

Technical Report for Movie Rating Project

1. Basic Information:

Title: *Movie Rating System*

Short Description:

The Movie Rating project allows users to register, log in, rate movies, and interact with a movie catalog. It supports dynamic querying and filtering of movies, and provides secure user authentication and role-based access controls.

Authors:

Afreen Saheer

Aycan Genc

Rumeysa Dikmen

Functionalities:

- User registration and login.
- Movie CRUD operations (admin-only).
- Rating functionality for users to rate movies.
- Query and filter movies based on genre, title, or rating.
- Authentication and role-based access control.

Technologies Used:

- **Backend:** ASP.NET Core, Entity Framework, JWT authentication, REST API/GraphQL API.
- **Frontend:** React (for Full Stack), HTML, CSS.
- **Database:** SQL Server.
- **Development Tools:** Visual Studio, Postman, Swagger.

Components Description:

- **Backend:**
 - **Controllers:** Manages user, movie, and rating operations (e.g., UserController, MovieController, RatingController).
 - **Models:** Represents data entities like User, Movie, and Rating.
 - **Authentication:** JWT-based authentication for user login and secure access to endpoints.
 - **Database:** Uses Entity Framework to interact with the SQL Server database.

2. Backend Description:

Structure of the Backend:

The backend consists of multiple components, each designed to handle specific tasks within the system. The components are grouped into controllers, models, services, and data handling mechanisms.

- **Controllers:** These handle incoming HTTP requests and return responses. They are responsible for defining the routes and managing business logic for each entity (users, movies, and ratings).
 - Example: UserController.cs, MovieController.cs, RatingController.cs.
- **Models:** Represent the data entities in the system. The models define the structure of the data and facilitate interactions with the database.
 - Example: User.cs, Movie.cs, Rating.cs.
- **Services/Business Logic:** If applicable, services encapsulate business logic like user authentication and movie recommendations.
 - Example: AuthenticationService.cs, MovieService.cs.
- **Database Context:** Utilizes Entity Framework to handle interactions with the database and manage data persistence.
 - Example: ApplicationDbContext.cs.

3. Frontend Description:

Structure of the Frontend:

The frontend consists of React components and static HTML files, providing a user interface for interaction with the backend API. The main components are grouped by functionality:

- **User Authentication:** Includes components for registration and login (LoginForm.js, RegisterForm.js).
- **Movie Browsing:** Displays the list of movies and allows filtering and sorting by title, genre, and rating (MovieList.js, MovieCard.js).
- **Rating:** Allows users to submit ratings for movies and view their rating history (RatingForm.js, UserRatings.js).

4. User System:

Roles in the System:

The system uses role-based access control to differentiate between two types of users:

- **Guest:**
 - Can view the movie catalog and ratings.
 - Cannot rate movies or access restricted sections like the admin panel.

- **User:**
 - Can register and log in.
 - Can rate movies and view their own rating history.
 - Access to the personal account dashboard and movie ratings.

Privileges:

- **Guest:**
 - Read-only access to movie information.
 - Can browse movies but cannot perform actions like rating or account management.
- **User:**
 - Full access to all features except admin-only functionalities.
 - Can rate movies, edit personal information, and access movie recommendations.

5. User Experience:

Instructions for Usage:

- **Running the Project:**
 1. Clone the repository to your local machine.
 2. Follow the setup instructions in the README.md for both the frontend and backend (ensure that the database is set up correctly).
 3. Launch the backend API server and the frontend application.
- **Using the Application:**
 1. **Guest View:**
 - Navigate to the homepage to browse movies.
 - Filter movies by genre or rating.
 - No login is required to view movie information.
 2. **User View:**
 - Register for an account or log in.
 - Once logged in, rate movies from the movie list or view your rating history.
 - Navigate to the personal dashboard for additional functionalities.

Screenshots:

- Login Screen:

[movierecommend](#) [Home](#) [Privacy](#) [ratings](#) [movies](#) [Register](#) [Login](#)

Log in

Use a local account to log in.

Email
afreen@gmail.com

Password
••••••

☒ Remember me?

Log in

[Forgot your password?](#)

[Register as a new user](#)

[Resend email confirmation](#)

Use another service to log in.

There are no external authentication services configured. See this [article about setting up this ASP.NET application to support logging in via external services](#).

- Registering Screen:

[movierecommend](#) [Home](#) [Privacy](#) [ratings](#) [movies](#) [Register](#) [Login](#)

Register

Create a new account.

Email
afreen@gmail.com

Password
••••••

Confirm Password
••••••

Register

Use another service to register.

There are no external authentication services configured. See this [article about setting up this ASP.NET application to support logging in via external services](#).

© 2025 - movierecommend - [Privacy](#)

- **Movie List:**

← → ↺ 🏠 📄 localhost:7223/movies ☆ 🔥 📦 🔄 ⋮

movierecommend Home Privacy ratings movies

Register Login

Index

[Create New](#)

Title	Genres	Description	
Barbie	Comedy	Barbie is an 11-inch- (29-cm-) tall plastic doll with the figure of an adult woman that was originally modeled on the Lilli doll (produced in West Germany from 1955 to 1964), a risqué gag gift for men. The Barbie target demographic is young children. Barbie's full name is Barbara Millicent Roberts.	Edit Details Delete

© 2025 - movierecommend - [Privacy](#)

- **Rating Form:**

movierecommend Home Privacy ratings movies

Register Login

Index

[Create New](#)

Title	Rating	
Barbie	5	Edit Details Delete

© 2025 - movierecommend - [Privacy](#)