

Movie Recommendation System

Project By: Lakshay Gupta and Sashwat Rana

Acknowledgment

I would like to express my sincere gratitude to my guide, Mr. Sumit Kumar, for their invaluable guidance and constant support throughout the duration of this project. I am also thankful to The NorthCap University for providing the necessary resources and environment to complete this project successfully. Lastly, I would like to thank my peers and family for their encouragement.

Abstract

This project report outlines the development of a Movie Recommendation System using Spring Boot and MongoDB. The system is designed to provide personalized movie suggestions to users based on their interaction history, preferences, and ratings. The goal is to enhance the user's experience by delivering tailored recommendations that match their tastes. The backend is powered by Java and Spring Boot, offering a scalable and secure structure, while MongoDB is used for flexible and efficient data storage. This report covers the design, development process, system architecture, key features, and potential future enhancements.

Introduction

With the massive amount of content available on streaming platforms today, users often face difficulty in choosing what to watch.

Recommendation systems address this challenge by analyzing user behavior and providing personalized suggestions. This project aims to build such a system using Java-based technologies. The developed Movie Recommendation System offers users a seamless way to explore, rate, and receive suggestions for movies they are likely to enjoy.

Objective

The primary objective of this project is to:

- Develop a movie recommendation web application using Java and Spring Boot.
- Implement user registration, authentication, and session management.
- Allow users to browse, rate, and review movies.
- Generate personalized movie recommendations based on user data and preferences.
- Ensure a responsive and user-friendly interface for interaction.

Literature Review

Recommendation systems are commonly used in e-commerce, streaming services, and social platforms. There are three primary types of recommendation systems:

- **Content-based filtering:** Recommends items similar to those the user liked in the past.
- **Collaborative filtering:** Suggests items liked by similar users.
- **Hybrid systems:** Combine both approaches.

In this project, we adopt a custom logic to recommend movies based on user ratings and past interactions, ensuring high relevance and personalization without relying on complex algorithms like KNN or matrix factorization.

Technology Stack

- **Backend Framework:** Spring Boot (Java-based web framework)
- **Programming Language:** Java
- **Database:** MongoDB (NoSQL database for scalability and flexibility)
- **Frontend:** HTML, CSS, JavaScript (for UI rendering)
- **Tools & IDEs:**
 - Spring Tool Suite / VS Code
 - Maven (Dependency management)
 - Postman (API testing)
 - GitHub (Version control)
 - MongoDB (Database)

System Architecture

1. User Interface Layer

- Handles all user interactions.
- Allows users to sign up, log in, rate movies, and view recommendations.

2. Backend Layer (Spring Boot)

- Manages business logic, API endpoints, and security.
- Processes and stores user data and movie ratings.

3. Database Layer (MongoDB)

- Stores movie details, user profiles, reviews, and preferences.
- Enables quick retrieval and flexible querying of user data.

Introduction

This project is a Movie Recommendation System developed using Spring Boot. The system is designed to suggest movies to users based on their preferences. By leveraging user data and implementing a machine learning algorithm, the system provides personalized movie recommendations to enhance user experience.

Introduction

This project is a Movie Recommendation System developed using Spring Boot. The system is designed to suggest movies to users based on their preferences. By leveraging user data and implementing a machine learning algorithm, the system provides personalized movie recommendations to enhance user experience.

Functional Modules

1. User Authentication Module

- Allows users to register and securely log in.
- Passwords are encrypted before storage.

2. Movie Listing Module

- Displays a list of movies available in the database.
- Users can browse and view movie details.

3. Rating and Review Module

- Users can rate movies on a scale (e.g., 1–5 stars).
- Users can also write textual reviews.

4. Recommendation Module

- Suggests movies to users based on their preferences and ratings.
- Leverages user interaction data to generate recommendations.

Implementation Details

- **Spring Boot APIs:** Developed RESTful APIs for user login, movie listing, rating, and fetching recommendations.
- **MongoDB Integration:** Used Spring Data MongoDB for data access operations.
- **Frontend Interface:** A responsive UI built using HTML/CSS for basic interaction and testing.
- **Error Handling:** Implemented global exception handling and validation for secure input.

Testing and Evaluation

Testing was conducted at various levels:

- **Unit Testing:** Each service and controller was tested independently.
- **Integration Testing:** Verified the interaction between APIs and the database.
- **UI Testing:** Manual testing through Postman and the web interface to check overall flow.
- **Performance Evaluation:** Tested API response time and database query efficiency.

Conclusion

The Movie Recommendation System successfully demonstrates how Java and Spring Boot can be used to build a dynamic, user-centric application. With a simple yet effective recommendation logic, it enhances user engagement and shows the potential of personalized experiences in web applications. The project lays a strong foundation for future improvements involving more complex algorithms and scalable architecture.

References

- Spring Boot Documentation – <https://spring.io/projects/spring-boot>
- MongoDB Documentation – <https://www.mongodb.com/docs/>
- JavaDocs – <https://docs.oracle.com/en/java/>
- Stack Overflow – <https://stackoverflow.com/>
- YouTube tutorials on Spring Boot and MongoDB integration