|  |
| --- |
| Software Requirement Specifications (SRS) For Movie Recommender System |
| Implementation of Pearson’s Correlation |

|  |
| --- |
| Shahir Jalal  7-2-2023 |

Contents

[1. Introduction 2](#_Toc126814078)

[1.1 Purpose 2](#_Toc126814079)

[1.2 Intended Audience and Reading Suggestion 2](#_Toc126814080)

[1.3 Project Scope 2](#_Toc126814081)

[2. Overall Description 4](#_Toc126814082)

[2.1 Product Perspective 4](#_Toc126814083)

[3. Software Features 6](#_Toc126814084)

[3.1 Login Page 6](#_Toc126814085)

[3.2 Registration Page 6](#_Toc126814086)

[3.3 List of Users Page 7](#_Toc126814087)

[3.4 Add User Page 8](#_Toc126814088)

[3.4.1 Add User Flowchart 10](#_Toc126814089)

[3.5 Delete User Function 10](#_Toc126814090)

[3.6 Update User Page 11](#_Toc126814091)

[3.6.1 Update User Flowchart 13](#_Toc126814092)

[3.7 List of Movies Page 14](#_Toc126814093)

[3.8 Add Movie Page 15](#_Toc126814094)

[3.8.1 Add Movie Flowchart 16](#_Toc126814095)

[3.9 Update Movie Page 17](#_Toc126814096)

[3.9.1 Update Movie Flowchart 18](#_Toc126814097)

[3.10 Delete Movie Function 19](#_Toc126814098)

[3.11 List of Recommended Movies Page 20](#_Toc126814099)

[4. Entity Relationship Diagram (ERD) 22](#_Toc126814100)

[5. Pearson’s Correlation 24](#_Toc126814101)

[5.1 Generating Similarity Matrix 25](#_Toc126814102)

[6. References 27](#_Toc126814103)

# Introduction

## Purpose

The purpose of this SRS (Software Requirements Specification) is to provide a comprehensive overview and description of the Movie Recommender System. The purpose is to outline the functional and non-functional requirements of the software, including its purpose, intended audience, and project scope. This SRS will serve as a reference guide for the development team and stakeholders to ensure that the software is built to meet the needs and expectations of its users. It will also provide a clear understanding of the features, functions, and performance criteria of the software, helping to ensure that it delivers the desired level of performance and user satisfaction.

## Intended Audience and Reading Suggestion

The intended audience for this SRS (Software Requirements Specification) are software developers, stakeholders, and project managers who are involved in the development, implementation, and maintenance of the Movie Recommender System. This document is also intended for quality assurance personnel and users who need to understand the requirements of the software.

Reading Suggestion: It is recommended that readers of this SRS have a basic understanding of software development and movie recommendation systems. Familiarity with Pearson's Correlation, user interface design, and software testing methods is also recommended. Before reading this SRS, it is advisable to have a clear understanding of the project scope, goals, and objectives.

## Project Scope

The project scope of the Movie Recommender System encompasses the design, development, and implementation of a movie recommendation system that utilizes Pearson's Correlation. The system will be able to recommend movies to users based on their movie choice. The project scope includes the following features:

1. User registration and login system: The software will allow users to create an account and login to access the recommendation system.
2. Movie database: The software will have a database of movies that can be recommended to users.
3. Recommendation engine: The recommendation engine will use Pearson's Correlation to generate movie recommendations for users based on their movie choice.
4. User interface: The software will have a user-friendly interface that will allow users to easily interact with the recommendation system.
5. Admin functionality: The software will have an admin functionality that allows the admin to view, update, add, and delete movies and users in the system.

This project scope will serve as the basis for the development of the Movie Recommender System and will guide the development team in their efforts to build a high-quality, user-friendly, and effective movie recommendation system.

# Overall Description

## Product Perspective

The Movie Recommender System is a web-based application that provides personalized movie recommendations to users based on their movie choice. The software utilizes Pearson's Correlation to generate recommendations, ensuring that the recommendations are relevant and accurate. The software is designed to be accessible to users through a web browser, making it easily accessible from anywhere, at any time.

The software is designed to be user-friendly and intuitive, with a simple and easy-to-use interface. This will allow users of all technical levels to easily navigate the software and access the movie recommendations they are looking for. The software has been carefully developed with a focus on performance and efficiency, ensuring that users can receive their recommendations quickly and without any hassle. The recommendation engine uses Pearson's Correlation, which helps to provide accurate and relevant movie recommendations to users. The software is also designed to be flexible and customizable, allowing users to easily adjust their movie recommendations based on their movie choice

The Movie Recommender System is intended for a wide range of users, from casual movie fans to film aficionados. The software will be scalable, allowing it to grow and expand as the needs of its users evolve. This will ensure that the software continues to meet the needs of its users as the movie industry and the market continue to change.

The software will be developed to integrate with other systems and applications, providing a seamless experience for users. This will ensure that the Movie Recommender System can be used in conjunction with other systems and applications, providing users with a comprehensive and convenient movie recommendation solution.

## Tools and Technologies

The software was developed using the following tools and technologies for the frontend, backend, and development environments respectively:

Frontend:

* React: a JavaScript library for building user interfaces
* Bootstrap: a front-end framework for designing responsive and mobile-first websites
* Axios: a popular JavaScript library used for making HTTP requests

Backend:

* Java: a general-purpose programming language used for developing the software
* Spring Boot: a Java-based framework for building standalone and production-ready applications
* Oracle Database: a widely used relational database management system (RDBMS) used for storing and retrieving data

Development Environments:

* IntelliJ IDEA: a popular Java integrated development environment (IDE)
* Visual Studio Code (VS Code): a multi-language source code editor developed by Microsoft
* Postman: a powerful collaboration platform for API development
* Oracle SQL Developer: a free, integrated development environment for working with SQL and PL/SQL

These tools and technologies were chosen for their stability, scalability, and ease of use, and were instrumental in building a robust and reliable software system.

# Software Features

## Login Page

This feature allows users to log in to this application.

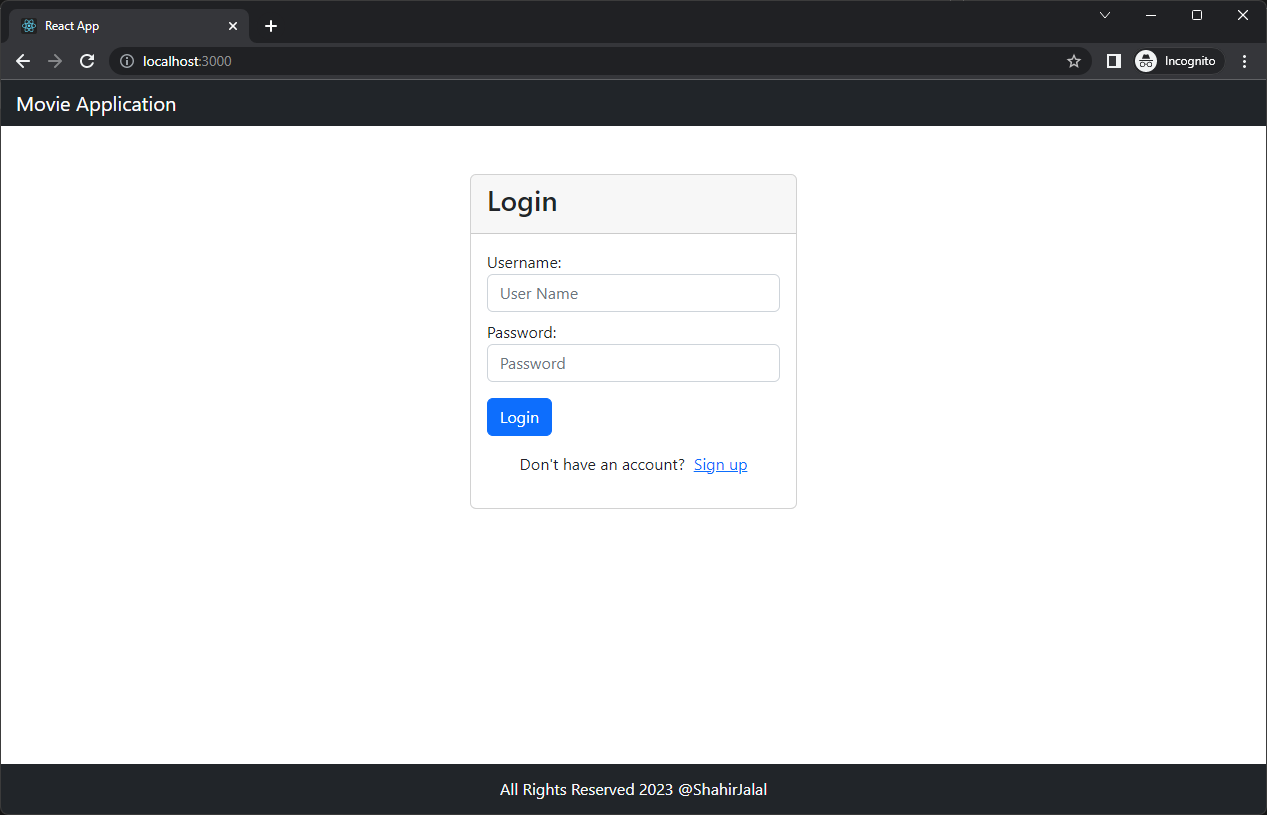


Figure 1. Login Page

## Registration Page

This feature allows users to register for an account to this application.

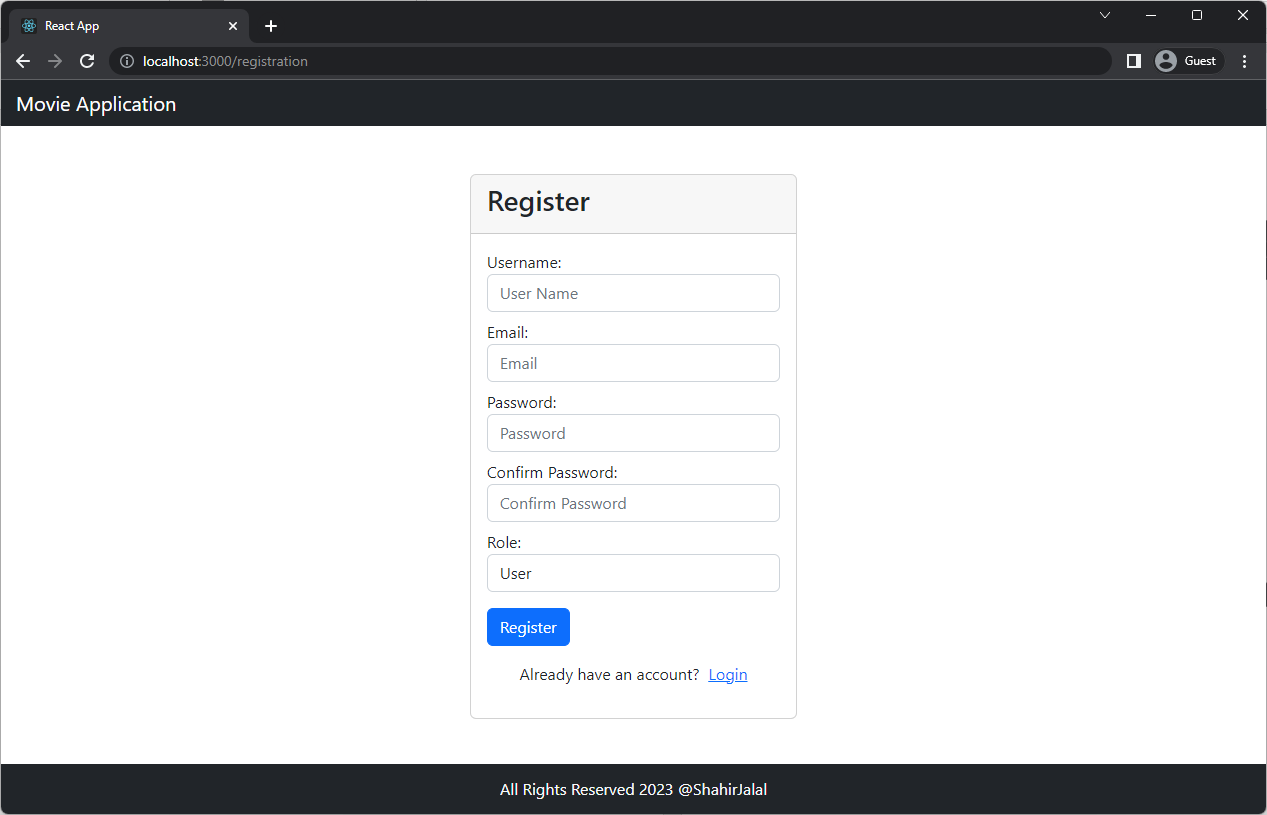


Figure 2. Registration Page

## List of Users Page

This feature allows admin to view all the users registered to this application.

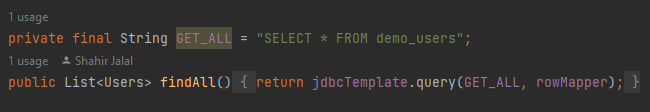


Figure 3: findAll Method in Repository

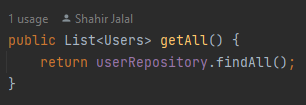


Figure 4. getAll Method in Service



Figure 5. getAll Method in Controller

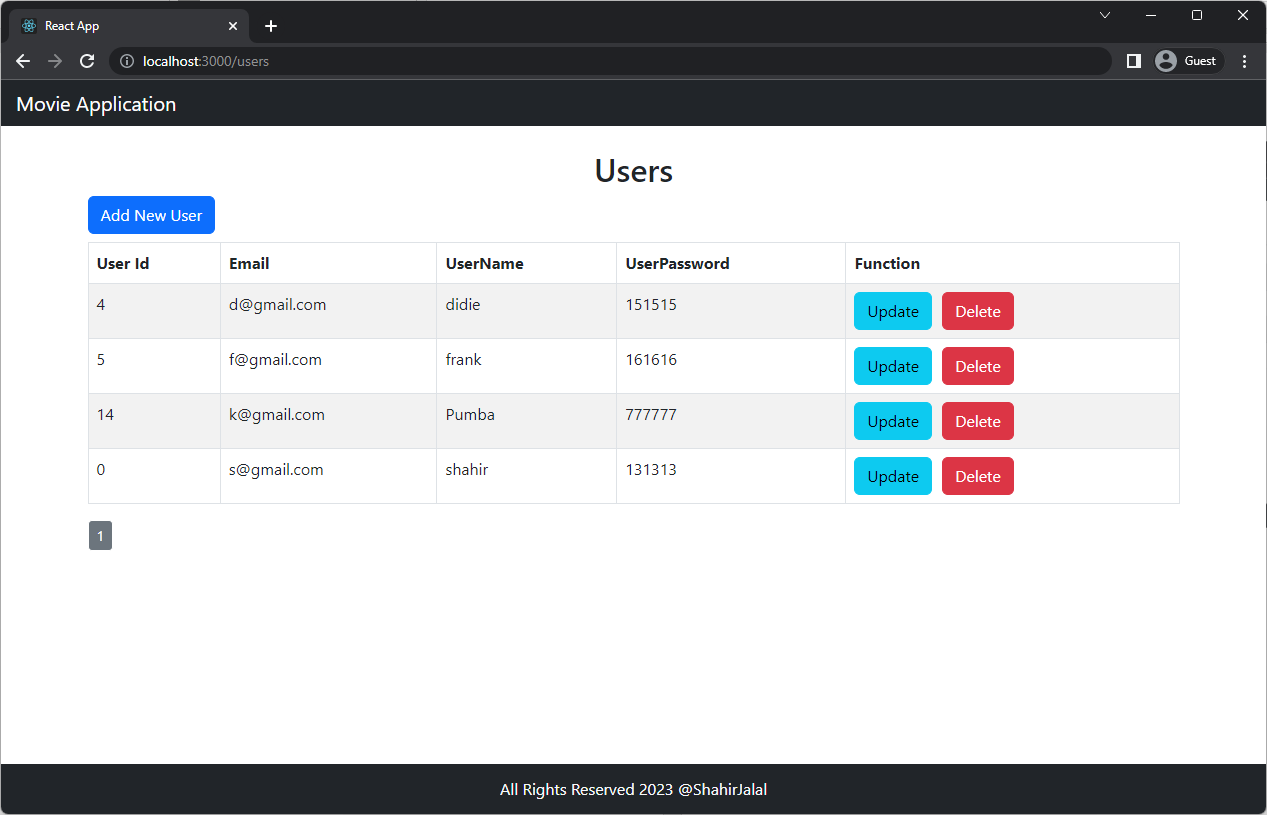


Figure 6. Users List Page

## Add User Page

This feature allows admin to add new users to this application.

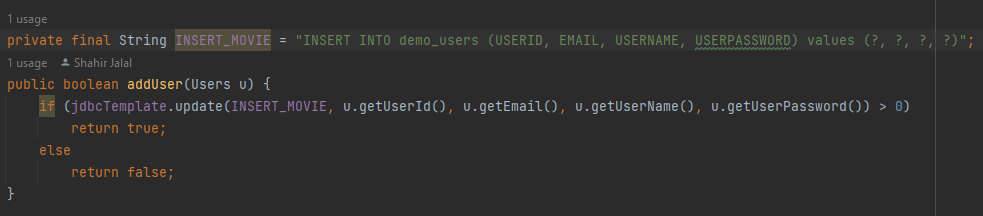


Figure 7. addUser Method in Repository

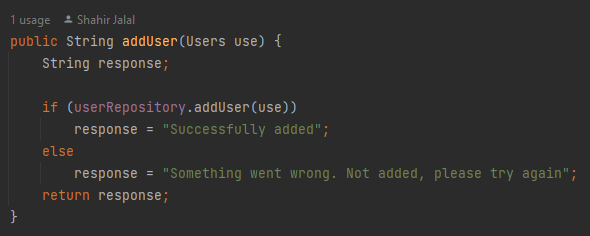


Figure 8. addUser Method in Service

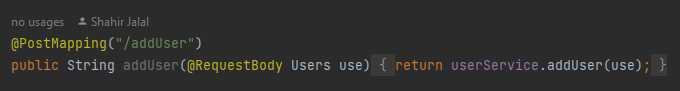


Figure 9. addUser Method in Controller

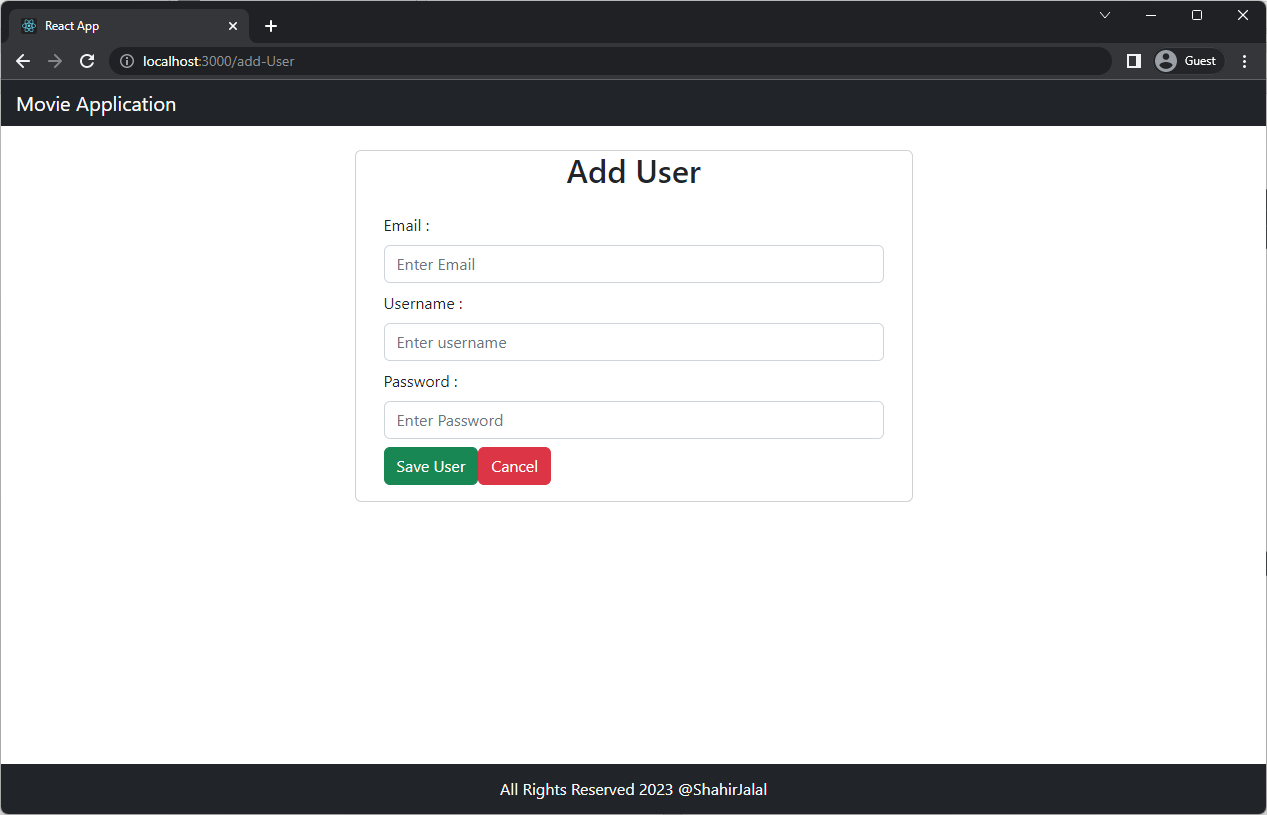


Figure 10. Add User Page

### Add User Flowchart

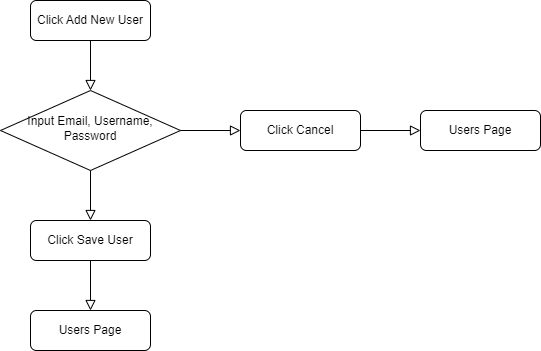


Figure 11. Add New User Flowchart

## Delete User Function

The "Delete User" function in the Movie Recommender System is an administrative tool that allows administrators to remove a user's account and associated data. This function is triggered by clicking the "Delete User" button and operates based on the user's unique identifier, the userId. When the button is clicked, the system checks the userId to ensure it is valid and then removes all records, including viewing history and preferences, associated with that user. The data is permanently deleted and cannot be recovered. This function is important for maintaining accurate and up-to-date user data in the system.

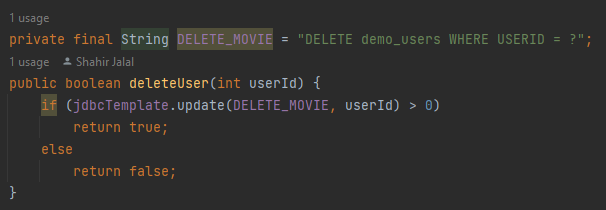


Figure 12. deleteUser Method in Repository

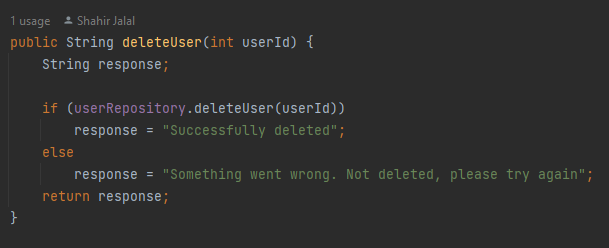


Figure 13. deleteUser Method in Service

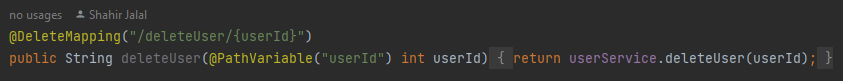


Figure 14. deleteUser Method in Controller

## Update User Page

This feature allows admin to update data for all the users registered to this application.

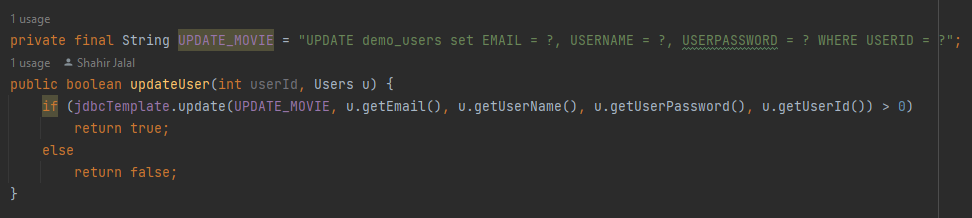


Figure 15. updateUser Method in Repository

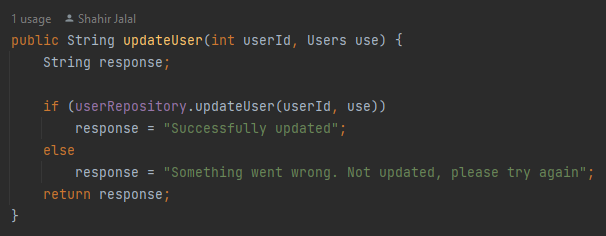


Figure 16. updateUser Method in Service

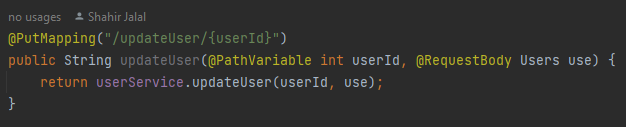


Figure 17. updateUser Method in Controller

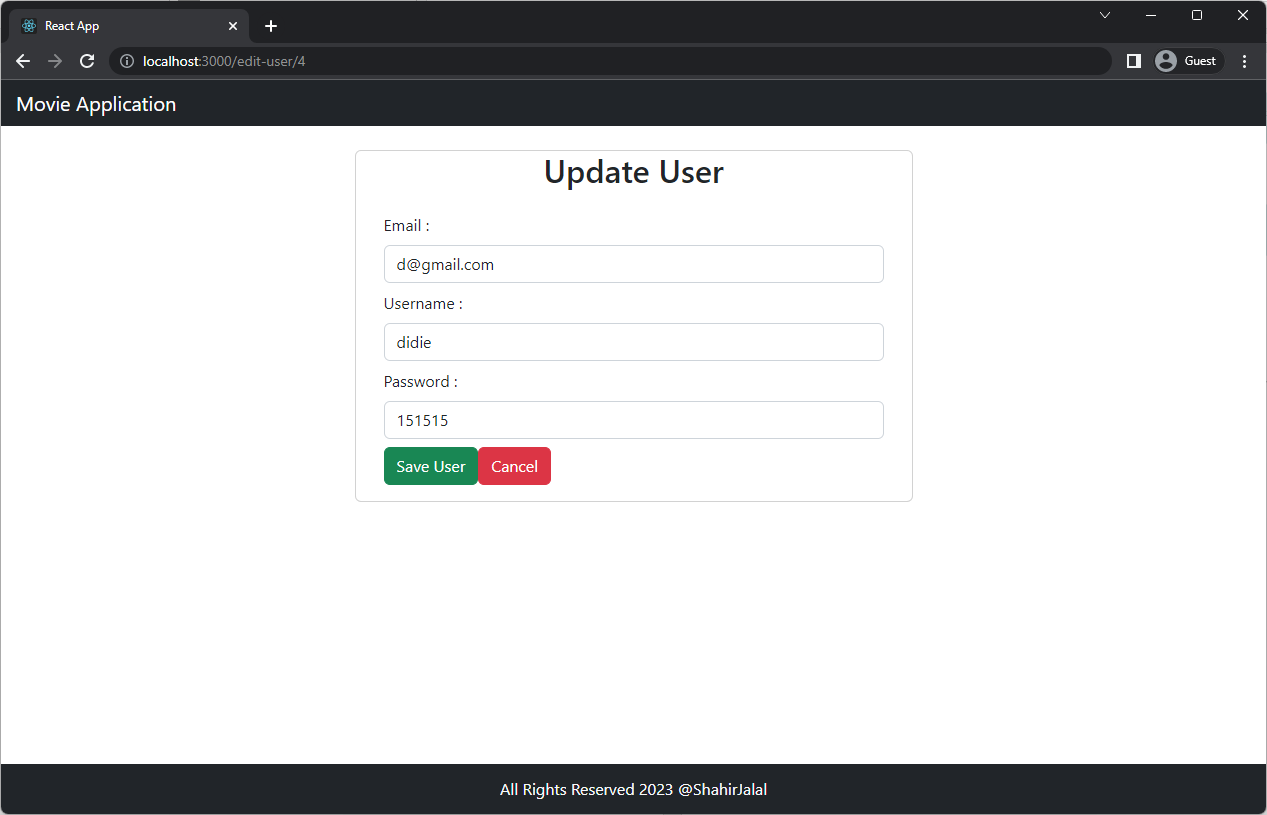


Figure 18. Update User Page

### Update User Flowchart

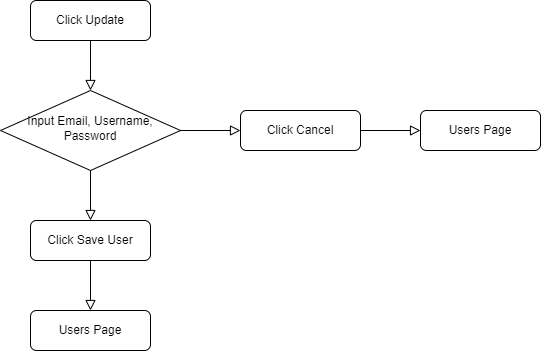


Figure 19. Update User Flowchart

## List of Movies Page

This feature allows users to view all the movies in this application.

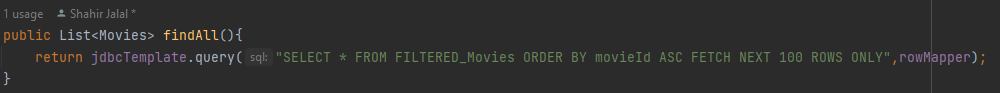


Figure 20. findAll Method in Repository



Figure 21. getAll Method in Service

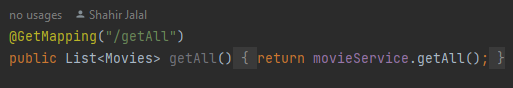


Figure 22. getAll Method in Controller

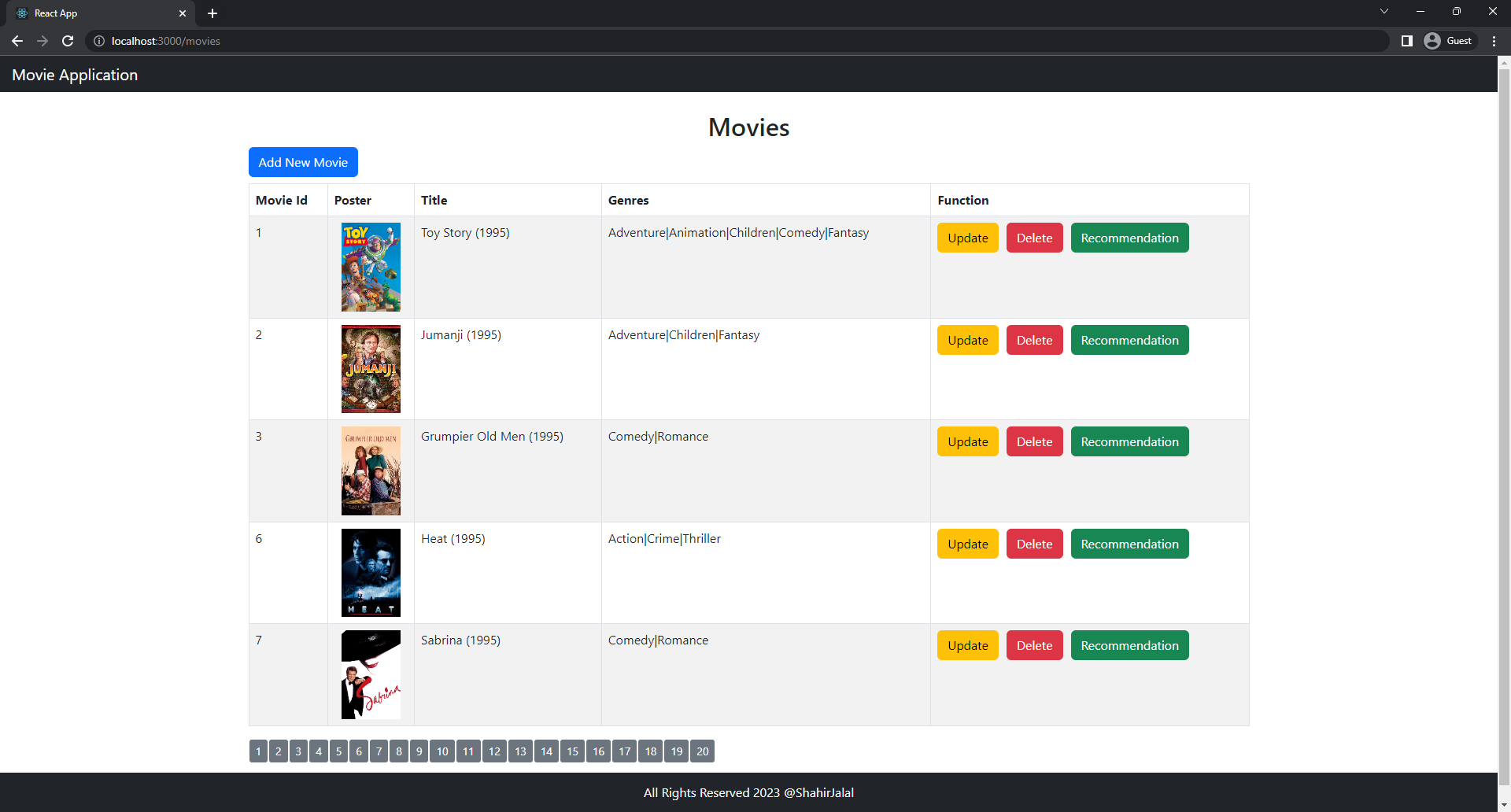


Figure 23. Movies Page

## Add Movie Page

This feature allows admin to add more movies to this application.

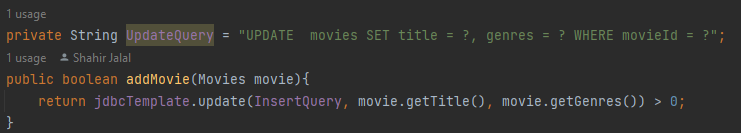


Figure 24. addMovie Method in Repository

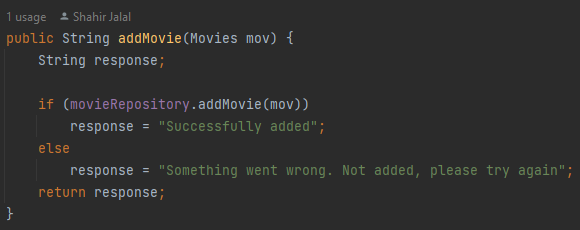


Figure 25. addMovie Method in Service

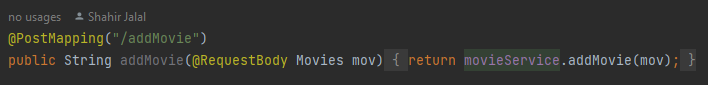


Figure 26. addMovie Method in Controller

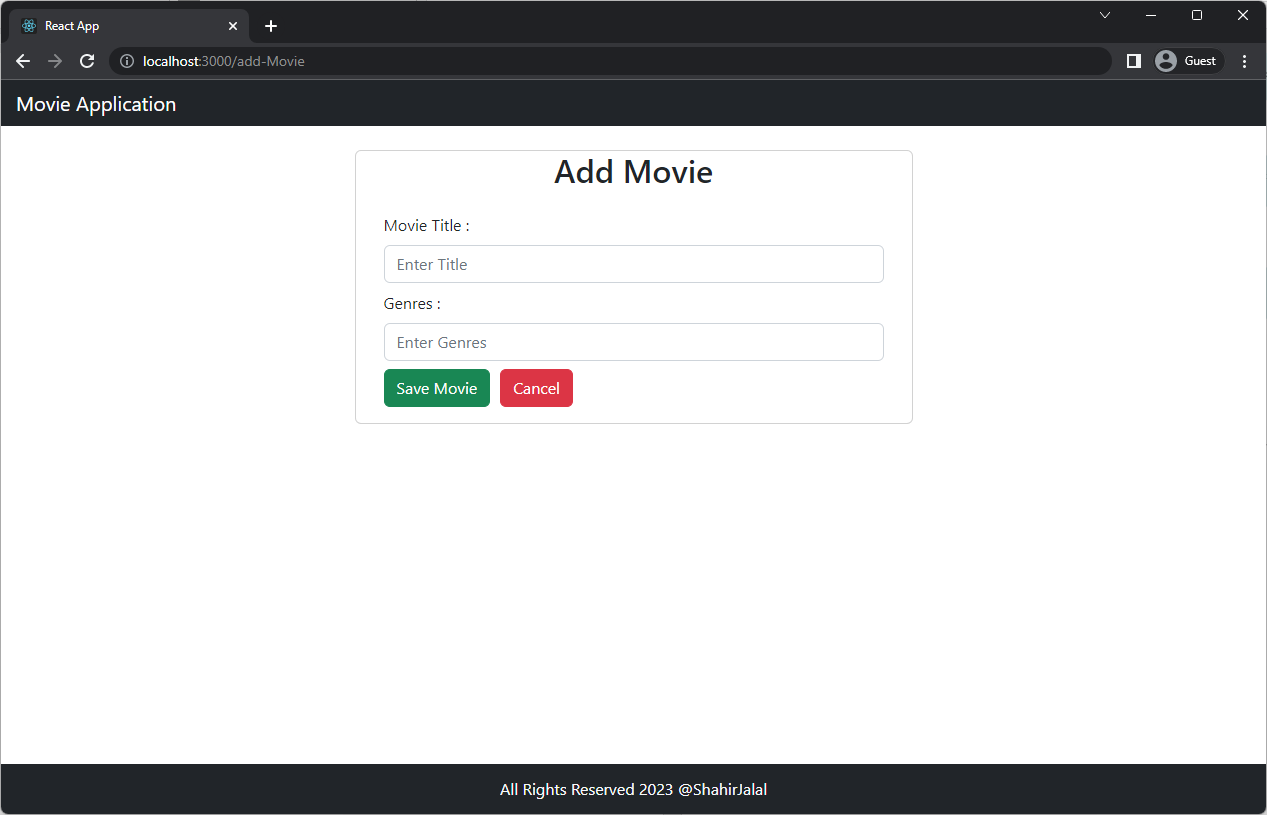


Figure 27. Add Movie Page

### Add Movie Flowchart

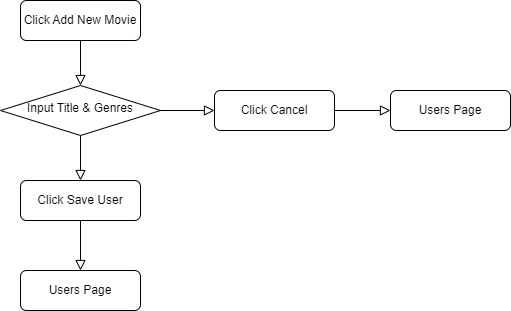


Figure 28. Add New Movie Flowchart

## Update Movie Page

This feature allows admin to update existing movies in this application.

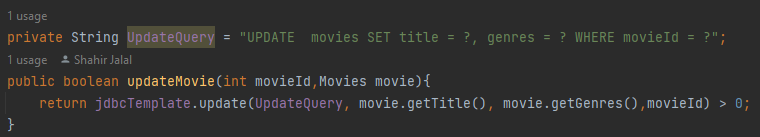


Figure 29. updateMovie Method in Repository

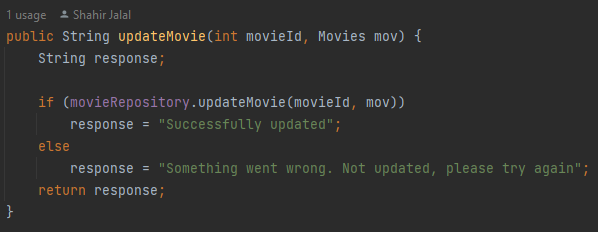


Figure 30. updateMovie Method in Service

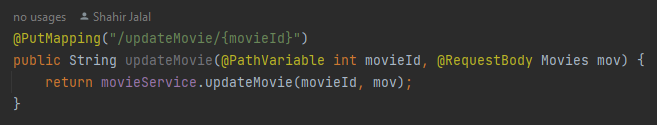


Figure 31. updateMovie Method in Controller

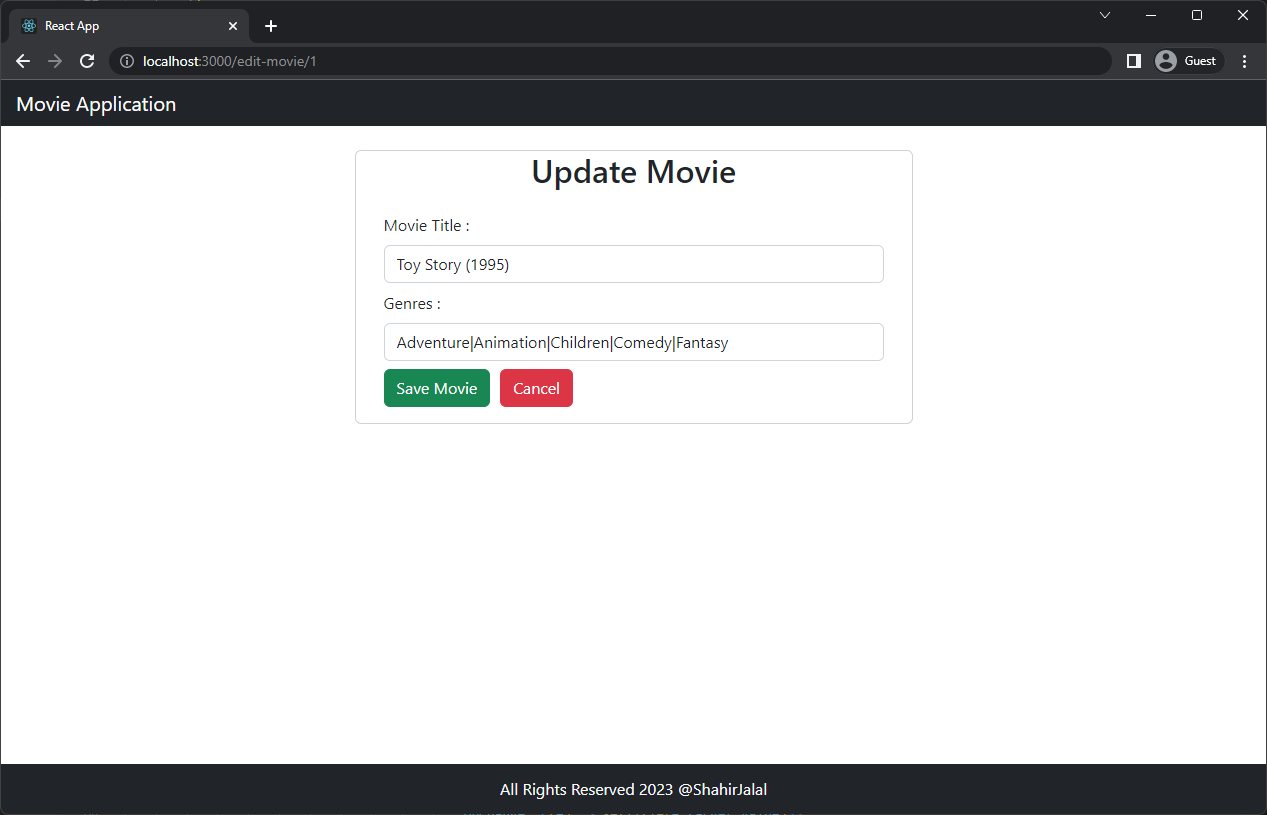


Figure 32. Update Movie Page

### Update Movie Flowchart

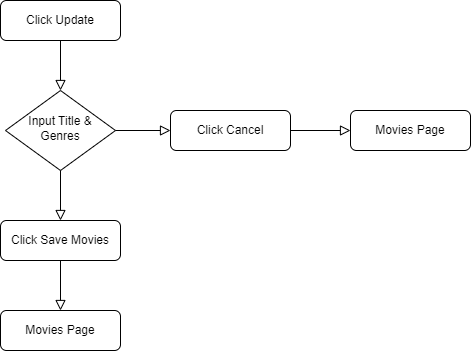


Figure 33. Update Movie Flowchart

## Delete Movie Function

The "Delete Movie" function in the Movie Recommender System enables administrators to remove a movie from the system by clicking the "Delete Movie" button. This function operates based on the movie's unique identifier, the movieId. The system checks the movieId, removes all records associated with the movie, and permanently deletes the data. The "Delete Movie" function helps maintain accurate and up-to-date data in the system by enabling administrators to remove movies.

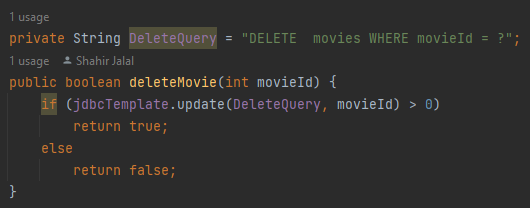


Figure 34. deleteMovie Method in Repository

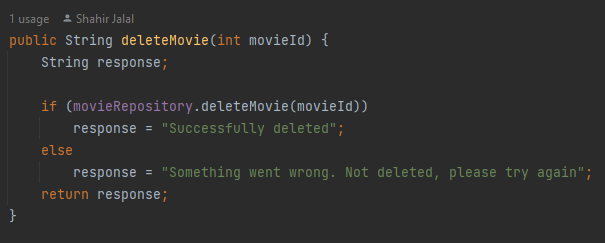


Figure 35. deleteMovie Method in Service

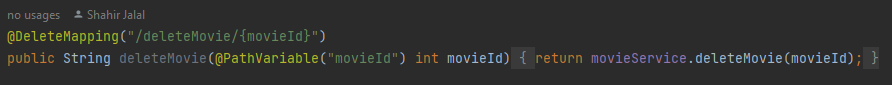


Figure 36. deleteMovie Method in Controller

## List of Recommended Movies Page

This feature allows users to view the most similar movies to the one they clicked on this application.



Figure 37. getSimilarMovies Method in Repository

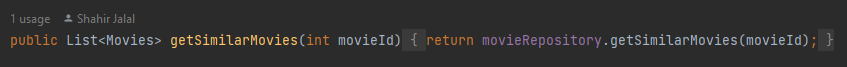


Figure 38. getSimilarMovies Method in Service

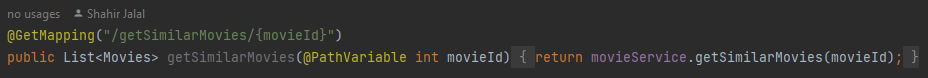


Figure 39. getSimilarMovies Method in Controller

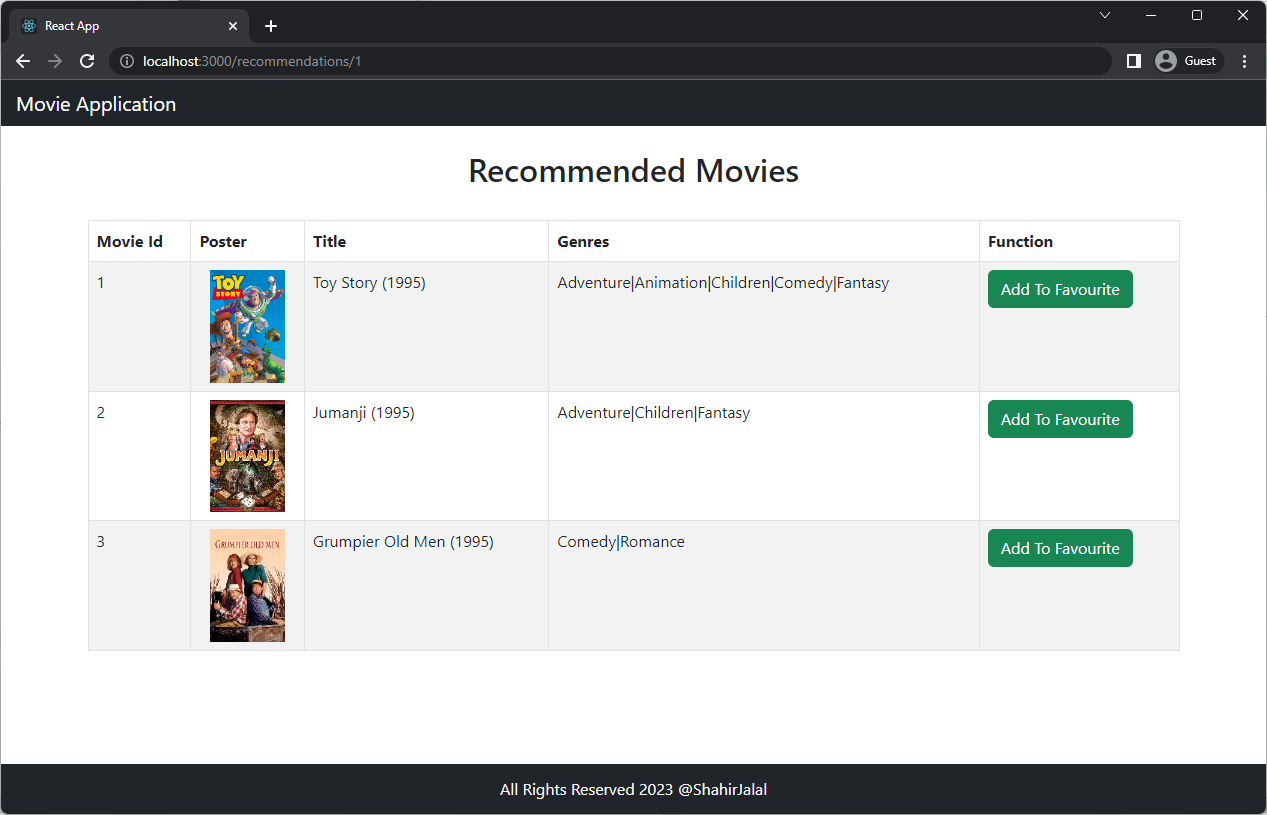


Figure 40. Recommended Movies Page

# Entity Relationship Diagram (ERD)

An Entity Relationship Diagram (ERD) is a visual representation of the relationships between entities in a database. In the context of the Movie Recommender System, the ERD provides an overview of the relationships between different tables in the system.

The ERD for the Movie Recommender System contains five main tables: pearson\_correlation, ratings, filtered\_movies\_medium\_tmdbId, links, and demo\_users.

The pearson\_correlation table contains movie IDs and a unique column named matched\_id. This table is used to store the similarity scores between movies, calculated using Pearson's Correlation, and is crucial to the movie recommendation process.

The ratings table contains userId, rating, and movieId columns, and is used to store the ratings given by users for individual movies.

The movies table contains movieId, title, and genres columns and is used to store information about each movie in the system.

The filtered\_movies\_medium\_tmdbId table contains movieId and tmdbId columns and is used to associate each movie with a unique identifier from the TMDb database.

The demo\_users table contains userId, email, userName, userPassword and role columns, and is used to store information about each registered user in the system, including their unique identifier, login details, and viewing history.

The relationships between these tables are represented graphically in the ERD, providing a clear visual representation of the relationships between entities in the system and how data is organized and stored in the database.

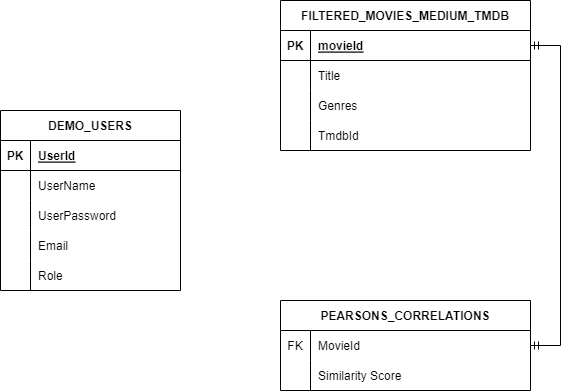


Figure 41. Entity Relationship Diagram

# Pearson’s Correlation

Pearson's Correlation, also known as Pearson's Product-Moment Correlation Coefficient, is a statistical method used to measure the linear relationship between two variables. It is one of the most widely used methods of measuring the strength and direction of the relationship between variables, and has been used in various fields of study including psychology, economics, and marketing.

The Pearson's Correlation coefficient is expressed as a value between -1 and 1, with a value of -1 indicating a perfect negative correlation, a value of 1 indicating a perfect positive correlation, and a value of 0 indicating no correlation. This value is used to quantify the strength and direction of the relationship between two variables, making it an extremely useful tool in understanding the relationship between variables.

In the context of the Movie Recommender System, Pearson's Correlation is used to measure the relationship between movies in a database. By analyzing this relationship, the software is able to determine the types of movies that are most likely to be of interest to the user based on their movie choice. This information is then used to generate personalized movie recommendations for the user.

Pearson's Correlation is an extremely powerful tool for data analysis, and its accuracy and reliability make it a valuable tool for the Movie Recommender System. The method takes into account multiple factors that may influence the relationship between variables, such as the number of data points, the distribution of data, and the presence of outliers. This results in a highly accurate and reliable measure of the relationship between variables, providing the foundation for accurate and relevant movie recommendations.

The implementation of Pearson's Correlation in the Movie Recommender System offers numerous benefits to users. By providing relevant and accurate movie recommendations, the software improves the user experience and increases user satisfaction. Furthermore, by continuously monitoring the relationship between a user's ratings and preferences, the software is able to adapt and evolve over time, ensuring that users receive the most relevant and up-to-date movie recommendations possible.

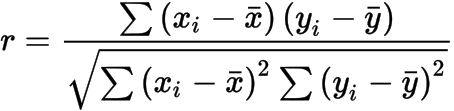
In conclusion, Pearson's Correlation is an essential component of the Movie Recommender System. Its accuracy and reliability make it a valuable tool in providing relevant and accurate movie recommendations to users. By utilizing Pearson's Correlation, the Movie Recommender System is able to improve the user experience and increase user satisfaction, making it a powerful tool for anyone looking to discover new and exciting movies.

## Generating Similarity Matrix

Making a similarity matrix using Pearson's Correlation is a process that involves several steps:

1. Data Collection: The first step is to collect the data. This data should consist of user ratings for different movies, with ratings ranging from 1 to 5. This data can be obtained from websites such as IMDb, TMDb, and Rotten Tomatoes.
2. Data Preparation: The next step is to prepare the data for analysis. This involves cleaning and pre-processing the data to ensure that it is in a suitable format for use in the similarity matrix.
3. Calculation of Pearson's Correlation: Once the data has been prepared, the next step is to calculate Pearson's Correlation between each pair of movies. The formula for Pearson's Correlation involves finding the covariance between the two variables and dividing this by the product of their standard deviations.

Equation 1: Pearson's Correlation Formula



Where,

|  |  |  |
| --- | --- | --- |
|  | = | correlation coefficient |
|  | = | values of the x-variable in a sample |
|  | = | mean of the values of the x-variable |
|  | = | values of the y-variable in a sample |
|  | = | mean of the values of the y-variable |

1. Creating the Similarity Matrix: With the Pearson's Correlation values calculated, the next step is to create the similarity matrix. This matrix is a square matrix with the number of rows and columns equal to the number of movies. Each cell in the matrix represents the Pearson's Correlation between two movies.
2. Normalization: Finally, the similarity matrix should be normalized so that the values fall within a range of -1 to 1. This normalization is important for ensuring that the values in the matrix are comparable and that the strongest similarities are easily recognizable.

By following these steps, a similarity matrix can be created using Pearson's Correlation, providing a useful tool for movie recommendations. The matrix can be used to identify the most similar movies based on user ratings, allowing for accurate and relevant movie recommendations to be generated for users.

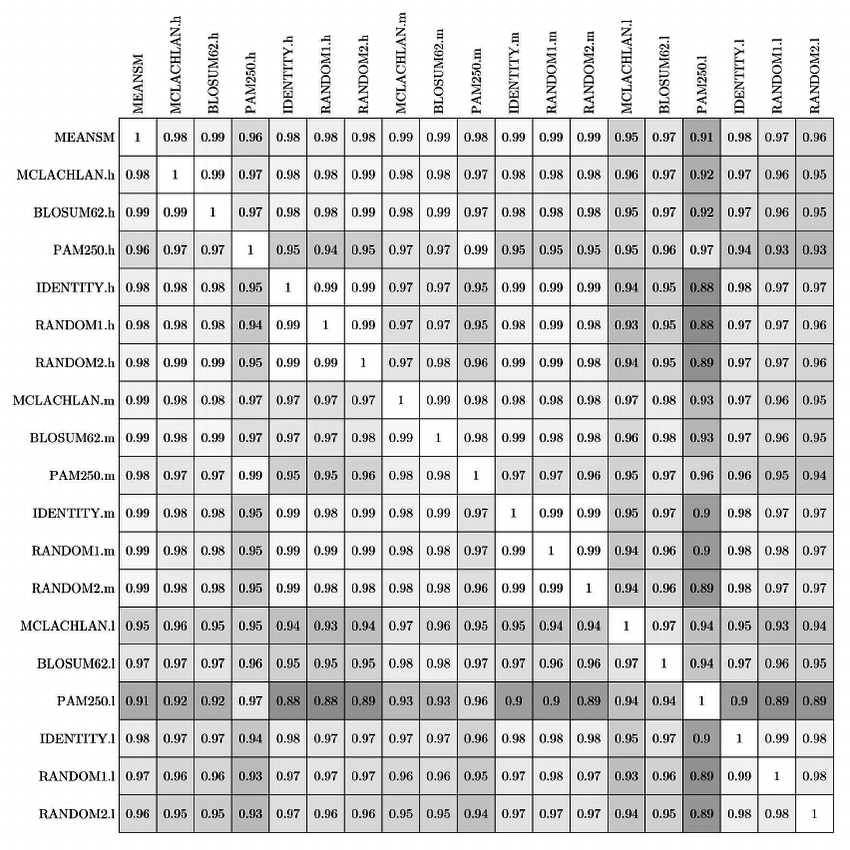


Figure 42. Example of Similarity Matrix

# References

1. <https://www.researchgate.net/figure/Pearson-correlation-coefficients-between-the-optimized-similarity-matrices-computed-with_fig5_46381440>
2. <https://en.wikipedia.org/wiki/Pearson_correlation_coefficient>