Project Report: Movie Database Web Application

Group Members:

18050111030 Ekin Doğucan Akkaya

18050111046 Behlül Mansur Çıldır

18050111005 Enes Güler

18050171001 Fatih Gülmez

Introduction and Project Description:

The movie website is a database web application developed as part of the CENG423 course at Ankara Yıldırım Beyazıt University. Our project aims to create a movie database web application that aligns with the objectives of our course.

We developed a web-based platform called Movie Database, which will compile movies released from the past to the present. Users will be able to rate, comment, and search for movies they want to watch and will get an idea about these movies. The application will provide a comprehensive platform for users to engage with movies, giving them the opportunity to express their opinions, explore other users' views, and access detailed information about movies.

To store the data, we used Microsoft SQL. We sent the database as backup. The comments of the users and the movies are stored properly in the database.

Implementation:

The application was implemented using .NET Core 6.0, a popular web development framework, Entity Framework Core simplifies database interactions and maps database tables to C# classes. MVC (Model-View-Controller) Design Pattern, Bootstrap, HTML, CSS, and JavaScript. Endpoints are used. The following features were developed to meet the project requirements:

User Registration and Authentication:

Users can register an account by providing an email, name and password. User authentication and authorization were implemented to secure certain parts of the application, ensuring that only authenticated users can access features. We used cookies for authentication and NETCore.Encrypt; a library that is used for cryptographic operations such as hashing. We separated the pages that can an admin and a user can reach. Made rules for authorization. Users passwords became unique hashes by implementing salting and hashing.

User:

Users usernames are determined as their emails. By this way we acquired global uniqueness. The user also can change their shown names, emails and passwords in profile page.

Profile:

A user profile page was provided, allowing users to view and update their personal information. This feature enhances the user experience by providing personalized options and easy management of user details.

Admin Panel:

Admins can use three main functions in the admin panel. They can add more movies, update already existing movies or delete the existing movies. Admins also can delete the comments of other users for user management. Admins can give admin privilege to another user when they are editing their profiles.

Movie Catalog:

A database was designed and implemented to store information about movies and users. The movie database schema includes attributes such as title, director, actors, category, rate, duration, description, publish year, image and trailer. The user database schema includes attributes such as name, mail, password, role. Users can browse the movie catalog, view detailed information about each movie as well as their star ratings, and search for movies based on title, director, artist, category, rate, produce year and minute.

Filtering:

We enable users to filter movies and find the movie they are looking for. Users can filter movies in various ways. Movie title, director, actors, rate, duration and release year. In addition, the validation process in the filtering form is done with JavaScript. This prevents the user from entering incorrectly. Also, the category list in the Filtering form comes from the database, not hardcoded.

Giving ratings:

Functionality was implemented to allow users to give ratings for the movie in movies page. Every user only can have one rating for each movie. They can update their existing ratings easily by just clicking different star rating. When a rate was given, the rating details are stored in the database, and then an overall star rating is calculated for the movie. The calculated overall star rating can be seen when the movie list is opened, or details of a movie is opened.

Commenting:

On our project, users can share their thoughts and opinions by leaving comments. This feature enhances the experience of the platform, allowing users to engage in discussions and express their viewpoints about different movies. Through the ability to leave comments, our project creates an inclusive and collaborative space, where users can actively participate and contribute to the dialogue about their favorite movies.

Additional Features:

As an optional addition, the application includes a Contact us page and a privacy policy page. We used Entity framework to make it easier on database operations, to prevent coupling, dependency injection is used for connecting the database, If in the future database is changed for example if we pass to MySQL, we can easily adapt our project.

Evaluation:

The project was evaluated based on the following criteria:

- ### Functionality:

Our application successfully meets the specified requirements which are movie catalog, user comments, and rating features. Users can register, authenticate, browse movies, give rating to them, and manage their user profiles. The core functionality of an online movie website was effectively implemented.

- ### Design and Architecture:

The application follows the practices that are shown in the lessons and is well-structured. The design choices we made consider scalability, maintainability, and code reusability. The use of .NET Core(Cross platform access) provides a solid foundation for the application's architecture.

- ### User Experience:

The user experience is intuitive and user-friendly. Our user interface design focuses on simplicity. Users can easily navigate through the home page, search for movies, and manage their profiles. Our interface is clean and unsophisticated, thus increasing the user experience further.

- ### Code Quality:

Our code is clean, well-organized, and properly documented. Naming conventions are followed, and code comments provide clarity and context. The codebase demonstrates good coding practices, making it maintainable and extensible.

Target Audience:

The target audience for our project includes movie enthusiasts who enjoy expressing their opinions, individuals searching for suitable movies, and people seeking insights through movie reviews.

Potential Impact and Benefits:

Our project has the potential to make a significant impact on the film industry and cultural trends. It can serve as a valuable resource for movie enthusiasts, contributing to their understanding of cinema and television culture. Additionally, it can provide entertainment options for individuals during their leisure time. Our project can be developed furthermore in future.

Acquirements:

When we did this final project, we got a better understanding of MVC logic, how .NET core works and how the entity framework works. By implementing the learnings that we saw on the lessons and in assignments, now we understand these topics much better than when we first start this term. Teamwork capabilities of all group members are developed further.

Conclusion:

Overall, our project utilizes.NET Core MVC as the main technology and uses Entity Framework Core for database operation. The Movie Database Web Application project successfully implements the core functionality of a movie database web application. Users can register, authenticate, browse and search for movies, Filter the search, watch their trailers, manage their comment and ratings and update their user profiles. Admins can add more movies, delete or edit the existing movies and can delete the comments of users. The application's design and architecture stick to our learnings in the class and best practices, ensuring scalability and maintainability. The user interface provides an intuitive and visually simple experience. The code is of high quality, with proper documentation. GitHub is used extensively to document the every step that all of our group members had took.