**Dự án MCS về web xem phim giống move9k**

* CN:

+ FE: React js, yup, react hook form,…

+ BE: MCS spring boot

* Các chức năng:

+ Sign up/in, Quên MK

+ Liệt kê danh sách phim đang chiếu, từng bộ phim sẽ có có bài viết về phim đó + Đánh giá

+ Liệt kê danh sách các phim sắp chiếu

**1.1. Bảng Người Dùng (users)**

Lưu thông tin người dùng (khách hàng, admin).

| **Column** | **Data Type** | **Description** |
| --- | --- | --- |
| id | INT (PK) | Mã người dùng (tự tăng) |
| full\_name | VARCHAR(100) | Họ và tên |
| email | VARCHAR(100) | Email (duy nhất) |
| password | VARCHAR(255) | Mật khẩu (mã hóa) |
| phone | VARCHAR(15) | Số điện thoại |
| role | ENUM('user', 'admin') | Vai trò người dùng |
| created\_at | TIMESTAMP | Ngày tạo tài khoản |

**1.2. Bảng Phim (movies)**

Lưu thông tin về các bộ phim.

| **Column** | **Data Type** | **Description** |
| --- | --- | --- |
| id | INT (PK) | Mã phim (tự tăng) |
| title | VARCHAR(255) | Tên phim |
| description | TEXT | Mô tả phim |
| genre | VARCHAR(100) | Thể loại |
| duration | INT | Thời gian (phút) |
| director | VARCHAR(255) | Đạo diễn |
| actors | TEXT | Diễn viên chính |
| release\_date | DATE | Ngày phát hành |
| image\_url | VARCHAR(255) | Ảnh poster phim |
| trailer\_url | VARCHAR(255) | URL trailer |
| created\_at | TIMESTAMP | Ngày thêm vào hệ thống |

**1.3. Bảng Rạp (cinemas)**

Lưu thông tin về các rạp chiếu phim.

| **Column** | **Data Type** | **Description** |
| --- | --- | --- |
| id | INT (PK) | Mã rạp (tự tăng) |
| name | VARCHAR(255) | Tên rạp |
| address | TEXT | Địa chỉ rạp |
| city | VARCHAR(100) | Thành phố |
| created\_at | TIMESTAMP | Ngày tạo rạp |

**1.4. Bảng Phòng Chiếu (rooms)**

Lưu thông tin các phòng chiếu trong các rạp.

| **Column** | **Data Type** | **Description** |
| --- | --- | --- |
| id | INT (PK) | Mã phòng chiếu (tự tăng) |
| cinema\_id | INT (FK) | Mã rạp |
| name | VARCHAR(50) | Tên phòng chiếu |
| capacity | INT | Sức chứa phòng |

**1.5. Bảng Ghế (seats)**

Lưu thông tin về các ghế trong phòng chiếu.

| **Column** | **Data Type** | **Description** |
| --- | --- | --- |
| id | INT (PK) | Mã ghế (tự tăng) |
| room\_id | INT (FK) | Phòng chiếu |
| seat\_number | VARCHAR(10) | Số ghế (A1, B1...) |
| type | ENUM('standard', 'vip') | Loại ghế |

**1.6. Bảng Suất Chiếu (showtimes)**

Lưu thông tin về các suất chiếu của từng phim trong từng phòng.

| **Column** | **Data Type** | **Description** |
| --- | --- | --- |
| id | INT (PK) | Mã suất chiếu (tự tăng) |
| movie\_id | INT (FK) | Mã phim |
| room\_id | INT (FK) | Phòng chiếu |
| start\_time | DATETIME | Thời gian chiếu |
| price | DECIMAL(10,2) | Giá vé |

**1.7. Bảng Đặt Vé (bookings)**

Lưu thông tin về các đơn đặt vé của người dùng.

| **Column** | **Data Type** | **Description** |
| --- | --- | --- |
| id | INT (PK) | Mã đơn đặt vé |
| user\_id | INT (FK) | Mã người dùng |
| showtime\_id | INT (FK) | Mã suất chiếu |
| total\_price | DECIMAL(10,2) | Tổng tiền đặt vé |
| status | ENUM('pending', 'paid', 'cancelled') | Trạng thái đơn |
| created\_at | TIMESTAMP | Ngày đặt vé |

**1.8. Bảng Chi Tiết Đặt Vé (booking\_details)**

Lưu thông tin các ghế được chọn trong một đơn đặt vé.

| **Column** | **Data Type** | **Description** |
| --- | --- | --- |
| id | INT (PK) | Mã chi tiết vé |
| booking\_id | INT (FK) | Mã đơn đặt vé |
| seat\_id | INT (FK) | Mã ghế |

**1.9. Bảng Thanh Toán (payments)**

Lưu thông tin về thanh toán của các đơn đặt vé.

| **Column** | **Data Type** | **Description** |
| --- | --- | --- |
| id | INT (PK) | Mã thanh toán |
| booking\_id | INT (FK) | Mã đơn đặt vé |
| payment\_method | ENUM('vnpay', 'momo', 'cash') | Phương thức thanh toán |
| status | ENUM('success', 'failed') | Trạng thái giao dịch |
| created\_at | TIMESTAMP | Ngày thanh toán |

**1.10. Bảng Sản Phẩm (products)**

Lưu thông tin về các sản phẩm bắp nước, đồ ăn, nước uống.

| **Column** | **Data Type** | **Description** |
| --- | --- | --- |
| id | INT (PK) | Mã sản phẩm |
| name | VARCHAR(255) | Tên sản phẩm |
| description | TEXT | Mô tả sản phẩm |
| price | DECIMAL(10,2) | Giá sản phẩm |
| image\_url | VARCHAR(255) | Hình ảnh sản phẩm |
| created\_at | TIMESTAMP | Ngày thêm sản phẩm |

**1.11. Bảng Đơn Hàng Bắp Nước (order\_items)**

Lưu thông tin các sản phẩm được mua trong mỗi đơn đặt vé.

| **Column** | **Data Type** | **Description** |
| --- | --- | --- |
| id | INT (PK) | Mã sản phẩm trong đơn đặt |
| booking\_id | INT (FK) | Mã đơn đặt vé |
| product\_id | INT (FK) | Mã sản phẩm |
| quantity | INT | Số lượng sản phẩm |
| total\_price | DECIMAL(10,2) | Tổng tiền cho sản phẩm |

**1.12. Bảng Đánh Giá (reviews)**

Lưu thông tin về các đánh giá của người dùng về phim.

| **Column** | **Data Type** | **Description** |
| --- | --- | --- |
| id | INT (PK) | Mã đánh giá |
| user\_id | INT (FK) | Người đánh giá |
| movie\_id | INT (FK) | Phim được đánh giá |
| rating | INT CHECK(rating BETWEEN 1 AND 5) | Số sao (1-5) |
| comment | TEXT | Nội dung bình luận |
| created\_at | TIMESTAMP | Ngày đánh giá |

| **Microservice** | **Port** |
| --- | --- |
| **API Gateway** | 8080 |
| **User Service** PostgreSQL | 8081 |
| **Movie Service** PostgreSQL | 8082 |
| **Cinema Service** MySQL | 8083 |
| **Showtime Service** MySQL | 8084 |
| **Booking Service** PostgreSQL | 8085 |
| **Payment Service** PostgreSQL | 8086 |
| **Product Service** MySQL | 8087 |
| **Notification Service Redis/ Mongo** | 8088 |
| **Eureka Server** (Service Discovery - nếu dùng) | 8761 |
| **Config Server** (Spring Cloud Config - nếu dùng) | 8888 |
| **Message Broker** (Kafka, RabbitMQ) | 9092 (Kafka) / 5672 (RabbitMQ) |
| **Database (MySQL, PostgreSQL)** | 3306 (MySQL) / 5432 (PostgreSQL) |
| **Redis (caching)** | 6379 |

Dưới đây là cách chia hệ thống thành các microservices hợp lý cho dự án web bán vé xem phim của bạn:

## **1. User Service**

Quản lý thông tin người dùng và xác thực người dùng.

* **Bảng liên quan**: users
* **Chức năng chính**:
  + Đăng ký, đăng nhập, xác thực JWT
  + Quản lý thông tin cá nhân
  + Phân quyền người dùng (user, admin)

## **2. Movie Service**

Quản lý thông tin phim.

* **Bảng liên quan**: movies, reviews
* **Chức năng chính**:
  + CRUD phim (thêm, sửa, xóa, lấy danh sách phim)
  + Quản lý đánh giá phim (thêm, xóa, chỉnh sửa review)
  + Tìm kiếm phim theo tên, thể loại, diễn viên, đạo diễn

## **3. Cinema Service**

Quản lý thông tin rạp, phòng chiếu, ghế.

* **Bảng liên quan**: cinemas, rooms, seats
* **Chức năng chính**:
  + Quản lý danh sách rạp chiếu
  + Quản lý danh sách phòng chiếu trong từng rạp
  + Quản lý danh sách ghế trong từng phòng chiếu

## **4. Showtime Service**

Quản lý suất chiếu của phim.

* **Bảng liên quan**: showtimes
* **Chức năng chính**:
  + Lấy danh sách suất chiếu của một bộ phim tại một rạp
  + Thêm, sửa, xóa suất chiếu
  + Kiểm tra suất chiếu có sẵn theo ngày, rạp, phim

## **5. Booking Service**

Xử lý đặt vé, giữ chỗ, quản lý đơn hàng.

* **Bảng liên quan**: bookings, booking\_details
* **Chức năng chính**:
  + Đặt vé
  + Xác nhận ghế đã được đặt
  + Quản lý đơn đặt vé (hủy, xác nhận)
  + Tính tổng tiền đặt vé

## **6. Payment Service**

Quản lý thanh toán.

* **Bảng liên quan**: payments
* **Chức năng chính**:
  + Xử lý thanh toán bằng VNPay, Momo, tiền mặt
  + Kiểm tra trạng thái thanh toán
  + Gửi thông báo khi thanh toán thành công

## **7. Product Service**

Quản lý sản phẩm bắp nước, đồ ăn trong rạp.

* **Bảng liên quan**: products, order\_items
* **Chức năng chính**:
  + CRUD sản phẩm
  + Quản lý đơn hàng bắp nước kèm vé xem phim
  + Tính tổng tiền sản phẩm

## **8. Notification Service**

Gửi thông báo về vé, thanh toán, lịch chiếu.

* **Bảng liên quan**: (Không cần bảng riêng, có thể lưu log nếu cần)
* **Chức năng chính**:
  + Gửi email/SMS xác nhận đặt vé, thanh toán thành công
  + Nhắc nhở người dùng về suất chiếu sắp đến
  + Gửi thông báo về phim mới, khuyến mãi

## **9. API Gateway**

* Cung cấp một điểm truy cập duy nhất cho client
* Định tuyến request đến từng service phù hợp
* Thực hiện xác thực JWT

### **Công nghệ gợi ý**

* **Spring Boot** để phát triển các service
* **Spring Cloud Gateway** để làm API Gateway
* **Spring Security + JWT** để xác thực người dùng
* **Kafka/RabbitMQ** để giao tiếp giữa các service (như thông báo, thanh toán)
* **MySQL/PostgreSQL** làm database chính
* **Redis** để caching dữ liệu (suất chiếu, danh sách phim)

Cách chia này giúp hệ thống linh hoạt, dễ mở rộng và duy trì. Bạn có thể triển khai từng service độc lập và scale riêng biệt khi cần. 🚀