***Synopsis: Movie Recommendation Platform***

**Introduction to the Project**

The Movie Recommendation Platform is a microservices-based application designed to provide personalized movie suggestions. It manages user profiles, movie catalogs, and user reviews/ratings, all integrated into a cohesive system. The platform leverages a rule-based recommendation approach eschewing complex machine learning models to combine user preferences, trending movies, and collaborative filtering techniques. Utilizing modern cloud-native technologies, the application is built to be scalable, flexible, and easy to maintain, offering movie enthusiasts a tailored viewing experience.

**Problem Statement**

Current movie recommendation systems often rely on complex machine learning models that require extensive data, computational resources, and continuous tuning. This complexity can hinder scalability, increase development time, and raise maintenance costs. Additionally, many platforms fail to offer a truly personalized experience due to rigid architectures. There is a need for a simpler, yet effective, solution that provides accurate, rule-based recommendations while ensuring high performance and ease of deployment.

**Objectives**

* Develop a scalable and modular movie recommendation system using microservices architecture.
* Implement distinct services to manage user profiles, movie catalogs, and user reviews/ratings using Spring Boot.
* Utilize rule-based techniques to generate personalized movie recommendations without the overhead of machine learning.
* Integrate with public movie APIs to continuously update and enrich the movie catalog.
* Employ Eureka Server for dynamic service discovery and an API Gateway for unified request routing and enhanced security.
* Containerize services using Docker and orchestrate them with Kubernetes for seamless deployment on cloud infrastructures.

**Scope of the Project**

* **Microservices Implementation:**
  + **User Profile Service:** Manages user registration, authentication, preferences (such as favorite genres, actors, directors), and watch history.
  + **Movie Catalog Service:** Maintains a comprehensive repository of movies by fetching and storing details from external public APIs.
  + **Review & Ratings Service:** Handles user reviews and ratings, providing data to refine recommendations and identify trending movies.
  + **Recommendation Service:** Generates personalized movie suggestions based on rule-based logic, incorporating inputs from user preferences, reviews, and trending data.
* **System Architecture:**
  + The services interact via RESTful APIs, registered dynamically with Eureka Server.
  + An API Gateway consolidates incoming requests, ensuring security and load balancing.
* **Deployment:**
  + Services are containerized with Docker and deployed using Kubernetes on cloud platforms, ensuring high availability and scalability.
* **Integration:**
  + The project integrates with external movie APIs (e.g., TMDB) to provide up-to-date movie details.

**Technologies Used**

* **Backend:** Spring Boot and Spring Cloud (including Eureka Server and API Gateway) for building and managing microservices.
* **Databases:** MySQL/PostgreSQL for structured data storage; optionally MongoDB for flexible review data storage.
* **Service Discovery:** Eureka Server to enable dynamic registration and lookup of microservices.
* **Containerization & Orchestration:** Docker for containerizing applications and Kubernetes for deployment, scaling, and management.
* **Cloud Deployment:** Deployment on cloud platforms such as AWS, GCP, or Azure to ensure reliability and scalability.
* **External Integration:** Public movie APIs (such as TMDB) to source updated movie information.

**Expected Outcome**

* A robust and user-friendly movie recommendation system that delivers personalized suggestions without the need for complex machine learning models.
* Enhanced user engagement driven by accurate, rule-based recommendations tailored to individual preferences and current trends.
* A scalable, resilient, and modular microservices architecture that simplifies future enhancements and maintenance.
* Seamless integration and deployment in a cloud environment, ensuring high availability, reduced downtime, and efficient load management.

**Github Link**

<https://github.com/romxai/ma-project>