        <div class="seat available">125</div>
        <div class="seat available">126</div>
        <div class="seat available">127</div>
        <div class="seat available">128</div>
        <div class="seat available">129</div>
    </div>
</div>
<div class="seat-wrapper">
    <!-- Left Side -->
```

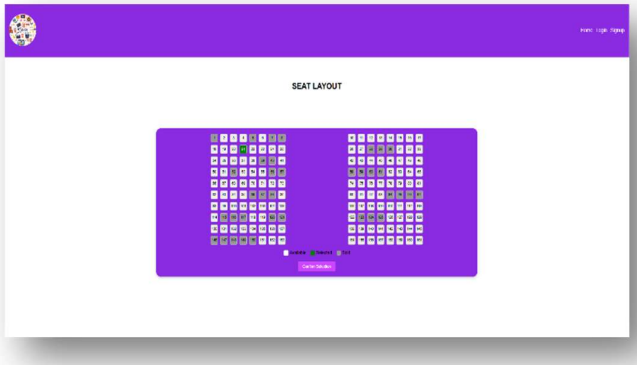
```
    <div class="row">
        <div class="seat available">130</div>
        <div class="seat available">131</div>
        <div class="seat available">132</div>
        <div class="seat available">133</div>
        <div class="seat available">134</div>
        <div class="seat available">135</div>
        <div class="seat available">136</div>
        <div class="seat available">137</div>
    </div>
    <!-- Right Side -->
    <div class="row">
        <div class="seat available">138</div>
        <div class="seat available">139</div>
        <div class="seat available">140</div>
        <div class="seat available">141</div>
        <div class="seat available">142</div>
        <div class="seat available">143</div>
        <div class="seat available">144</div>
        <div class="seat available">145</div>
    </div>
</div>
<div class="seat-wrapper">
    <!-- Left Side -->
    <div class="row">
        <div class="seat available">146</div>
        <div class="seat available">147</div>
        <div class="seat available">148</div>
        <div class="seat available">149</div>
        <div class="seat available">150</div>
        <div class="seat available">151</div>
        <div class="seat available">152</div>
```

```
            <div class="seat available">153</div>
        </div>
        <!-- Right Side -->
        <div class="row">
            <div class="seat available">154</div>
            <div class="seat available">155</div>
            <div class="seat available">156</div>
            <div class="seat available">157</div>
            <div class="seat available">158</div>
            <div class="seat available">159</div>
            <div class="seat available">160</div>
            <div class="seat available">161</div>
        </div>
    </div>
</div>
<div class="legend">
    <div><span class="available-seat"></span> Available</div>
    <div><span class="selected-seat"></span> Selected</div>
    <div><span class="sold-seat"></span> Sold</div>
</div>
<button class="confirm-btn" onclick="confirmSelection()">Confirm Selection</button>
</div>
<script>
    let maxSeats = localStorage.getItem('selectedSeatsCount') || 1;
    maxSeats = parseInt(maxSeats);
    let selectedSeats = 0;
    document.querySelectorAll('.seat.available').forEach(seat => {
        seat.addEventListener('click', function () {
            if (this.classList.contains('selected')) {
                this.classList.remove('selected');
                selectedSeats--;
            } else {
```

```
            if (selectedSeats < maxSeats) {
                this.classList.add('selected');
                selectedSeats++;
            } else {
                alert(`You can only select ${maxSeats} seats.`);
            }
        }
    });
});
    function confirmSelection() {
let selectedSeatNumbers = [];
document.querySelectorAll('.seat.selected').forEach(seat => {
    selectedSeatNumbers.push(seat.textContent);
});
if (selectedSeatNumbers.length === 0) {
    alert("Please select at least one seat.");
    return;
}
// Store selected seats in localStorage for payment summary page
localStorage.setItem('selectedSeats', JSON.stringify(selectedSeatNumbers));
fetch('http://localhost:3000/book-seats', {
    method: 'POST',
    headers: { 'Content-Type': 'application/json' },
                    body: JSON.stringify({ seats: selectedSeatNumbers, user:
localStorage.getItem('username') })
})
.then(response => response.json())
.then(data => {
    if (data.success) {
        alert("Seats booked successfully!");
        // Redirect to payment summary page
        window.location.href = 'payment-summary.html';
```

```
        } else {
            alert("Some seats were already booked. Try again.");
            location.reload();
        }
    })
    .catch(error => console.error('Error:', error));
}
        fetch('http://localhost:3000/get-booked-seats')
            .then(response => response.json())
            .then(data => {
                data.bookedSeats.forEach(seatNumber => {
                        let seatElement = [...document.querySelectorAll('.seat')].find(seat => seat.textContent === seatNumber);
                    if (seatElement) {
                        seatElement.classList.add('sold');
                        seatElement.classList.remove('available');
                    }
                });
            });
    </script>
</body>
</html>
```

Output:

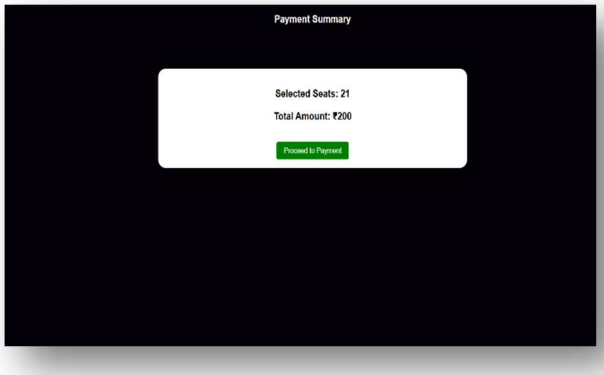


PAYMENT-SUMMARY.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Payment Summary</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            text-align: center;
            background: rgb(4, 0, 7);
            color: white;
        }
        .container {
            margin: auto;
            margin-top: 100px;
            width: 40%;
            background: white;
            padding: 20px;
            border-radius: 20px;
            box-shadow: 0px 4px 10px rgba(0, 0, 0, 0.2);
            text-align: center;
            font-size: 20px;
            color: black;
        }
        .summary {
            font-size: 24px;
            font-weight: bold;
        }
        .proceed-btn {
            margin-top: 20px;
```

```
            padding: 10px 20px;
            background: green;
            color: white;
            border: none;
            font-size: 18px;
            cursor: pointer;
            border-radius: 5px;
        }
    </style>
</head>
<body>
    <h2>Payment Summary</h2>
    <div class="container">
        <p class="summary">Selected Seats: <span id="selectedSeats"></span></p>
        <p class="summary">Total Amount: ₹<span id="totalAmount"></span></p>
                    <button class="proceed-btn" onclick="proceedToPayment()">Proceed to Payment</button>
    </div>
    <script>
        const seatPrice = 200;
        let selectedSeats = JSON.parse(localStorage.getItem("selectedSeats")) || [];
        let totalAmount = selectedSeats.length * seatPrice;
        document.getElementById("selectedSeats").innerText = selectedSeats.join(", ");
        document.getElementById("totalAmount").innerText = totalAmount;
        function proceedToPayment() {
            window.location.href="payment-successful.html";
        }
    </script>
</body>
</html>
```

Output



PAYMENT-SUCCESSFUL.html

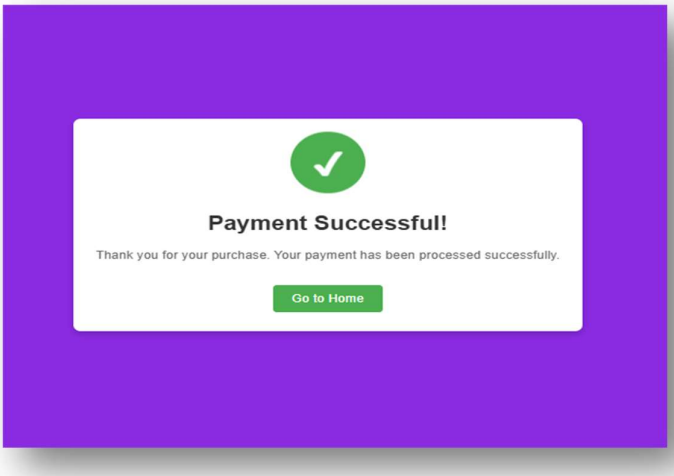
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Payment Successful</title>
    <style>
        body {
            display: flex;
            justify-content: center;
            align-items: center;
            height: 100vh;
            background: #f4f4f4;
            font-family: Arial, sans-serif;
            flex-direction: column;
            text-align: center;
            overflow: hidden;
        }
        .container {
            background: white;
```

```
    padding: 30px;
    border-radius: 10px;
    box-shadow: 0 4px 10px rgba(0, 0, 0, 0.2);
    animation: fadeIn 1s ease-in-out;
    transform: scale(0.8);
    animation: scaleUp 0.6s ease-in-out forwards;
}
.checkmark {
    width: 100px;
    height: 100px;
    border-radius: 50%;
    background: #4CAF50;
    display: flex;
    justify-content: center;
    align-items: center;
    margin: auto;
    position: relative;
    animation: bounce 1s ease-in-out infinite alternate;
}
.checkmark::after {
    content: "✓";
    font-size: 50px;
    color: white;
    font-weight: bold;
    position: absolute;
    animation: popIn 0.5s ease-in-out forwards;
}
h1 {
    color: #333;
    margin-top: 20px;
    opacity: 0;
    animation: slideIn 1s ease-in-out 0.5s forwards;
```

```
}
p {
    font-size: 18px;
    color: #666;
    opacity: 0;
    animation: slideIn 1s ease-in-out 0.7s forwards;
}
.btn {
    display: inline-block;
    margin-top: 20px;
    padding: 12px 25px;
    background: #4CAF50;
    color: white;
    text-decoration: none;
    font-size: 18px;
    border-radius: 5px;
    transition: 0.3s;
    opacity: 0;
    animation: slideIn 1s ease-in-out 0.9s forwards;
}
.btn:hover {
    background: #388E3C;
    box-shadow: 0 0 15px rgba(72, 239, 128, 0.6);
}
/* Keyframes for Animations */
@keyframes fadeIn {
    from { opacity: 0; transform: translateY(-20px); }
    to { opacity: 1; transform: translateY(0); }
}
@keyframes popIn {
    from { transform: scale(0); opacity: 0; }
    to { transform: scale(1); opacity: 1; }
```

```
        }
        @keyframes bounce {
            from { transform: translateY(0); }
            to { transform: translateY(-10px); }
        }
        @keyframes scaleUp {
            from { transform: scale(0.8); opacity: 0; }
            to { transform: scale(1); opacity: 1; }
        }
        @keyframes slideIn {
            from { transform: translateY(20px); opacity: 0; }
            to { transform: translateY(0); opacity: 1; }
        }
    </style>
</head>
<body>
    <div class="container">
        <div class="checkmark"></div>
        <h1>Payment Successful!</h1>
        <p>Thank you for your purchase. Your payment has been processed successfully.</p>
        <a href="home1.html" class="btn">Go to Home</a>
    </div>
</body>
</html>
```

Output:



Database Connectivity code:

.env:

MONGO_URI= mongodb://127.0.0.1:27017/movieApp

PORT=3000

App.js

```
require('dotenv').config();
const express = require('express');
const bodyParser = require('body-parser');
const mongoose = require('mongoose');
const bcrypt = require('bcrypt');
const path = require('path');
const User = require('./models/User');
const Movie = require('./models/Movie'); // Import only once
const cors = require("cors");
const jwt = require("jsonwebtoken");
const app = express();
// Middleware
```

```
app.use(express.json());
app.use(express.urlencoded({ extended: false }));
app.use(express.static(path.join(__dirname, 'public')));
app.use(cors());
app.use(bodyParser.json());

// MongoDB Connection
const uri = process.env.MONGO_URI;
if (!uri) {
   console.error('MongoDB URI not defined. Check your .env file.');
   process.exit(1);
}
mongoose
   .connect(uri)
   .then(() => console.log('Connected to MongoDB'))
   .catch(err => {
      console.error('Error connecting to MongoDB:', err);
      process.exit(1);
   });
// Routes
app.get('/', (req, res) => {
      res.send('<h1>Welcome</h1><a  href="/register.html">Register</a>  |  <a
href="/login.html">Login</a>');
});
// ✅ User Registration
app.post('/register', async (req, res) => {
   try{
   const {name, email, password } = req.body;
   if (!name || !email || !password) {
```

```
        return res.status(400).json({ error: "All fields are required." });
    }
        const existingUser = await User.findOne({ email });
        if (existingUser) {
            return res.status(400).json({ error: "Email already registered." });
        }
        const hashedPassword = await bcrypt.hash(password, 10);
        const newUser = new User({ name, email, password: hashedPassword });
        await newUser.save();
          return res.json({ success: true, message: "User registered successfully!" });return res.json({ success: true, message: "User registered successfully!" });
    } catch (err) {
        console.error('Error during registration:', err);
        return res.status(500).json({ error: "Server error." });
    }
});
// ✅ User & Admin Login
app.post('/login', async (req, res) => {
    const { email, password } = req.body;
    if (!email || !password) {
        return res.status(400).send('<h1>All fields are required.</h1>');
    }
    try {
        console.log("🔍 Searching for user with email:", email);
        const user = await User.findOne({ email });
        console.log("🛠️ Found user:", user);
        if (!user) {
            return res.status(400).json({ error: "User not found!" });
        }
```

```
        console.log("🔑 Entered password:", password);
        console.log("🔒 Stored hashed password:", user.password);

        const isMatch = await bcrypt.compare(password, user.password);
        console.log("✅ Password Match Status:", isMatch);
        if (!isMatch) {
            return res.status(400).json({ error: "Invalid credentials!" });
        }
        // Generate a token (optional but recommended)
        const token = jwt.sign({ userId: user._id }, "yourSecretKey", { expiresIn: "1h" });
        res.json({ message: "Login successful", token });
    } catch (error) {
        console.error("Login error:", error);
        res.status(500).json({ error: "Server error" });
    }
});
// ✅ Admin Dashboard - View All Users
app.get('/admin', async (req, res) => {
    try {
        const users = await User.find({}, '-password'); // Exclude passwords
        const userCount = await User.countDocuments();
        let userTable = '<h1>All Users</h1><table border="1"><tr><th>Username</th><th>Email</th><th>Role</th></tr>';
        users.forEach(user => {
            userTable += `<tr><td>${user.username}</td><td>${user.email}</td><td>${user.role}</td></tr>`;
        });
        userTable += '</table>';
```

```
                res.send(`<h1>Admin Dashboard</h1><h2>Total Users: ${userCount}</h2>${userTable}`);
    } catch (err) {
        console.error('Error fetching users:', err);
        res.status(500).send('<h1>Internal Server Error</h1>');
    }
});
// Add a new movie
app.post('/add-movie', async (req, res) => {
    const { title, genre } = req.body;
    try {
        const newMovie = new Movie({ title, genre });
        await newMovie.save();
        res.json({ message: "Movie added successfully", movie: newMovie });
    } catch (error) {
        res.status(500).json({ error: "Failed to add movie" });
    }
});
// Get all movies
app.get('/movies', async (req, res) => {
    try {
        const movies = await Movie.find();
        res.json(movies);
    } catch (error) {
        res.status(500).json({ error: "Failed to fetch movies" });
    }
});
// Seat Schema
const seatSchema = new mongoose.Schema({
```

```
    seatNumber: String,
    bookedBy: String
});

const Seat = mongoose.model("Seat", seatSchema);
// Get all booked seats
app.get("/get-booked-seats", async (req, res) => {
    try {
        const bookedSeats = await Seat.find();
        res.json({ bookedSeats: bookedSeats.map(seat => seat.seatNumber) });
    } catch (error) {
        res.status(500).json({ error: "Internal Server Error" });
    }
});
// Book seats
app.post("/book-seats", async (req, res) => {
    const { seats, user } = req.body;
    try {
        // Check if seats are already booked
        const existingBookings = await Seat.find({ seatNumber: { $in: seats } });
        if (existingBookings.length > 0) {
            return res.status(400).json({ success: false, message: "Some seats are already booked!" });
        }
        // Save new bookings
        const newBookings = seats.map(seat => new Seat({ seatNumber: seat, bookedBy: user }));
        await Seat.insertMany(newBookings);
        res.json({ success: true });
```

```
    } catch (error) {
        res.status(500).json({ error: "Internal Server Error" });
    }
});
// ✅ Start the server (Only Once)
const PORT = process.env.PORT || 3000;
app.listen(PORT, () => {
    console.log(`Server running at http://localhost:${PORT}`);
});
```

Output: