

TRƯỜNG ĐẠI HỌC ĐẠI NAM
KHOA CÔNG NGHỆ THÔNG TIN



BÀI TẬP LỚN

**HỌC PHẦN: PHÁT TRIỂN PHẦN MỀM MÃ NGUỒN
MỞ**

**ĐỀ TÀI: XÂY DỰNG ỨNG DỤNG WEB QUẢN LÝ ĐẶT VÉ
ONLINE**

Giảng viên: ThS. Lê Tuấn Anh

Sinh viên thực hiện:

- 1. Nguyễn Thanh Tùng – 1771020729**
- 2. Đỗ Văn Tuyên - 1771020734**

Hà Nội, năm 2025

TRƯỜNG ĐẠI HỌC ĐẠI NAM
KHOA CÔNG NGHỆ THÔNG TIN



BÀI TẬP LỚN

**HỌC PHẦN: PHÁT TRIỂN PHẦN MỀM MÃ NGUỒN
MỞ**

**ĐỀ TÀI: XÂY DỰNG ỨNG DỤNG WEB QUẢN LÝ ĐẶT VÉ
ONLINE**

<i>TT</i>	<i>Mã sv</i>	<i>Họ và Tên</i>	<i>Ngày Sinh</i>	<i>Điểm</i>	
				<i>Bảng Số</i>	<i>Bảng Chữ</i>
<i>1</i>	<i>1771020729</i>	<i>Nguyễn Thanh Tùng</i>	<i>01/09/2004</i>		
<i>2</i>	<i>1771020734</i>	<i>Đỗ Văn Tuyên</i>	<i>10/06/2005</i>		

CÁN BỘ CHẤM THI 1

CÁN BỘ CHẤM THI 2

ThS. Lê Tuấn Anh

Hà Nội, năm 2025

LỜI NÓI ĐẦU

Trong kỷ nguyên số hóa, nhu cầu về các dịch vụ trực tuyến tiện lợi và nhanh chóng đã trở thành một tiêu chuẩn bắt buộc trong mọi lĩnh vực, đặc biệt là ngành giải trí và điện ảnh. Việc đặt vé truyền thống tại quầy vé không còn đáp ứng được tốc độ và tính tiện lợi mà khách hàng hiện đại mong đợi. Nhận thức rõ ràng về xu hướng này, đề tài "Xây dựng ứng dụng web quản lý đặt vé xem phim" được triển khai nhằm mục đích tối ưu hóa quy trình kinh doanh và nâng cao trải nghiệm khách hàng cho chuỗi rạp SCARLET CINEMA.

Mục tiêu chính của đề tài là xây dựng một hệ thống phần mềm toàn diện, mạnh mẽ, đáp ứng đồng thời hai nhu cầu:

Phía Khách hàng (Client): Cung cấp giao diện trực quan, cho phép người dùng dễ dàng xem thông tin phim, kiểm tra lịch suất chiếu, chọn ghế trên sơ đồ động và hoàn tất giao dịch thanh toán.

Phía Quản trị (Admin Panel): Cung cấp các công cụ quản lý chuyên sâu (CRUD) cho toàn bộ dữ liệu nghiệp vụ (Phim, Rạp, Phòng chiếu, Suất chiếu) và các chức năng báo cáo, thống kê hiệu suất kinh doanh.

Đề tài đã vận dụng các công nghệ hiện đại như PHP và MySQL (sử dụng PDO để đảm bảo an toàn truy vấn) cùng với framework Tailwind CSS để xây dựng một ứng dụng ổn định, bảo mật và có tính thẩm mỹ cao. Đặc biệt, các nghiệp vụ phức tạp của ngành điện ảnh như kiểm tra chồng lấn lịch chiếu và quản lý sơ đồ ghế động đã được nghiên cứu và triển khai thành công.

Qua quá trình thực hiện, chúng tôi không chỉ đạt được các mục tiêu kỹ thuật đã đề ra mà còn có cơ hội áp dụng kiến thức chuyên môn vào giải quyết một bài toán thực tế, từ đó nâng cao kinh nghiệm trong việc phân tích, thiết kế và triển khai một ứng dụng web hoàn chỉnh.

Chúng tôi xin chân thành cảm ơn... (Thêm lời cảm ơn giảng viên hướng dẫn, gia đình, bạn bè).

MỤC LỤC

CHƯƠNG 1: TỔNG QUA VỀ ĐỀ TÀI.....	9
1.1. Giới thiệu đề tài.....	9
1.1.1. Lý do chọn đề tài.....	9
1.1.2. Mục tiêu đề tài.....	9
1.2. Phạm vi đề tài.....	10
1.1.1. Đối tượng nghiên cứu.....	10
1.1.2. Phạm vi nghiên cứu.....	10
1.3. Dự kiến công việc.....	11
1.4. Công nghệ sử dụng.....	12
1.4.1. Cơ sở Lý thuyết & Công cụ Lập trình.....	12
1.4.2. Thư viện và Framework.....	13
CHƯƠNG 2 : KHẢO SÁT PHÂN TÍCH YÊU CẦU.....	14
2.1. Tìm hiểu nghiệp vụ.....	14
2.1.1. Mô tả nghiệp vụ thực tế của rạp chiếu phim.....	14
2.1.2. Nhu cầu cần phát triển ứng dụng web.....	15
2.2. Phân tích yêu cầu.....	16
2.3. Thiết kế DataBase.....	24
2.3.1. Cấu trúc Từng Bảng Trong Database.....	24
2.3.2. Sơ đồ quan hệ thực thể (ERD).....	28
Chương 3: Lập trình xây dựng và cài đặt hệ thống.....	29
3.1. Xây dựng database trên Mysql.....	29
3.1.1. Cấu trúc các bảng trên Mysql.....	29

3.1.2. Sơ đồ quan hệ đã xây dựng	32
3.2. Tổ chức mã nguồn	34
3.2.1. Cấu trúc folder	34
3.2.2. Mô tả cấu trúc và chức năng file mã nguồn	35
3.3. Xây dựng các chức năng theo thiết kế	40
3.3.1. Giao diện website	40
3.3.2. Moudle mã nguồn	48
KẾT LUẬN	58
Link Source Project	60
DANH MỤC TÀI LIỆU THAM KHẢO	61

MỤC LỤC HÌNH ẢNH

Hình ảnh 01: Biểu đồ phân cấp	19
Hình ảnh 02: Use case Tổng quát	20
Hình 04: Usecase tổng quát khách hàng	21
Hình 04: Cấu trúc bảng Booking_items	29
Hình 05: Cấu trúc bảng Screens	29
Hình 06: Cấu trúc bảng Bookings	30
Hình 07: Cấu trúc bảng movies	30
Hình 08: Cấu trúc bảng shows	31
Hình 09: Cấu trúc bảng theaters	31
Hình 10: Cấu trúc bảng users	32
Hình 12: Cấu trúc foder	34
Hình 13: Cấu trúc foder	35
Hình 16: Giao diện đăng ký	41
Hình 17: Giao diện đăng nhập	41
Hình 18: Giao diện quên mật khẩu	42
Hình 19: Giao diện trang khách hàng	42
Hình 20: Giao diện đặt vé	43
Hình 21: Giao diện trang quản lý vé đã đặt	43
Hình 22: Giao diện dashboard(admin)	44
Hình 23: Giao diện quản lý phim(admin)	44
Hình 24: Giao diện Quản lý người dùng(admin)	45
Hình 25: Giao diện quản lý đơn đặt vé(admin)	46

Hình 26: Giao diện quản lý rạp phim(admin)	46
Hình 27: Giao diện quản lý phòng chiếu(admin)	46
Hình 28: Giao diện quản lý sơ đồ ghế(admin)	47
Hình 29: Giao diện quản lý suất chiếu(admin)	48
Hình 30: Giao diện Tìm kiếm(admin)	48

MỤC LỤC BẢNG

Bảng 01: Dự kiến công việc	12
Bảng 02: Cơ sở lý thuyết.....	13
Bảng 03: Thư viện và framework	13
Bảng 04: Nhu cầu phát triển	16
Bảng 05: Phân tích yêu cầu	17
Bảng 06: Đối tượng sử dụng	18
Bảng 07: Phân hệ khách hàng	21
Bảng 08: Phân hệ quản trị	23
Bảng 09: Bảng người dùng trong database	24
Bảng 10: Bảng rạp phim trong database	24
Bảng 11: Bảng Phim trong database	25
Bảng 12: Phân Loại Ghế trong database	25
Bảng 13: Phòng Chiếu trong database	25
Bảng 14: Bảng ghế ngồi trong database	26
Bảng 15: Bảng suất chiếu trong database	26
Bảng 16: Hóa đơn đặt vé	27
Bảng 17: Chi tiết hóa đơn trong database	27
Hình 03: Sơ đồ quan hệ thực thể URD	28
Bảng 18: Thư mục BTL_BOOK_MOVIE_TICKETS/asset (file tĩnh)	36
Bảng 19: Thư mục BTL_BOOK_MOVIE_TICKETS/function (xử lý database)	37
Bảng 20: Thư mục BTL_BOOK_MOVIE_TICKETS/handle (Xử lý Logic)	39
Bảng 21: Thư mục BTL_BOOK_MOVIE_TICKETS/views (Giao diện)	40

CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI

1.1. Giới thiệu đề tài

Đề tài tập trung vào việc nghiên cứu và phát triển một Ứng dụng web đặt vé xem phim hoàn chỉnh mang tên Scarlet Cinema. Ứng dụng này được thiết kế theo kiến trúc Client-Server, cung cấp hai giao diện chính: Giao diện Khách hàng (Client-side) cho phép người dùng cuối đặt vé và quản lý hồ sơ, và Giao diện Quản trị (Admin Panel) chuyên biệt cho việc quản lý toàn bộ dữ liệu nghiệp vụ và theo dõi hiệu suất.

1.1.1. Lý do chọn đề tài

Trong bối cảnh công nghệ số phát triển mạnh mẽ, nhu cầu về các dịch vụ trực tuyến tiện lợi, nhanh chóng ngày càng tăng cao. Việc xây dựng một hệ thống đặt vé xem phim trực tuyến là cần thiết vì những lý do sau:

Nâng cao Trải nghiệm Người dùng (UX): Hệ thống cho phép khách hàng xem thông tin phim, kiểm tra suất chiếu và chọn ghế một cách trực quan từ mọi thiết bị, loại bỏ sự bất tiện của việc xếp hàng chờ đợi tại rạp. Giao diện được thiết kế theo phong cách Dark Mode hiện đại bằng Tailwind CSS, tối ưu hóa sự thoải mái khi sử dụng.

Tối ưu hóa Vận hành và Quản lý Rạp: Hệ thống cung cấp các công cụ Admin mạnh mẽ để quản lý danh mục (phim, rạp, phòng chiếu) và đặc biệt là logic phức tạp của ngành điện ảnh như kiểm tra chông lán lịch chiếu (có tính cả thời gian buffer).

Kiểm soát Tài chính và Thống kê: Admin có thể theo dõi Tổng doanh thu, Tổng vé đã bán và phân tích biểu đồ thống kê đơn hàng theo tháng, hỗ trợ ra quyết định kinh doanh hiệu quả.

Đảm bảo Tính toàn vẹn và Bảo mật: Ứng dụng sử dụng Prepared Statements (PDO) để chống lại SQL Injection và triển khai logic chặn Race Condition trong quá trình đặt vé để đảm bảo không có hai người dùng nào đặt cùng một ghế.

1.1.2. Mục tiêu đề tài

Mục tiêu cụ thể của đề tài là xây dựng thành công một ứng dụng web đạt các yêu cầu sau:

- Hoàn thiện Module Khách hàng: Cho phép người dùng Đăng ký/Đăng nhập, Xem lịch sử đặt vé (bao gồm chi tiết từng vé), và thực hiện quy trình đặt vé 4 bước (Chọn ngày, suất, ghế, thanh toán).

- Phát triển Module Quản trị: Cung cấp đầy đủ chức năng CRUD (Create, Read, Update, Delete) cho Phim, Rạp, Phòng chiếu, Suất chiếu và Người dùng.

- Triển khai Logic Nghiệp vụ Phức tạp: Thành công trong việc lập trình logic kiểm tra trùng lịch suất chiếu và xây dựng công cụ chỉnh sửa sơ đồ ghế trực quan cho Admin.

- Thống kê và Báo cáo: Tích hợp Dashboard hiển thị các chỉ số hiệu suất quan trọng (KPIs) và biểu đồ phân tích.

1.2. Phạm vi đề tài

1.1.1. Đối tượng nghiên cứu

Đề tài tập trung nghiên cứu và xây dựng các thành phần cốt lõi của một hệ thống đặt vé:

Cơ sở Dữ liệu: Nghiên cứu cấu trúc dữ liệu quan hệ (bảng movies, shows, screens, bookings, booking_items) và các ràng buộc toàn vẹn dữ liệu (khóa ngoại, ràng buộc UNIQUE).

Logic Backend (PHP): Nghiên cứu các mô hình thiết kế phần mềm (như Repository Pattern) và các thuật toán xử lý nghiệp vụ (ví dụ: thuật toán kiểm tra chồng lấn thời gian suất chiếu).

Giao diện và Tương tác (JavaScript/CSS): Nghiên cứu cách sử dụng CSS Grid và JavaScript để tạo sơ đồ ghế động và quản lý trạng thái chọn ghế trên Client-side.

1.1.2. Phạm vi nghiên cứu

Phạm vi triển khai của ứng dụng được giới hạn trong các chức năng sau:

Module Khách hàng:

Auth: Đăng ký, Đăng nhập, Đăng xuất, Đổi mật khẩu.

Booking: Xem phim, chọn suất chiếu, chọn ghế theo sơ đồ (seat map), chuyển đến bước thanh toán (giả lập VNPay/Cash).

Profile: Xem và cập nhật thông tin cá nhân, xem lịch sử đơn đặt vé chi tiết.

Module Quản trị (Admin):

Quản lý Tài khoản: CRUD Người dùng (Admin/Customer) và Phân quyền.

Quản lý Nội dung: CRUD Phim, Rạp, Phòng chiếu, Suất chiếu (bao gồm kiểm tra trùng lịch và chỉnh sửa layout ghế).

Báo cáo: Dashboard tổng hợp và danh sách Đơn đặt vé.

1.3. Dự kiến công việc

Giai đoạn/Nhóm Công việc	Công việc Chi tiết	Thành phần CSDL/Code
Thiết lập Hạ tầng	Cài đặt môi trường phát triển (Local Server, MySQL), xây dựng cấu trúc CSDL ban đầu, thiết kế lớp Repository và các hàm Auth Helper.	db.php, repository.php, auth_helper.php, cinema_update.sql
Phát triển Module Admin	Xây dựng các file handler (*_handle.php) và các file view (views/admin/*.php) cho toàn bộ chức năng CRUD. Triển khai logic phức tạp về kiểm tra trùng lịch chiếu và sơ đồ ghế.	shows_handle.php, editSeatLayout.php, views/admin/*

Phát triển Module Khách hàng	Xây dựng giao diện Khách hàng (Trang chủ, Auth, Profile), lập trình logic chọn ghế động (booking_clinet.js), và luồng xử lý đặt vé (booking_process.php).	views/clinet/*, booking_clinet.js, booking_process.php
Kiểm thử và Báo cáo	Kiểm thử các chức năng nghiệp vụ cốt lõi (trùng ghế, trùng lịch). Hoàn thiện trang Dashboard, báo cáo thống kê, và chỉnh sửa lỗi giao diện cuối cùng.	index.php (Admin), style.css

Bảng 01: Dự kiến công việc

1.4. Công nghệ sử dụng

1.4.1. Cơ sở Lý thuyết & Công cụ Lập trình

Khu vực	Công nghệ/Khái niệm	Vai trò và Tầm quan trọng
Backend	PHP 8+	Ngôn ngữ chính để xử lý logic Server-side, quản lý luồng dữ liệu và API.
Dữ liệu	MySQL & PDO	Lưu trữ dữ liệu quan hệ; PDO đảm bảo an toàn truy vấn thông qua Prepared Statements.
Kiến trúc	Repository Pattern	Tách biệt tầng truy cập dữ liệu (reponsitory.php) khỏi tầng logic nghiệp vụ, tăng khả năng bảo trì và mở rộng.

Frontend/UX	Html,Css,JavaScript	Cung cấp logic động cho Carousel, Modal, và các chức năng cốt lõi như tính giá và vẽ sơ đồ ghế.
-------------	---------------------	---

Bảng 02: Cơ sở lý thuyết

1.4.2. Thư viện và Framework

Thư viện	Mục đích sử dụng	Tệp tham chiếu chính
Tailwind CSS	Phát triển giao diện nhanh chóng và thiết lập phong cách Dark Mode hiện đại, đảm bảo tính Responsive.	style.css, tất cả các tệp View
CKEditor 5	Trình soạn thảo Rich Text chuyên nghiệp cho việc nhập liệu trường Mô tả phim trong Admin Panel.	addMovie.php, editMovie.php
Lucide Icons	Bộ icon đơn giản, hiện đại được sử dụng để tăng tính trực quan cho Navigation Bar và Buttons.	header.php, side_bar.php

Bảng 03: Thư viện và framework

CHƯƠNG 2 : KHẢO SÁT PHÂN TÍCH YÊU CẦU

2.1. Tìm hiểu nghiệp vụ

2.1.1. Mô tả nghiệp vụ thực tế của rạp chiếu phim

- Nghiệp vụ thực tế của một rạp chiếu phim hiện đại được chia thành ba luồng chính: Quản lý Nội dung/Lịch chiếu, Vận hành phòng vé, và Trải nghiệm Khách hàng.

- Luồng Nghiệp vụ Quản lý Nội dung (Admin/Staff)

Luồng này đảm bảo rằng rạp chiếu phim có thể lên lịch và chuẩn bị các suất chiếu một cách hiệu quả:

Quản lý Danh mục Phim (Movies): Nghiệp vụ yêu cầu lưu trữ chi tiết các thông tin về phim, bao gồm tiêu đề, thời lượng (phút), rating, mô tả, ngày phát hành và các URL tài nguyên (banner, trailer).

Quản lý Cấu trúc Rạp (Theaters & Screens): Cần quản lý thông tin địa điểm của rạp và cấu hình chi tiết từng phòng chiếu (sức chứa, loại phòng 2D/3D/IMAX).

Quản lý Lịch chiếu (Shows) và Nghiệp vụ Chồng lấn: Đây là nghiệp vụ phức tạp nhất. Khi lên lịch một suất chiếu, hệ thống phải đảm bảo rằng suất chiếu mới không bị chồng lấn về thời gian với bất kỳ suất chiếu nào khác trong cùng một phòng chiếu.

Tính toán Thời gian Kết thúc: Thời gian kết thúc suất chiếu phải được tính bằng Thời gian Bắt đầu + Thời lượng Phim + Thời gian Buffer (ví dụ 15 phút) để dọn dẹp/quảng cáo.

Quản lý Sơ đồ Ghế: Cấu hình chi tiết bố cục ghế của từng phòng chiếu, định nghĩa loại ghế (Standard, VIP, Disabled) và lối đi. Việc này ảnh hưởng trực tiếp đến giá vé và trải nghiệm khách hàng.

- Luồng Nghiệp vụ Giao dịch (Booking & Payment) Luồng này mô tả quá trình từ khi khách hàng chọn ghế cho đến khi giao dịch hoàn tất:

Lựa chọn Ghế theo Thực tế: Khách hàng cần một sơ đồ trực quan, hiển thị rõ ràng những ghế nào đã được bán cho suất chiếu đó và những ghế nào còn trống.

Định giá Động: Giá vé cuối cùng không chỉ dựa trên giá cơ bản của suất chiếu mà còn phụ thuộc vào loại ghế mà khách hàng chọn (ví dụ: ghế VIP có hệ số giá 1.5, ghế Disabled có hệ số giá 0.8).

Ngăn chặn Trùng lặp (Race Condition): Trong môi trường đa người dùng, nghiệp vụ yêu cầu kiểm tra lại lần cuối ngay trước khi ghi vào CSDL để đảm bảo ghế vừa chọn chưa bị người khác đặt mua trong tích tắc đó.

Quản lý Trạng thái: Giao dịch cần ghi nhận trạng thái của đơn hàng (pending, confirmed, cancelled) và trạng thái thanh toán (unpaid, paid).

2.1.2. Nhu cầu cần phát triển ứng dụng web

Phát triển ứng dụng web là giải pháp then chốt để giải quyết các hạn chế và nâng cao năng lực kinh doanh của Scarlet Cinema:

Nhu cầu từ phía Khách hàng (Client-side)

Nhu cầu	Ứng dụng Giải quyết như thế nào	Module Hỗ trợ
Tiện lợi & Tương tác	Cung cấp giao diện đặt vé 24/7. Sơ đồ ghế trực quan (seat map) giúp khách hàng dễ dàng hình dung và chọn vị trí yêu thích.	Trang đặt vé, booking_clinet.js
Quản lý Cá nhân	Cho phép khách hàng tự đăng ký, cập nhật hồ sơ, đổi mật khẩu và tra cứu lịch sử đơn hàng chi tiết (thông tin phim, ghế, giá vé).	Module Auth/Profile
Minh bạch thông tin	Khách hàng dễ dàng xem thông tin chi tiết các phim đang và sắp chiếu, bao gồm cả trailer và rating.	Trang Chủ, Modal Chi tiết

Nhu cầu từ phía Quản trị (Admin Panel)

Nhu cầu	Ứng dụng Giải quyết như thế nào	Module Hỗ trợ
Quản lý tập trung	Cung cấp một giao diện Admin duy nhất để thực hiện toàn bộ CRUD cho tất cả các danh mục (Phim, Rạp, Suất chiếu), thay vì phải thao tác trực tiếp trên CSDL hoặc nhiều hệ thống rời rạc.	Admin Panel (views/admin/*.php)
Nghệp vụ phức tạp	Tự động hóa việc kiểm tra chồng lấn lịch chiếu và cung cấp công cụ vẽ sơ đồ ghế (tạo JSON layout).	Screens Handle, Shows Handle
Kiểm soát và Báo cáo	Cung cấp Dashboard tổng quan các KPI tài chính (doanh thu, vé đã bán) và biểu đồ phân tích. Cho phép Admin xem chi tiết từng đơn đặt vé (booking items) qua AJAX.	Dashboard, Bookings List
Bảo mật và Phân quyền	Đảm bảo chỉ những tài khoản có vai trò admin mới được phép truy cập và thực hiện thay đổi.	Auth Helper, Side Bar

Bảng 04: Nhu cầu phát triển

2.2. Phân tích yêu cầu

Danh Sách Chức Năng và Đối Tượng Sử Dụng Hệ thống SCARLET CINEMA được chia thành hai phân hệ chính: Phân hệ Khách hàng (Client) và Phân hệ Quản trị (Admin).

Danh sách Chức năng (Functional Requirements - FR)

Mã FR	Tên Chức năng	Phân hệ	Đối tượng sử dụng
FR01	Đăng ký/Đăng nhập/Đăng xuất	Chung	Khách hàng, Admin
FR02	Xem thông tin Phim	Client	Khách hàng
FR03	Lọc và chọn Suất chiếu	Client	Khách hàng
FR04	Chọn ghế trực quan (Seat Map)	Client	Khách hàng
FR05	Xử lý thanh toán/Đặt vé	Client	Khách hàng
FR06	Quản lý Hồ sơ & Lịch sử đặt vé	Client	Khách hàng
FR07	Quản lý Phim (CRUD)	Admin	Admin
FR08	Quản lý Rạp/Phòng chiếu (CRUD & Sơ đồ ghế)	Admin	Admin
FR09	Quản lý Suất chiếu (CRUD & Kiểm tra trùng lịch)	Admin	Admin
FR10	Quản lý Đơn đặt vé (Xem chi tiết)	Admin	Admin
FR11	Quản lý Người dùng (CRUD & Phân quyền)	Admin	Admin
FR12	Xem Báo cáo/Thống kê (Dashboard)	Admin	Admin

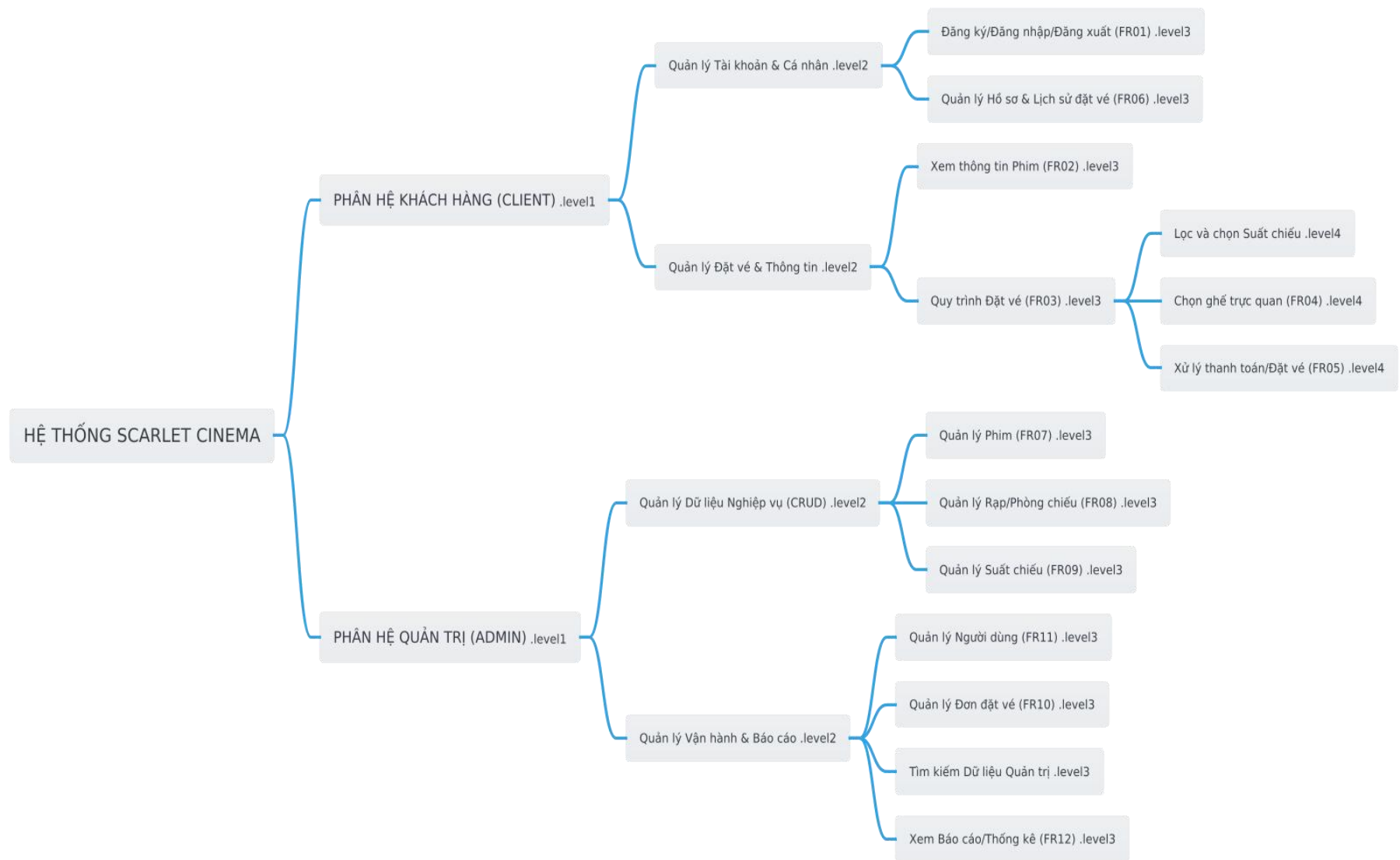
Bảng 05: Phân tích yêu cầu

Danh sách Đối tượng Sử dụng (Actors)

Đối tượng	Vai trò Chính trong hệ thống	Module/Tập Liên quan
Khách hàng	Đăng ký, Đặt vé, Xem lịch sử giao dịch.	views/clinet/*
Admin	Quản lý toàn bộ dữ liệu nghiệp vụ, cấp phát quyền, theo dõi hiệu suất.	views/admin/*, *_handle.php
Hệ thống CSDL	Lưu trữ và truy vấn dữ liệu.	reponsitory.php

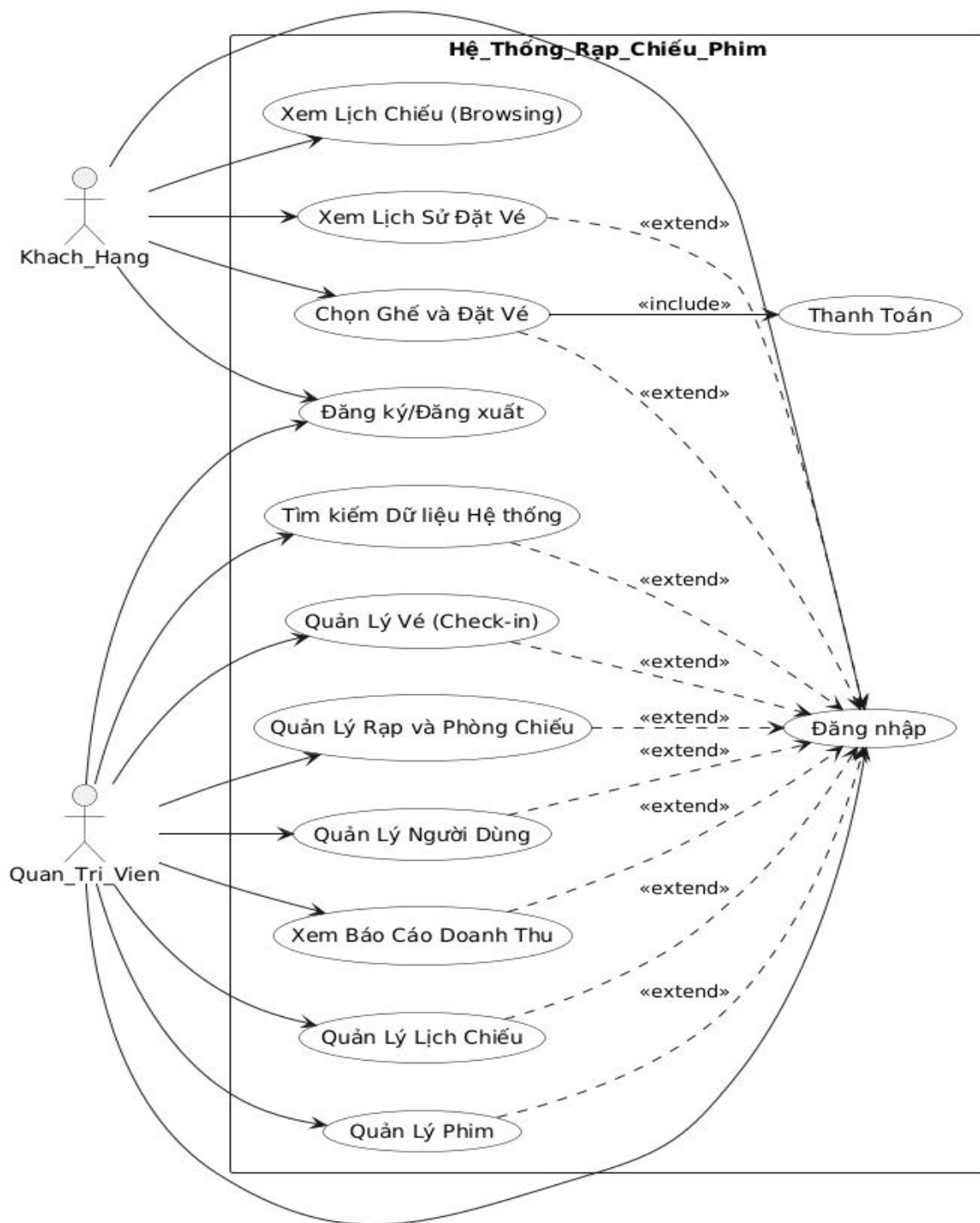
Bảng 06: Đối tượng sử dụng

Biểu Đồ Phân Cấp Chức Năng (Functional Decomposition Diagram - FDD)



Hình ảnh 01: Biểu đồ phân cấp

Biểu Đồ Use Case Tổng Quát



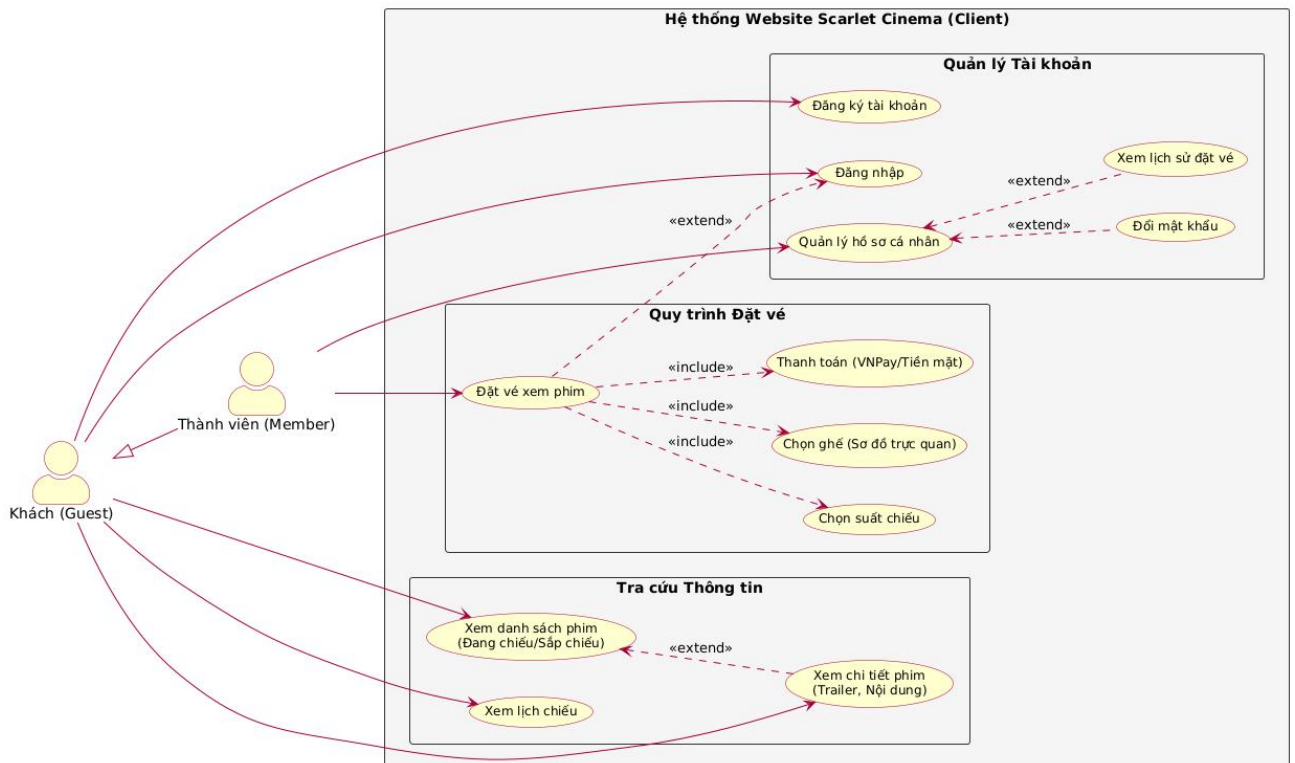
Hình ảnh 02: Use case Tổng quát

Phân hệ Khách hàng (Client Use Cases)

Use Case	Chi tiết	Actor	Tập liên quan
Đăng ký/Đăng nhập	Xác thực người dùng để truy cập các chức năng đặt vé.	Khách hàng	account_handle.php
Xem Phim & Lịch chiếu	Tra cứu phim đang/sắp chiếu và lọc suất chiếu theo ngày/rạp.	Khách hàng	booking.php
Đặt vé	Lựa chọn ghế, xác nhận giỏ hàng, và chuyển sang thanh toán.	Khách hàng	booking_clinet.js, booking_process.php
Quản lý Profile	Cập nhật thông tin cá nhân và xem lịch sử đặt vé chi tiết.	Khách hàng	profile.php

Bảng 07: Phân hệ khách hàng

- Biểu đồ tổng quát Use Case khách hàng:



Hình 04: Usecase tổng quát khách hàng

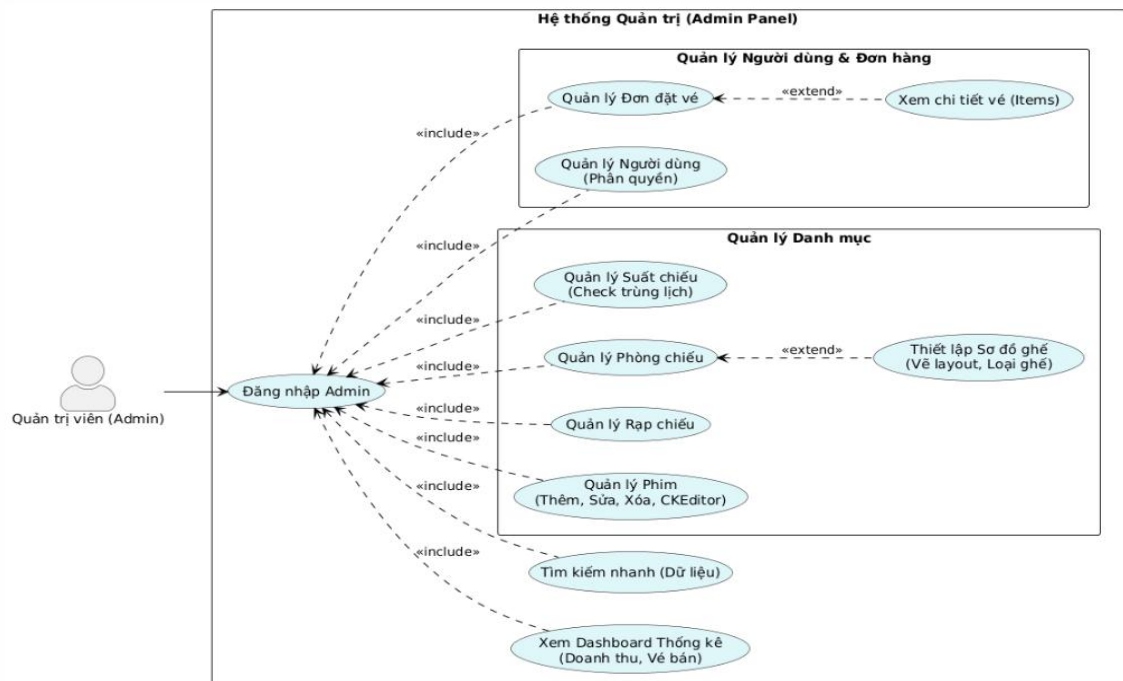
Phân hệ Quản trị (Admin Use Cases)

Use Case	Chi tiết	Actor	Tập liên quan (Backend/Frontend)
Quản lý Danh mục Phim	CRUD thông tin chi tiết phim: tiêu đề, thời lượng, mô tả (CKEditor), rating, URLs banner/trailer.	Admin	movies_handle.php, movies.php, addMovie.php, editMovie.php
Quản lý Rạp	CRUD thông tin rạp chiếu: tên, địa chỉ, thành phố.	Admin	theaters_handle.php, theaters.php, editTheater.php
Quản lý Phòng chiếu & Sơ đồ Ghế	CRUD thông tin phòng (tên, loại phòng). Quản lý chi tiết sơ đồ ghế (tạo/sửa cấu trúc ghế, loại ghế (Standard/VIP/Disabled/Aisle) sau khi chuyển sang mô hình 3NF.	Admin	screens_handle.php, screens.php, editScreen.php, editSeatLayout.php
Quản lý Suất chiếu	CRUD lịch chiếu. Kiểm tra nghiệp vụ chồng lấn (có tính Buffer time) trong cùng một phòng chiếu.	Admin	shows_handle.php, shows.php, editShow.php, addShow.php
Quản lý Người dùng	Thêm, sửa, xóa tài khoản người dùng, thiết lập vai trò (customer, admin) và cập nhật mật khẩu.	Admin	user_handle.php, users.php, editUser.php, addUser.php
Quản lý Đơn hàng	Xem danh sách tất cả đơn đặt vé, trạng thái (pending/confirmed) và trạng	Admin	bookings.php, booking_detail_handler.php

	thải thanh toán (paid/unpaid). Xem chi tiết từng mục vé (booking items).		
Xem Thống kê	Theo dõi các chỉ số KPI (Tổng Doanh thu, Tổng vé đã bán), biểu đồ đơn hàng theo tháng, và phân bổ loại vé trên Dashboard.	Admin	index.php (Admin)
Tìm kiếm Nhanh	Tìm kiếm dữ liệu tổng hợp trên tất cả các bảng (Users, Movies, Shows, Bookings...) hoặc theo từng thực thể cụ thể.	Admin	search.php, search_handle.php

Bảng 08: Phân hệ quản trị

- Biểu đồ tổng quát usecase admin:



Hình 03: Usecase tổng quát admin

2.3. Thiết kế DataBase

2.3.1.Cấu trúc Từng Bảng Trong Database

- Bảng Người dùng:

Tên Cột	Kiểu Dữ Liệu	Ràng Buộc	Mô Tả
id	bigint unsigned	PK,AUTO_INCREMENT	Khóa chính, ID người dùng.
email	varchar(255)	UNIQUE	Địa chỉ email (dùng để đăng nhập).
password_hash	varchar(255)	NOT NULL	Mật khẩu đã hash.
full_name	varchar(150)	NULL	Tên đầy đủ của người dùng.
role	enum	NOT NULL (default 'customer')	Vai trò ('customer', 'staff', 'admin').

Bảng 09: Bảng người dùng trong database

-Bảng rạp phim:

Tên Cột	Kiểu Dữ Liệu	Ràng Buộc	Mô Tả
id	int unsigned	PK, AUTO_INCREMENT	ID rạp chiếu phim.
name	varchar(200)	NOT NULL	Tên rạp.
address	varchar(300)	NULL	Địa chỉ chi tiết.
city	varchar(100)	NULL	Thành phố.

Bảng 10: Bảng rạp phim trong database

- Bảng phim:

Tên Cột	Kiểu Dữ Liệu	Ràng Buộc	Mô Tả
id	int unsigned	PK, AUTO_INCREMENT	ID phim.

title	varchar(255)	NOT NULL	Tiêu đề phim.
duration_min	smallint unsigned	NOT NULL	Thời lượng phim (phút).
release_date	date	NULL	Ngày phát hành.
rating	varchar(16)	NULL	Phân loại độ tuổi (ví dụ: G, PG-13).

Bảng 11: Bảng Phim trong database

- Bảng Phân loại ghế:

Tên Cột	Kiểu Dữ Liệu	Ràng Buộc	Mô Tả
id	tinyint unsigned	PK , AUTO_INCREMENT	ID loại ghế.
code	varchar(20)	UNIQUE	Mã loại ghế ('standard', 'vip', v.v.).
name_vi	varchar(100)	NOT NULL	Tên tiếng Việt của loại ghế.
price_modifier	decimal(5,2)	NOT NULL (default 1.00)	Hệ số nhân giá (ví dụ: 1.5 cho VIP).

Bảng 12: Phân Loại Ghế trong database

-Bảng phòng chiếu:

Tên Cột	Kiểu Dữ Liệu	Ràng Buộc	Mô Tả
id	int unsigned	PK , AUTO_INCREMENT	ID phòng chiếu.
theater_id	int unsigned	FK (theaters)	ID rạp chiếu chứa phòng này.
name	varchar(100)	NOT NULL	Tên phòng (ví dụ: Phòng 1, Imax 3D).
capacity	smallint unsigned	NULL	Sức chứa tối đa của phòng.

Bảng 13: Phòng Chiếu trong database

- Bảng ghế ngồi:

Tên Cột	Kiểu Dữ Liệu	Ràng Buộc	Mô Tả
id	bigint unsigned	PK , AUTO_INCREMENT	ID ghế.
screen_id	int unsigned	FK (screens)	ID phòng chiếu chứa ghế này.
seat_type_id	tinyint unsigned	FK (seat_types)	ID loại ghế (Standard, VIP).
seat_code	varchar(16)	UNIQUE (screen_id)	Mã ghế (ví dụ: A1, B5).
row_letter	varchar(2)	NOT NULL	Hàng ghế (A, B, C...).

Bảng 14: Bảng ghế ngồi trong database

- Bảng suất chiếu:

Tên Cột	Kiểu Dữ Liệu	Ràng Buộc	Mô Tả
id	bigint unsigned	PK , AUTO_INCREMENT	ID suất chiếu.
movie_id	int unsigned	FK (movies)	ID phim được chiếu.
screen_id	int unsigned	FK (screens)	ID phòng chiếu.
show_time	datetime	NOT NULL	Thời gian bắt đầu chiếu.
format	enum	DEFAULT '2D'	Định dạng chiếu (2D, 3D, IMAX...).
price	decimal(10,2)	NOT NULL (default 0.00)	Giá vé cơ bản của suất chiếu.

Bảng 15: Bảng suất chiếu trong database

- Bảng hóa đơn đặt vé:

Tên Cột	Kiểu Dữ Liệu	Ràng Buộc	Mô Tả
id	bigint unsigned	PK , AUTO_INCREMENT	ID đơn đặt vé.
user_id	bigint unsigned	FK (users), NULL	ID người dùng đặt (có thể

			NULL nếu là khách vắng lai).
show_id	bigint unsigned	FK (shows)	ID suất chiếu đã đặt.
status	enum	DEFAULT 'pending'	Trạng thái đặt vé (pending, confirmed, cancelled...).
payment_status	enum	DEFAULT 'unpaid'	Trạng thái thanh toán (unpaid, paid, refunded).
total_amount	decimal(15,0)	NOT NULL (default 0)	Tổng số tiền đơn hàng.

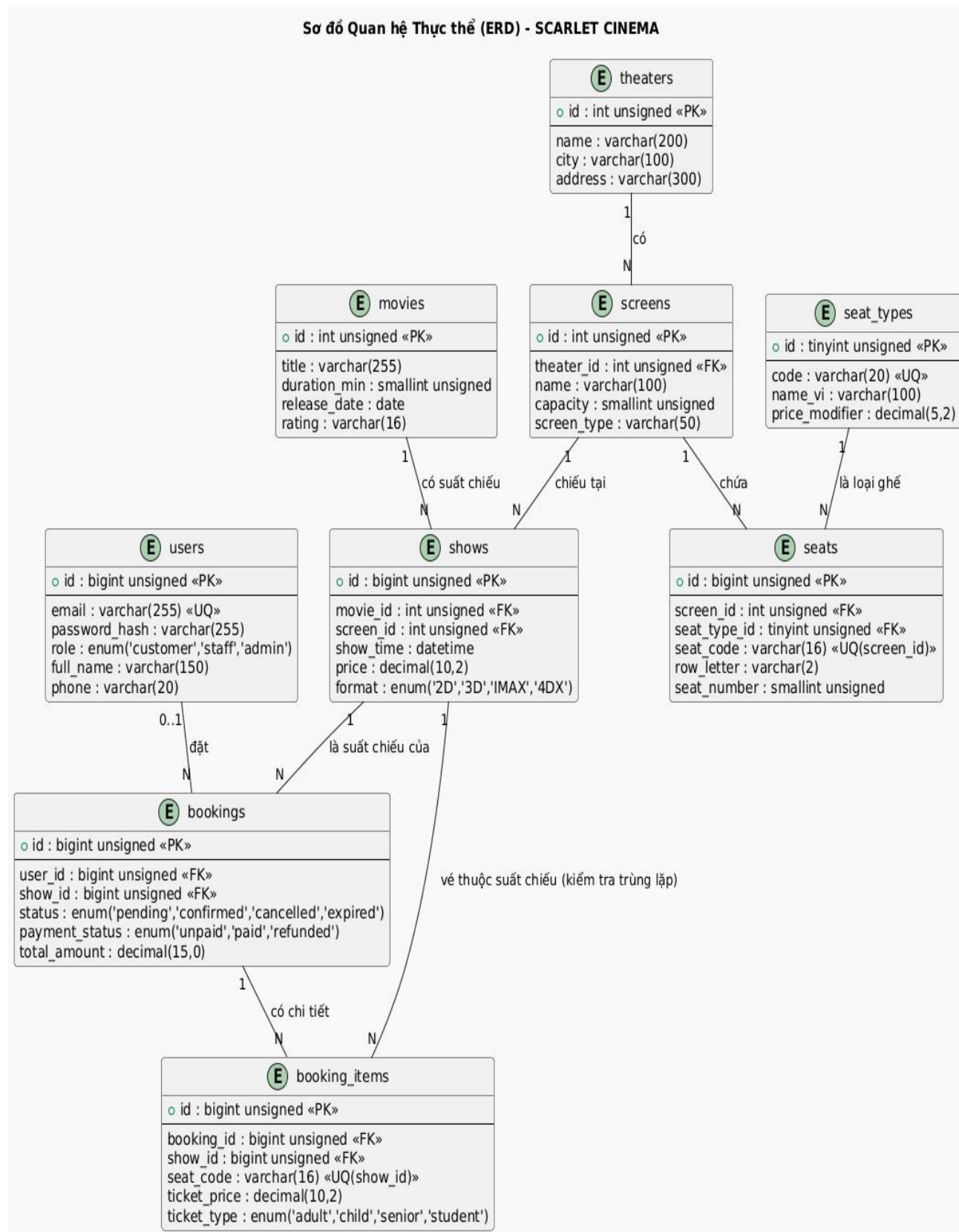
Bảng 16: Hóa đơn đặt vé trong database

- Bảng chi tiết hóa đơn đặt vé:

Tên Cột	Kiểu Dữ Liệu	Ràng Buộc	Mô Tả
id	bigint unsigned	PK, AUTO_INCREMENT	ID chi tiết vé.
booking_id	bigint unsigned	FK (bookings)	ID đơn đặt vé mẹ.
show_id	bigint unsigned	FK (shows)	ID suất chiếu (để kiểm tra ràng buộc).
seat_code	varchar(16)	UNIQUE (show_id)	Mã ghế đã đặt trong suất chiếu này.
ticket_price	decimal(10,2)	NOT NULL (default 0.00)	Giá vé cuối cùng cho ghế này.
ticket_type	enum	DEFAULT 'adult'	Loại vé (adult, child, senior...).

Bảng 17: Chi tiết hóa đơn trong database

2.3.2. Sơ đồ quan hệ thực thể (ERD)



Hình 03: Sơ đồ quan hệ thực thể URD

Chương 3: Lập trình xây dựng và cài đặt hệ thống

3.1. Xây dựng database trên Mysql

3.1.1. Cấu trúc các bảng trên Mysql

Table: **booking_items**

Columns:

id	bigint UN AI PK
booking_id	bigint UN
show_id	bigint UN
seat_code	varchar(16)
ticket_price	decimal(10,2)
ticket_type	enum('adult','child','senior','student')
status	enum('booked','cancelled','checked_in')
created_at	datetime

Hình 04: Cấu trúc bảng Booking_items.

Table: **screens**

Columns:

id	int UN AI PK
theater_id	int UN
name	varchar(100)
capacity	smallint UN
screen_type	varchar(50)
seat_layout	json
created_at	datetime
updated_at	datetime

Hình 05: Cấu trúc bảng Screens

Table: bookings

Columns:

id	bigint UN AI PK
user_id	bigint UN
show_id	bigint UN
status	enum('pending','confirmed','cancelled','expired')
total_amount	decimal(15,0)
payment_method	varchar(50)
payment_status	enum('unpaid','paid','refunded')
created_at	datetime
updated_at	datetime

Hình 06: Cấu trúc bảng Bookings.

Table: movies

Columns:

id	int UN AI PK
title	varchar(255)
duration_min	smallint UN
description	text
rating	varchar(16)
release_date	date
banner_url	varchar(512)
trailer_url	varchar(512)
created_at	datetime
updated_at	datetime

Hình 07: Cấu trúc bảng movies.

Table: `shows`

Columns:

id	bigint UN AI PK
movie_id	int UN
screen_id	int UN
show_time	datetime
format	enum('2D','3D','IMAX','4DX')
price	decimal(10,2)
status	varchar(20)
created_at	datetime
updated_at	datetime

Hình 08: Cấu trúc bảng shows

Table: `theaters`

Columns:

id	int UN AI PK
name	varchar(200)
address	varchar(300)
city	varchar(100)
phone	varchar(20)
created_at	datetime
updated_at	datetime

Hình 09: Cấu trúc bảng theaters.

Table: **users**

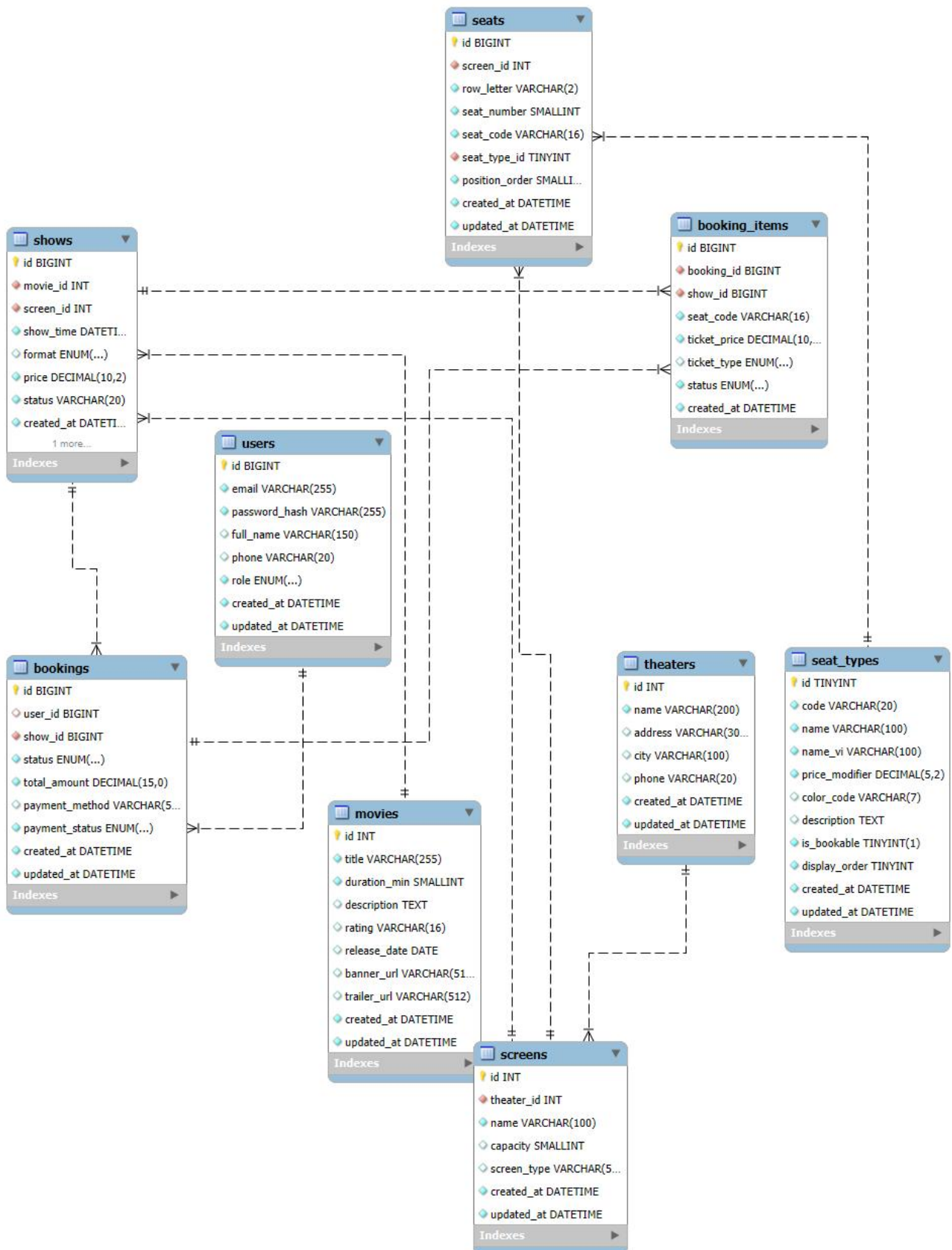
Columns:

id	bigint UN AI PK
email	varchar(255)
password_hash	varchar(255)
full_name	varchar(150)
phone	varchar(20)
role	enum('customer','staff','admin')
created_at	datetime
updated_at	datetime

Hình 10: Cấu trúc bảng users

3.1.2. Sơ đồ quan hệ đã xây dựng

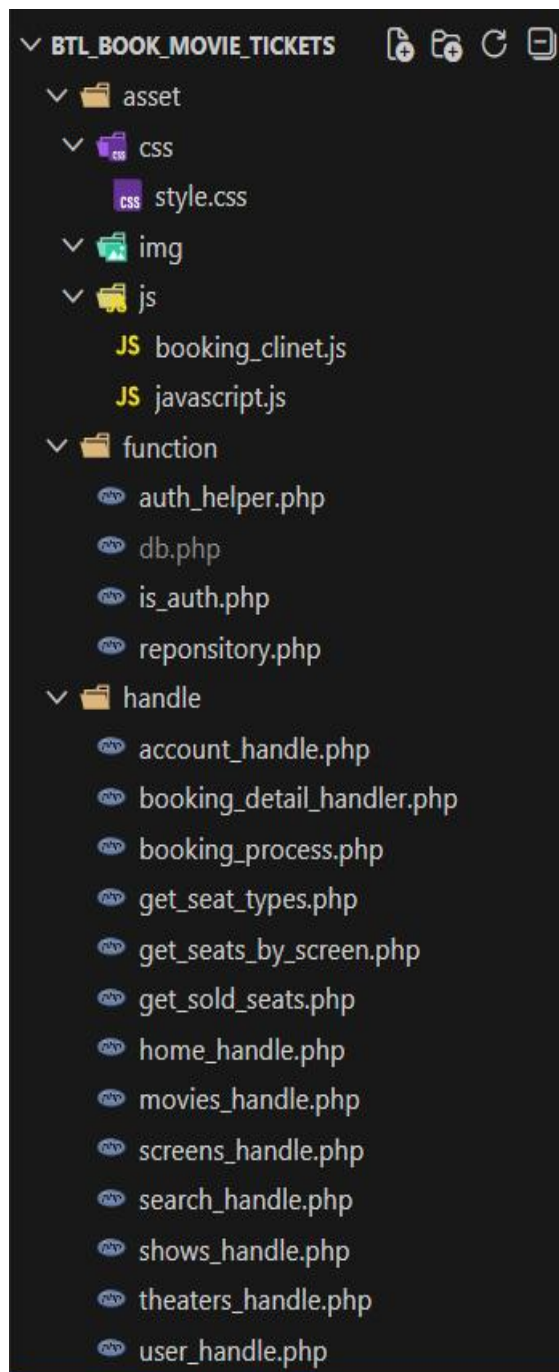
Sơ đồ quan hệ đã xây dựng trên MySql



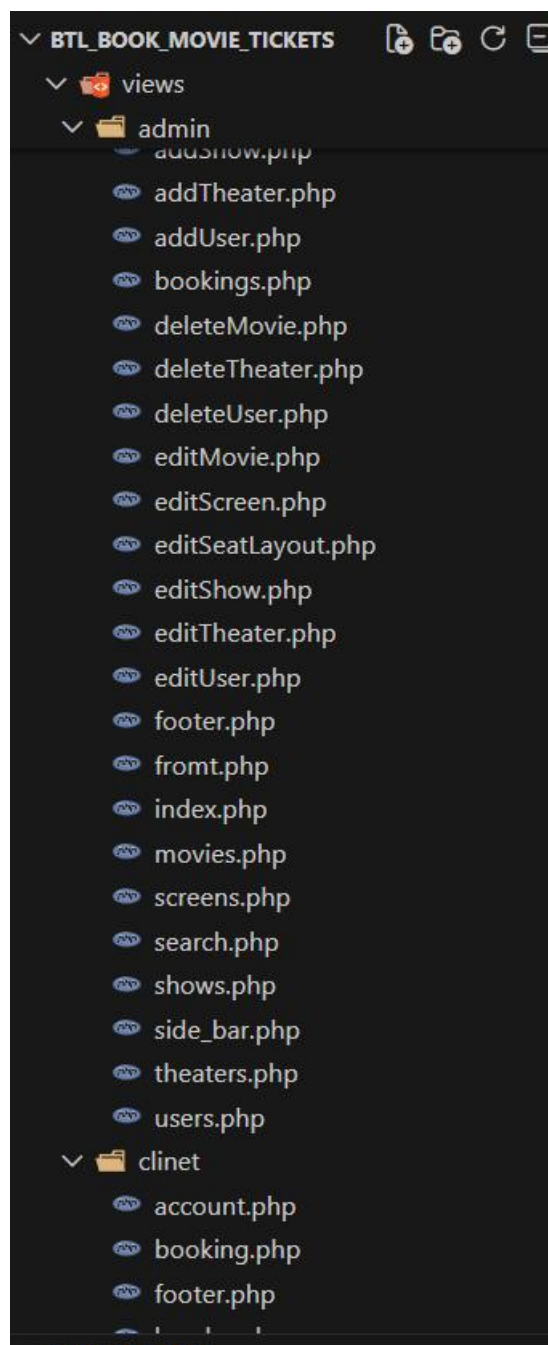
Hình 11: Sơ đồ quan hệ đã xây dựng trên MySQL

3.2. Tổ chức mã nguồn

3.2.1. Cấu trúc folder



Hình 12: Cấu trúc folder



Hình 13: Cấu trúc folder

3.2.2. Mô tả cấu trúc và chức năng file mã nguồn

1. Thư mục BTL_BOOK_MOVIE_TICKETS/asset (file tĩnh)

Tên file	Đường dẫn	Mô tả
style.css	asset/css/	Chứa mã CSS định kiểu (style) cho toàn bộ giao diện người dùng và quản trị viên của hệ thống.
booking_clinet.js	asset/js/	Chứa mã JavaScript xử lý logic đặt vé phía người dùng (client) như chọn ghế, cập nhật tổng tiền.
javascript.js	asset/js/	Chứa mã JavaScript tổng quát hoặc các hàm tiện ích dùng chung cho giao diện người dùng.
img	asset/img/	Thư mục chứa các file hình ảnh (logo, poster phim, icon, v.v.) được sử dụng trong hệ thống.

Bảng 18: Thư mục BTL_BOOK_MOVIE_TICKETS/asset (file tĩnh)

2. Thư mục BTL_BOOK_MOVIE_TICKETS/function (xử lý database)

Tên file	Đường dẫn	Mô tả
auth_helper.php	function/	Chứa các hàm hỗ trợ xác thực (authentication) và ủy quyền (authorization), ví dụ: kiểm tra người dùng đã đăng nhập chưa, lấy thông tin người dùng hiện tại.
db.php	function/	Chứa mã nguồn để kết nối và cấu hình cơ sở dữ liệu

		(database connection), thường là MySQL.
is_auth.php	function/	Chứa logic kiểm tra quyền truy cập hoặc trạng thái đăng nhập của người dùng trước khi cho phép họ thực hiện một hành động nào đó.
reponsitory.php	function/	Thường là viết tắt của repository (kho lưu trữ), chứa các lớp hoặc hàm chịu trách nhiệm truy vấn, thêm, sửa, xóa (CRUD) dữ liệu trực tiếp với database.

Bảng 19: Thư mục BTL_BOOK_MOVIE_TICKETS/function (xử lý database)

3. Thư mục BTL_BOOK_MOVIE_TICKETS/handle (Xử lý Logic)

Tên file	Đường dẫn	Mô tả
account_handle.php	handle/	Xử lý các yêu cầu liên quan đến tài khoản người dùng (đăng nhập, đăng ký, cập nhật thông tin cá nhân).
booking_detail_handler.php	handle/	Xử lý logic nghiệp vụ cho việc xem chi tiết đơn đặt vé đã tạo.
booking_process.php	handle/	Xử lý logic chính của quá trình đặt vé (tạo đơn hàng, trừ tiền, cập nhật trạng thái ghế).
get_seat_types.php	handle/	Xử lý yêu cầu lấy danh

		sách hoặc thông tin các loại ghế (ghế thường, VIP, đôi) cho giao diện.
get_seats_by_screen.php	handle/	Xử lý yêu cầu lấy sơ đồ ghế và thông tin của từng ghế theo mã phòng chiếu/màn hình cụ thể.
get_sold_seats.php	handle/	Xử lý yêu cầu lấy danh sách các ghế đã được bán/đặt cho một suất chiếu cụ thể.
home_handle.php	handle/	Xử lý logic cho Trang chủ (ví dụ: lấy danh sách phim đang chiếu, phim sắp chiếu).
movies_handle.php	handle/	Xử lý logic cho quản lý và hiển thị thông tin phim (thêm, sửa, xóa phim, chi tiết phim).
screens_handle.php	handle/	Xử lý logic cho quản lý phòng chiếu (thêm, sửa, xóa phòng chiếu).
search_handle.php	handle/	Xử lý logic cho chức năng tìm kiếm phim hoặc suất chiếu.
shows_handle.php	handle/	Xử lý logic cho quản lý suất chiếu (thêm, sửa, xóa suất chiếu).
theaters_handle.php	handle/	Xử lý logic cho quản lý rạp chiếu phim (thêm, sửa, xóa thông tin rạp).
user_handle.php	handle/	Xử lý logic cho quản lý

		người dùng (thêm, sửa, xóa tài khoản, phân quyền).
--	--	--

Bảng 20: Thư mục BTL_BOOK_MOVIE_TICKETS/handle (Xử lý Logic)

4. Thư mục BTL_BOOK_MOVIE_TICKETS/views (Giao diện)

Tên file	Đường dẫn	Mô tả
account.php	views/client/	Giao diện trang thông tin tài khoản cá nhân của khách hàng.
booking.php	views/client/	Giao diện chính của quá trình đặt vé (hiển thị sơ đồ ghế, nút xác nhận).
footer.php	views/client/	Phần chân trang (footer) của giao diện khách hàng.
index.php	views/client/	Giao diện Trang chủ của khách hàng.
movies.php	views/client/	Giao diện danh sách phim đang chiếu hoặc sắp chiếu.
addShow.php	views/admin/	Giao diện cho phép Quản trị viên thêm suất chiếu mới.
addTheater.php	views/admin/	Giao diện cho phép Quản trị viên thêm rạp chiếu mới.
addUser.php	views/admin/	Giao diện cho phép Quản trị viên thêm người dùng mới.
bookings.php	views/admin/	Giao diện danh sách quản lý các đơn đặt vé đã được tạo.
deleteMovie.php	views/admin/	Thao tác/Xử lý xóa phim (có thể là một trang xác nhận hoặc chỉ là file xử

		lý).
editSeatLayout.php	views/admin/	Giao diện cho phép Quản trị viên chỉnh sửa bố cục ghế trong phòng chiếu.
footer.php	views/admin/	Phần chân trang (footer) của giao diện quản trị viên.
index.php	views/admin/	Giao diện trang đăng nhập hoặc trang tổng quan của Quản trị viên.
movies.php	views/admin/	Giao diện danh sách và quản lý phim.
screens.php	views/admin/	Giao diện danh sách và quản lý phòng chiếu/màn hình.
search.php	views/admin/	Giao diện tìm kiếm trong khu vực quản trị.
shows.php	views/admin/	Giao diện danh sách và quản lý suất chiếu.
side_bar.php	views/admin/	Thanh điều hướng bên (sidebar) của giao diện quản trị viên.
theaters.php	views/admin/	Giao diện danh sách và quản lý rạp chiếu.
users.php	views/admin/	Giao diện danh sách và quản lý người dùng.

Bảng 21: Thư mục BTL_BOOK_MOVIE_TICKETS/views (Giao diện)

3.3. Xây dựng các chức năng theo thiết kế

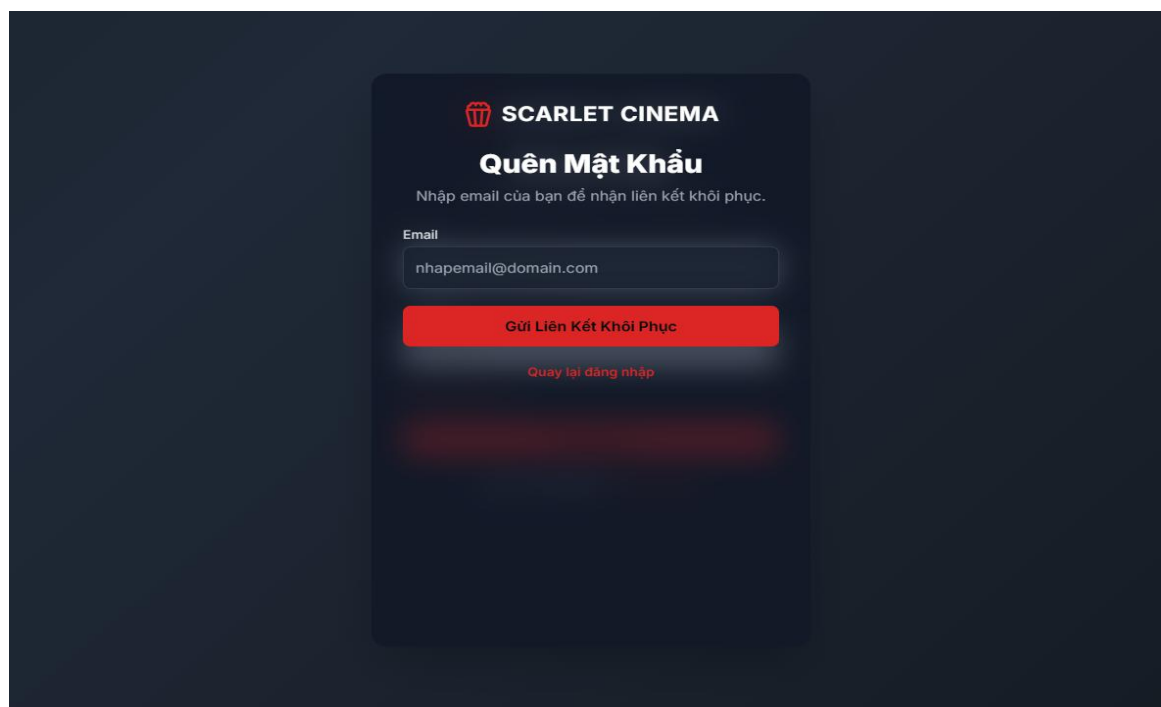
3.3.1. Giao diện website

The image shows a registration form for SCARLET CINEMA. At the top, there is a logo with a red gift icon and the text "SCARLET CINEMA". Below the logo, the title "Tạo Tài Khoản" (Create Account) is displayed in bold, followed by the subtitle "Gia nhập Scarlet để nhận ưu đãi!" (Join Scarlet to receive discounts!). The form contains four input fields: "Họ và Tên" (Last Name and First Name) with the value "Nguyễn Văn A", "Email" with the value "adkey@gmail.com", "Mật khẩu" (Password) with masked characters "*****", and "Xác nhận mật khẩu" (Confirm password) with the placeholder "Nhập lại mật khẩu". A red button labeled "Tạo Tài Khoản" is positioned below the input fields. At the bottom, there is a link "Đã có tài khoản? Đăng nhập ngay" (Already have an account? Log in now).

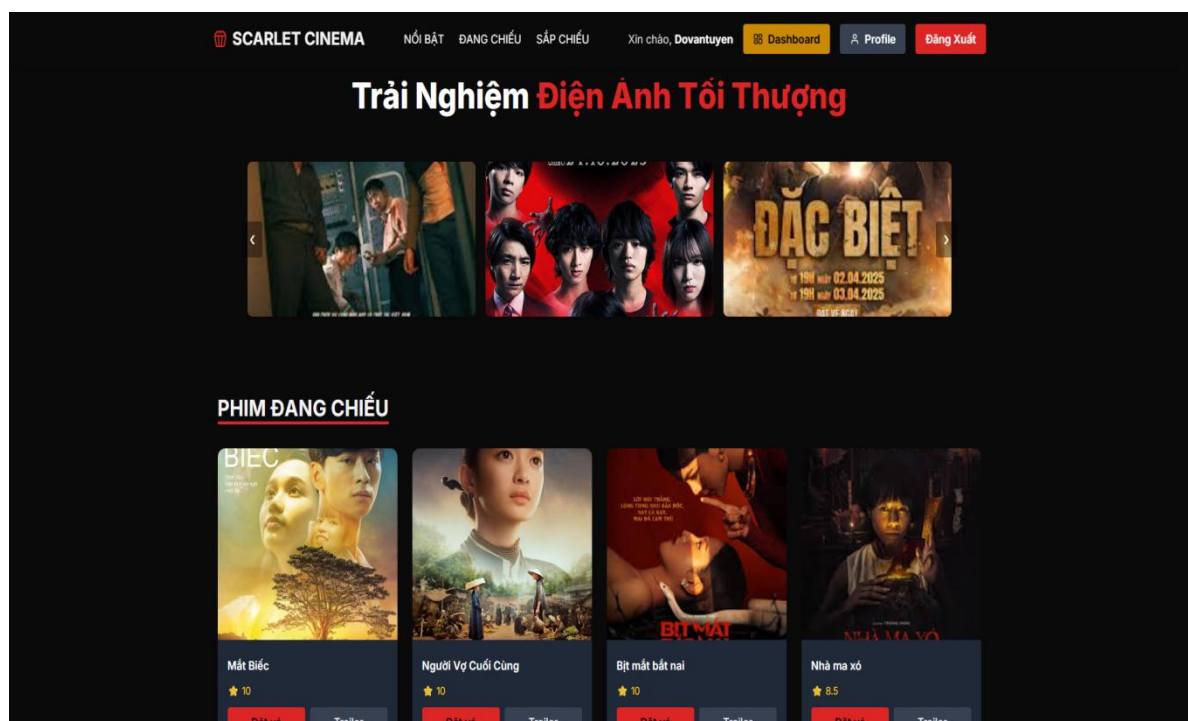
Hình 16: Giao diện đăng ký

The image shows a login form for SCARLET CINEMA. At the top, there is a logo with a red gift icon and the text "SCARLET CINEMA". Below the logo, the title "Đăng Nhập" (Login) is displayed in bold, followed by the subtitle "Trải nghiệm điện ảnh tối thượng!" (Experience the ultimate cinema!). The form contains two input fields: "Email" with the value "adkey@gmail.com" and "Mật khẩu" (Password) with masked characters "*****". A red button labeled "Đăng Nhập" is positioned below the input fields. Above the button, there is a link "Quên mật khẩu?" (Forgot password?). At the bottom, there is a link "Chưa có tài khoản? Đăng ký ngay" (Don't have an account? Register now).

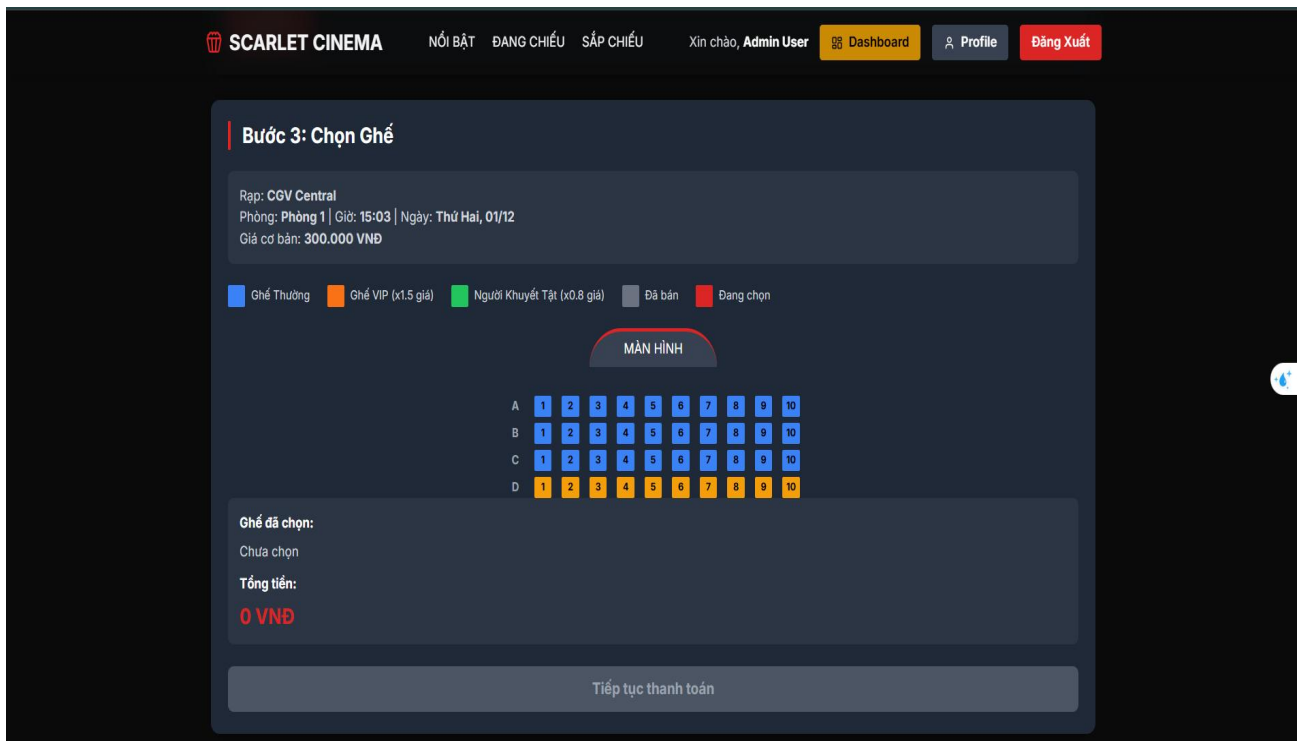
Hình 17: Giao diện đăng nhập



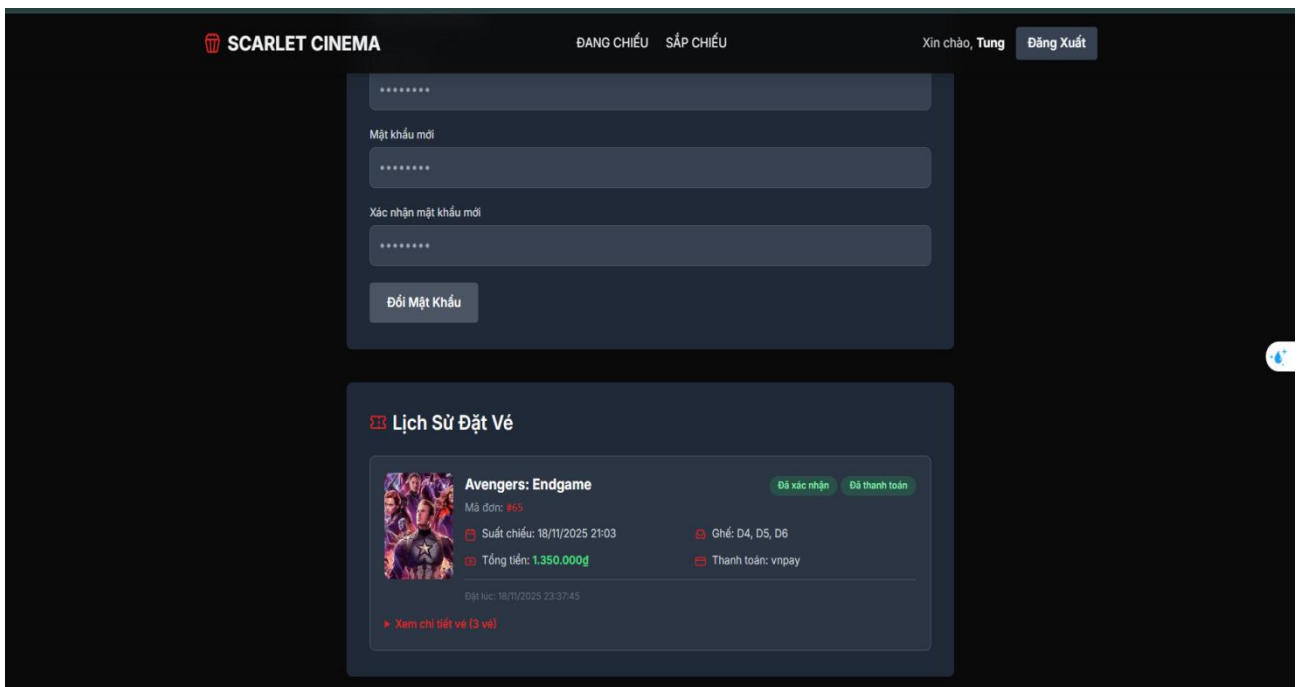
Hình 18: Giao diện quên mật khẩu



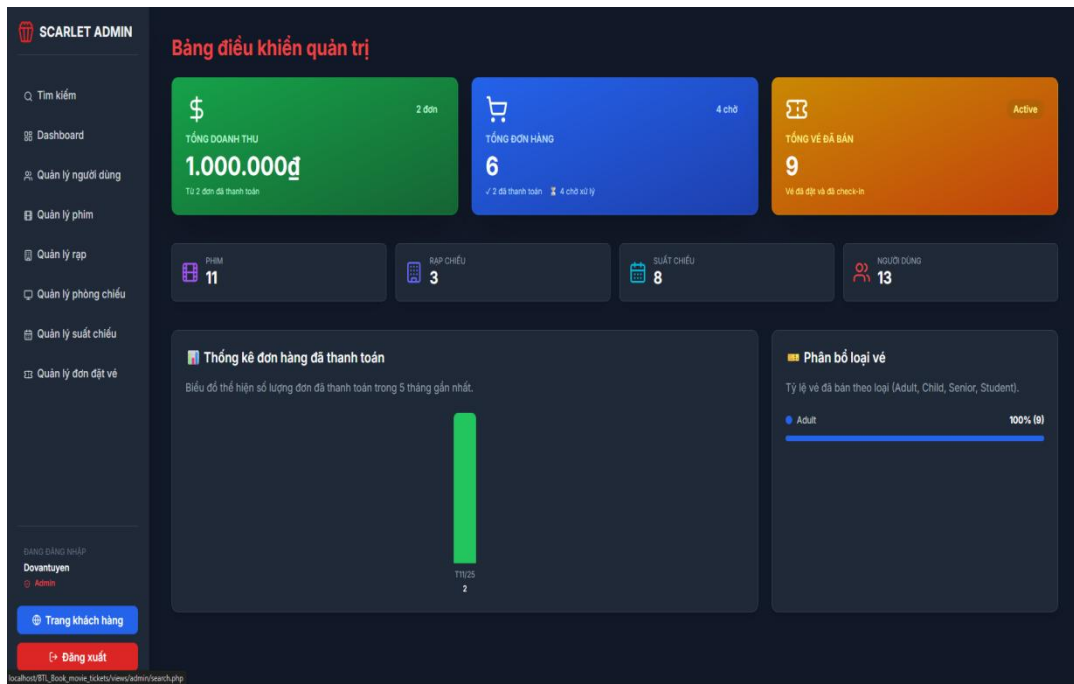
Hình 19: Giao diện trang khách hàng



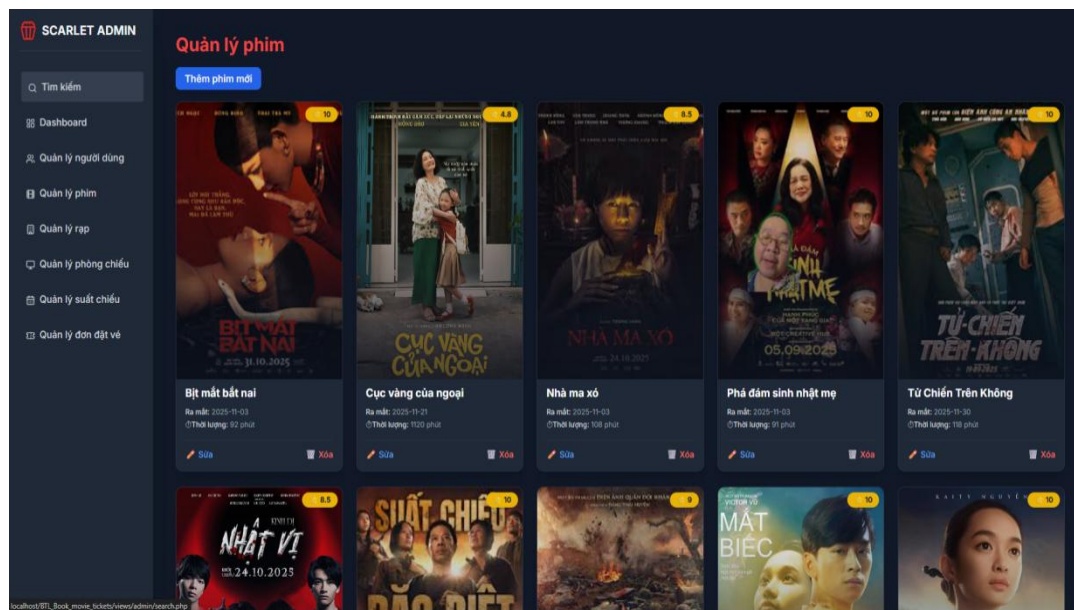
Hình 20: Giao diện đặt vé



Hình 21: Giao diện trang quản lý vé đã đặt

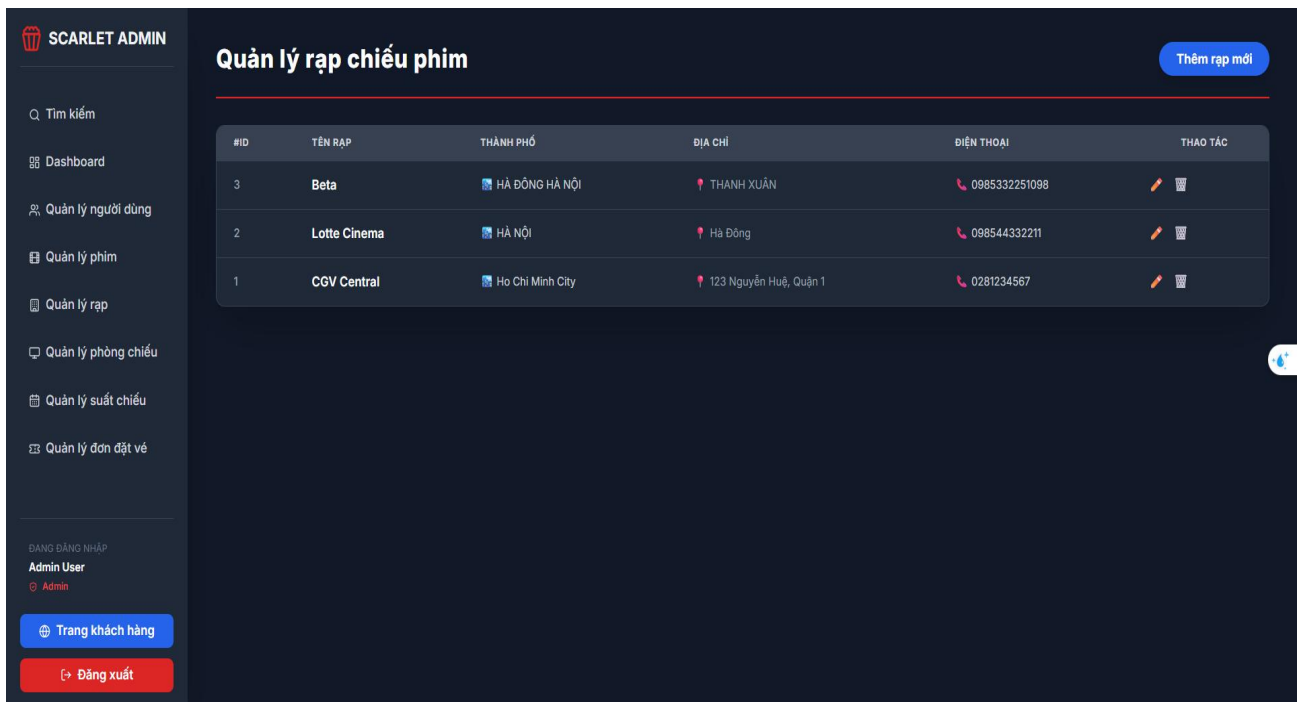


Hình 22: Giao diện dashboard(admin)



Hình 23 Giao diện quản lý phim(admin)

Hình 25: Giao diện quản lý đơn đặt vé(admin)



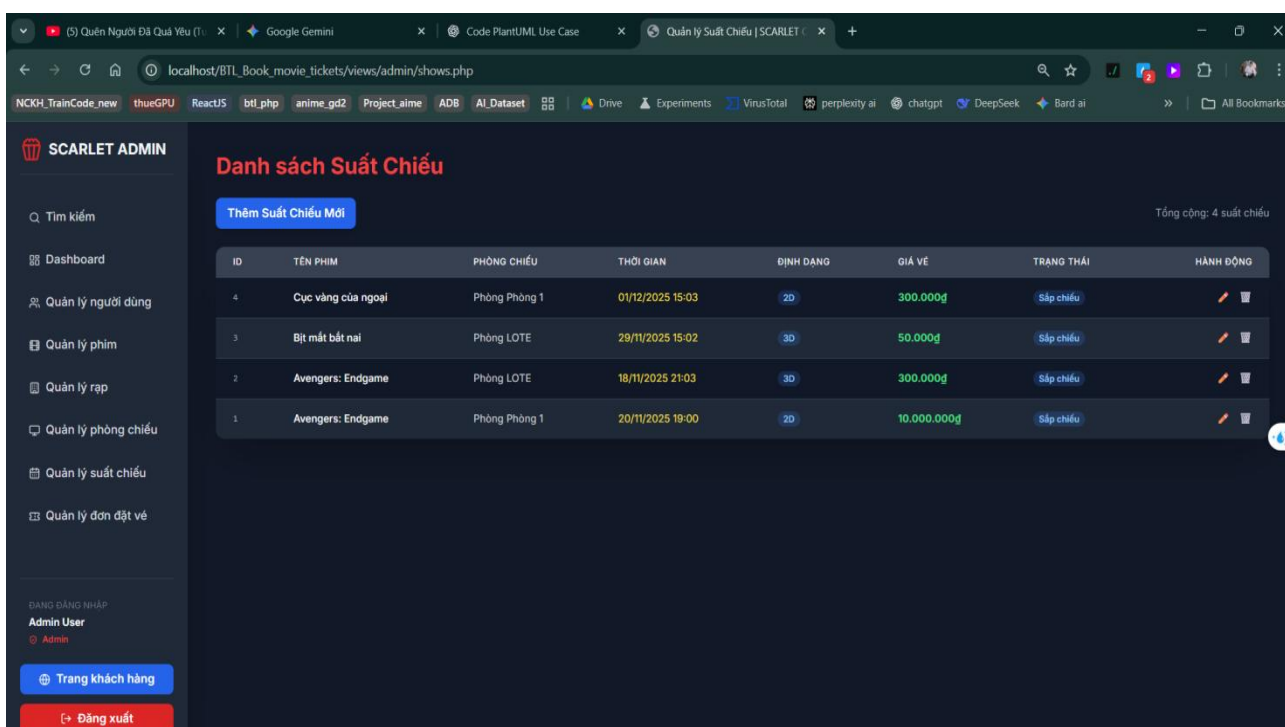
Hình 26: Giao diện quản lý rạp phim(admin)



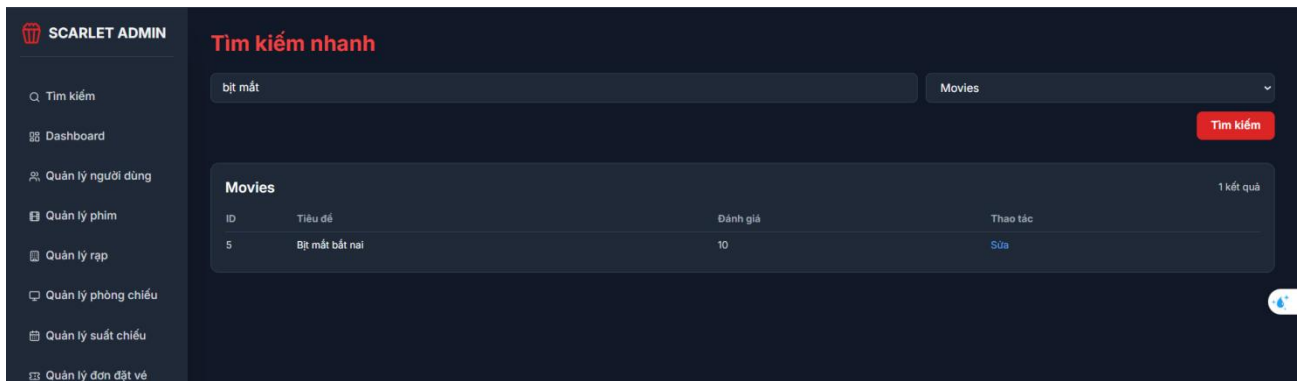
Hình 27: Giao diện quản lý phòng chiếu(admin)



Hình 28: Giao diện quản lý sơ đồ ghế(admin)



Hình 29: Giao diện quản lý suất chiếu(admin)



Hình 30: Giao diện Tìm kiếm(admin)

3.3.2. Moudle mã nguồn

1. Module xử lý đăng ký đăng nhập

```
<?php

if (session_status() === PHP_SESSION_NONE)
    session_start();

require_once __DIR__ . '/../function/reponsitory.php';

function handleUser($action, $data = [], $id = null)
{
    $repo = new Repository('users');
    $response = [
        'success' => false,
        'message' => 'Có lỗi xảy ra, vui lòng thử lại.'
    ];

    try {
        switch ($action) {
            case 'add':
                // Kiểm tra dữ liệu
                if (empty($data['full_name']) || empty($data['email']) ||
                    empty($data['password_hash'])) {
                    $response['message'] = 'Vui lòng nhập đầy đủ thông tin.';
                    return $response;
                }
                // Kiểm tra định dạng email
```

```

        if (!filter_var($data['email'], FILTER_VALIDATE_EMAIL)) {
            $response['message'] = ' Định dạng email không hợp lệ.';
            return $response;
        }
        // Kiểm tra email đã tồn tại
        $existing = $repo->findBy('email', $data['email']);
        if ($existing) {
            $response['message'] = ' Email này đã được sử dụng.';
            return $response;
        }
        // Thêm mới
        if ($repo->insert($data)) {
            $response['success'] = true;
            $response['message'] = '✔ Thêm người dùng thành công!';
        } else {
            $response['message'] = '✗ Không thể thêm người dùng.';
        }
        return $response;
    case 'edit':
        if (!$id || empty($data)) {
            $response['message'] = ' Thiếu dữ liệu để cập nhật.';
            return $response;
        }
        // Nếu có email mới thì kiểm tra trùng
        if (isset($data['email']) && !empty($data['email'])) {
            $existing = $repo->findBy('email', $data['email']);
            if ($existing && $existing['id'] != $id) {
                $response['message'] = ' Email này đã tồn tại ở người dùng khác.';
                return $response;
            }
        }
        if ($repo->update($id, $data)) {
            $response['success'] = true;
            $response['message'] = '✔ Cập nhật thông tin người dùng thành công!';
        } else {
            $response['message'] = '✗ Cập nhật thất bại.';
        }
        return $response;
    case 'delete':

```

```

        if (!$id) {
            $response['message'] = 'Thiếu ID người dùng để xóa.';
            return $response;
        }

        if (session_status() === PHP_SESSION_NONE)
            session_start();

        if (isset($_SESSION['user']) && intval($_SESSION['user']['id']) === intval($id))
        {
            $response['message'] = 'Bạn không thể xóa chính mình.';
            return $response;
        }

        $ok = $repo->delete($id);

        $response['success'] = (bool) $ok;

        $response['message'] = $ok ? 'Xóa người dùng thành công.' : 'Xóa thất bại.';
        return $response;

        default:
            $response['message'] = 'Hành động không hợp lệ.';
            return $response;
    }

    } catch (Exception $e) {
        $response['message'] = 'Lỗi hệ thống: ' . $e->getMessage();
        return $response;
    }
}

$redirect_url = '../views/clinet/account.php';

// Lấy action và view
$action = $_POST['action'] ?? $_GET['action'] ?? '';
$view_from_form = $_POST['view'] ?? $_GET['view'] ?? '';

// 1. Xử lý ĐĂNG KÝ (REGISTER) - Dùng POST
if ($_SERVER['REQUEST_METHOD'] === 'POST' && $action === 'register') {

    $data = [
        'full_name' => trim($_POST['fullname'] ?? ''),
        'email' => trim($_POST['email'] ?? ''),
        'password_hash' => password_hash($_POST['password'] ?? '', PASSWORD_DEFAULT),
        'role' => 'customer' // QUAN TRỌNG: Phải là 'customer' để khớp với ENUM trong database
    ];

    if ((($_POST['password'] ?? '') !== ($_POST['confirmPassword'] ?? ''))) {
        $res = ['success' => false, 'message' => '✗ Mật khẩu xác nhận không trùng khớp.'];
    }
}

```

```

        $_SESSION['flash_view'] = 'register';
    } else {
        $res = handleUser('add', $data);
        if ($res['success']) {
            // Đăng ký thành công -> Chuyển về form Login để đăng nhập
            $res['message'] = '✓ Đăng ký thành công! Vui lòng đăng nhập.';
            $_SESSION['flash_view'] = 'login';
        } else {
            // Đăng ký thất bại -> Giữ ở form Register
            $_SESSION['flash_view'] = 'register';
        }
    }

    // Lưu thông báo và chuyển hướng về trang form
    $_SESSION['flash_message'] = $res['message'];
    $_SESSION['flash_success'] = $res['success'];
    // Quay về trang form account.php
    header('Location: ' . $redirect_url . '?view=' . $_SESSION['flash_view']);
    exit;
}

// 2. Xử lý ĐĂNG NHẬP (LOGIN) - Dùng POST
if ($_SERVER['REQUEST_METHOD'] === 'POST' && $action === 'login') {

    $email = trim($_POST['email'] ?? '');
    $password = $_POST['password'] ?? '';
    $repo = new Repository('users');
    $user = $repo->findBy('email', $email);
    if ($user && password_verify($password, $user['password_hash'])) {
        // Đăng nhập thành công
        $_SESSION['user'] = [
            'id' => $user['id'],
            'full_name' => $user['full_name'],
            'email' => $user['email'],
            'role' => $user['role']
        ];
        // Chuyển hướng người dùng về trang chủ (hoặc trang admin)
        if ($user['role'] === 'admin') {
            // Có thể thêm flash message thành công nếu cần thiết
            header('Location: ../views/admin/index.php'); // Giả định admin index
        } else {

```

```

        // Có thể thêm flash message thành công nếu cần thiết
        $_SESSION['flash_message'] = '✓ Đăng nhập thành công!';
        $_SESSION['flash_success'] = true;

        header('Location: ../views/clinnet/index.php'); // Giả định client index
    }

    exit;
} else {
    // Đăng nhập thất bại
    $_SESSION['flash_message'] = '✗ Sai email hoặc mật khẩu.';
    $_SESSION['flash_success'] = false;
    $_SESSION['flash_view'] = 'login';

    // **SỬA LỖI CHUYỂN HƯỚNG:** Quay lại trang form account.php để hiển thị lỗi
    header('Location: ' . $redirect_url . '?view=login');
    exit;
}
}

// Nếu không có hành động hợp lệ (và không phải đang trong account.php)
if (!empty($action)) {
    header('Location: ' . $redirect_url);
    exit;
}

```

2. Module thêm phim

```

// --- Xử lý Thêm phim (Tương tự ADD USER) ---
if ($action === 'add' && $_SERVER['REQUEST_METHOD'] === 'POST') {
    $data = [
        'title' => trim($_POST['title'] ?? ''),
        'duration_min' => intval($_POST['duration_min'] ?? 0),
        'rating' => floatval($_POST['rating'] ?? 0),
        'release_date' => trim($_POST['release_date'] ?? null),
        'banner_url' => trim($_POST['banner_url'] ?? null),
        'trailer_url' => trim($_POST['trailer_url'] ?? null),
        'description' => $_POST['description'] ?? '',
    ];

    $res = handleMovie('add', $data, null);

    $_SESSION['flash_message'] = $res['message'];
}

```

```

        $_SESSION['flash_success'] = $res['success'];

        if (!$res['success']) {

            $_SESSION['form_data'] = $_POST;

            header('Location: ' . $_SERVER['HTTP_REFERER']);

        } else {

            header('Location: ../views/admin/movies.php'); // Chuyển hướng đến trang danh sách

        }

        exit;

    }

```

3. Module xử lý đặt vé:

```

<?php

ini_set('display_errors', 1);
ini_set('display_startup_errors', 1);
error_reporting(E_ALL);

if (session_status() === PHP_SESSION_NONE) {
    session_start();
}

require_once __DIR__ . "../function/reponsitory.php";
// BẮT BUỘC NGƯỜI DÙNG PHẢI ĐĂNG NHẬP MỚI ĐẶT VÉ ĐƯỢC
if (!isset($_SESSION['user']) || !isset($_SESSION['user']['id'])) {
    $_SESSION['flash_message'] = 'Vui lòng đăng nhập để đặt vé!';
    $_SESSION['flash_success'] = false;
    header('Location: ../views/clinnet/account.php?view=login');
    exit;
}

$user_id = $_SESSION['user']['id'];
// Lấy dữ liệu từ form
$show_id = $_POST['show_id'] ?? null;
$selected_seats_json = $_POST['selected_seats'] ?? null;
$total_amount = $_POST['total_amount'] ?? 0;
$payment_method = $_POST['payment_method'] ?? null;
// === BẮT ĐẦU: LOGIC XỬ LÝ TRẠNG THÁI THANH TOÁN ===
$booking_status = '';
$payment_status = '';
$flash_message = '';

if ($payment_method === 'vnpay' || $payment_method === 'credit_card') {

```

```

        // Nếu là VNPay (hoặc thẻ), coi như đã thanh toán thành công
        $booking_status = 'confirmed'; // Đã xác nhận
        $payment_status = 'paid';      // Đã thanh toán
        $flash_message = 'Thanh toán thành công! Vé của bạn đã được xác nhận.';
    } else {
        // Mặc định là 'cash' (tiền mặt)
        $booking_status = 'pending';
        $payment_status = 'unpaid';
        $flash_message = 'Đặt vé thành công! Vui lòng thanh toán tại quầy.';
    }

    // === KẾT THÚC: LOGIC XỬ LÝ TRẠNG THÁI ===
    // Validate dữ liệu
    if (!$show_id || !$selected_seats_json || !$payment_method) {
        $_SESSION['flash_message'] = 'Dữ liệu đặt vé không đầy đủ!';
        $_SESSION['flash_success'] = false;
        header('Location: ' . $_SERVER['HTTP_REFERER']);
        exit;
    }

    // Parse JSON ghế đã chọn
    $selected_seats = json_decode($selected_seats_json, true);
    if (!is_array($selected_seats) || empty($selected_seats)) {
        $_SESSION['flash_message'] = 'Vui lòng chọn ít nhất một ghế!';
        $_SESSION['flash_success'] = false;
        header('Location: ' . $_SERVER['HTTP_REFERER']);
        exit;
    }

    // Khởi tạo repositories
    $bookingRepo = new Repository('bookings');
    $bookingItemRepo = new Repository('booking_items');
    $showRepo = new Repository('shows');
    $screenRepo = new Repository('screens');

    try {
        // 1. Lấy thông tin suất chiếu và giá cơ bản
        $show = $showRepo->find($show_id);

        if (!$show) {
            throw new Exception('Suất chiếu không tồn tại!');
        }

        $base_price = floatval($show['price']);
    }

```

```

$screen = $screenRepo->find($show['screen_id']);

if (!$screen) {
    throw new Exception('Phòng chiếu không tồn tại!');
}

// 2. Kiểm tra ghế đã được đặt chưa (Race condition prevention)
foreach ($selected_seats as $seat_code) {
    $existing = $bookingItemRepo->findByMultipleFields([
        'show_id' => $show_id,
        'seat_code' => $seat_code
    ]);

    if ($existing && in_array($existing['status'], ['booked', 'checked_in'])) {
        throw new Exception("Ghế {$seat_code} đã được đặt bởi người khác!");
    }
}

// 3. Tạo bản ghi Booking chính (ĐÃ CẬP NHẬT TRẠNG THÁI)
$booking_data = [
    'user_id' => $user_id,
    'show_id' => $show_id,
    'status' => $booking_status, // <-- Đã sửa
    'total_amount' => $total_amount,
    'payment_method' => $payment_method,
    'payment_status' => $payment_status // <-- Đã sửa
];

$bookingRepo->insert($booking_data);

// Lấy booking_id vừa tạo
$booking_id = $bookingRepo->pdo->lastInsertId();

// 4. Lấy thông tin ghế từ bảng seats và tạo booking_items
$seatRepo = new Repository('seats');
$seatTypeRepo = new Repository('seat_types');

foreach ($selected_seats as $seat_code) {
    // Tìm ghế trong bảng seats
    $seat = $seatRepo->findByMultipleFields([

```

```

        'screen_id' => $screen['id'],
        'seat_code' => $seat_code
    ]);

    if (!$seat) {
        throw new Exception("Ghế {$seat_code} không tồn tại trong phòng chiếu!");
    }

    // Lấy thông tin loại ghế từ bảng seat_types
    $seatType = $seatTypeRepo->find($seat['seat_type_id']);
    if (!$seatType) {
        throw new Exception("Loại ghế không tồn tại cho ghế {$seat_code}!");
    }

    // Kiểm tra không cho đặt lối đi
    if (!$seatType['is_bookable'] || $seatType['code'] === 'aisle') {
        throw new Exception("Không thể đặt loại ghế này: {$seat_code} ({$seatType['name_vi']})");
    }

    // Tính giá ghế dựa trên price_modifier từ bảng seat_types
    $price_modifier = floatval($seatType['price_modifier']);
    $seat_price = $base_price * $price_modifier;

    // Tạo booking_item
    $item_data = [
        'booking_id' => $booking_id,
        'show_id' => $show_id,
        'seat_code' => $seat_code,
        'ticket_price' => $seat_price,
        'ticket_type' => 'adult', // Mặc định là người lớn
        'status' => 'booked'
    ];

    $bookingItemRepo->insert($item_data);
}

// 5. Thành công - Redirect về trang xác nhận hoặc profile
$_SESSION['flash_message'] = 'Đặt vé thành công! Vui lòng thanh toán tại quầy.';
$_SESSION['flash_success'] = true;

```

```
        // Redirect về trang profile hoặc booking confirmation
        header('Location: ../views/clinnet/profile.php');
        exit;

    } catch (Exception $e) {
        // Xử lý lỗi
        $_SESSION['flash_message'] = 'Lỗi: ' . $e->getMessage();
        $_SESSION['flash_success'] = false;
        header('Location: ' . $_SERVER['HTTP_REFERER']);
        exit;}
    }
```

KẾT LUẬN

Về ưu điểm, đề tài đã xây dựng thành công một hệ thống đặt vé xem phim trực tuyến hoàn chỉnh, đáp ứng tốt các yêu cầu nghiệp vụ phức tạp của rạp chiếu phim. Hệ thống sở hữu kiến trúc mã nguồn mạch lạc nhờ áp dụng mô hình Repository Pattern và tách biệt rõ ràng giữa Client-Admin, giúp việc bảo trì và mở rộng trở nên dễ dàng. Điểm sáng của ứng dụng là khả năng xử lý các logic nghiệp vụ khó như ngăn chặn xung đột đặt ghế (Race Condition) trong môi trường đa người dùng, thuật toán kiểm tra chồng lấn lịch chiếu tự động và giao diện chọn ghế trực quan, thân thiện. Bên cạnh đó, việc sử dụng PDO và Prepared Statements đã đảm bảo tính an toàn dữ liệu và bảo mật cho hệ thống trước các nguy cơ tấn công phổ biến.

Tuy nhiên, hệ thống vẫn còn tồn tại một số hạn chế nhất định do giới hạn về thời gian và tài nguyên thực hiện. Các tính năng liên quan đến bên thứ ba như thanh toán trực tuyến (VNPay) và gửi email khôi phục mật khẩu hiện tại mới chỉ dừng lại ở mức mô phỏng quy trình (Simulation) trên giao diện mà chưa tích hợp API thực tế. Ngoài ra, hệ thống chưa ứng dụng công nghệ thời gian thực (Real-time) để cập nhật trạng thái ghế ngay lập tức cho các người dùng khác khi có người đang chọn, mà vẫn phụ thuộc vào cơ chế kiểm tra lại dữ liệu (Re-validation) tại bước thanh toán cuối cùng.

Về hướng phát triển, để nâng cao tính ứng dụng thực tiễn, đề tài cần tập trung tích hợp các cổng thanh toán điện tử thực tế và dịch vụ SMTP để gửi vé điện tử qua email. Hệ thống cũng cần mở rộng thêm các module quản lý khách hàng thân thiết (tích điểm, hạng thành viên) và tính năng áp dụng mã giảm giá (Voucher/Coupon). Đặc biệt, việc nghiên cứu và tích hợp công nghệ WebSockets sẽ là bước tiến quan trọng để đồng bộ hóa sơ đồ ghế theo thời gian thực, mang lại trải nghiệm đặt vé mượt mà và chuyên nghiệp hơn trong các khung giờ cao điểm.

Link Source Project

Link GitHub: https://github.com/nguyenthantung2k4/BTL_Book-movie-tickets

DANH MỤC TÀI LIỆU THAM KHẢO

- 1) PHP Documentation – PHP Manual. Truy cập tại: <https://www.php.net/manual/>
- 2) Nguyễn Văn Vinh (2020). Giáo trình Phát triển ứng dụng Web với PHP và MySQL. Nhà xuất bản khoa học và kỹ thuật.
- 3) Xử lý Race Condition và Concurrency trong MySQL:
<https://www.sqlines.com/mysql/how-to/select-update-single-statement-race-condition>