

LOGIC HIỂN THỊ PHIM - CINEBOOK

Mục lục

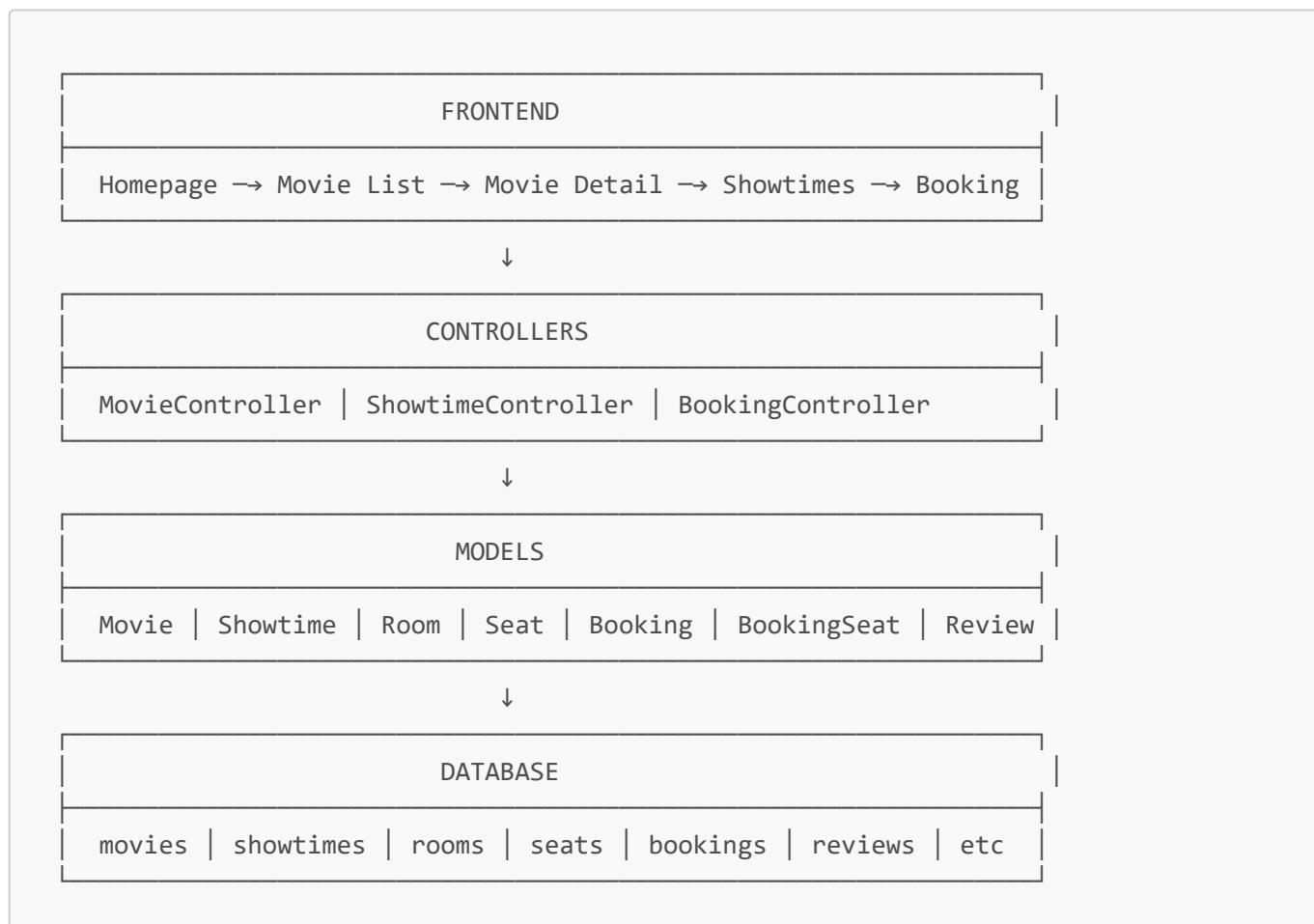
1. [Tổng quan hệ thống](#)
2. [Cấu trúc Database](#)
3. [Models và Relationships](#)
4. [Controllers](#)
5. [Views](#)
6. [Routes](#)
7. [Luồng hoạt động chi tiết](#)
8. [Logic phân loại phim](#)
9. [Quy trình đặt vé](#)

1. Tổng quan hệ thống

Hệ thống quản lý và hiển thị phim bao gồm:

- **Frontend:** Hiển thị phim cho người dùng (homepage, danh sách, chi tiết)
- **Admin Panel:** Quản lý phim, suất chiếu, phòng chiếu
- **Booking System:** Đặt vé, chọn ghế, thanh toán

Sơ đồ tổng quan



2. Cấu trúc Database

Bảng movies

```
CREATE TABLE movies (
    id          INT PRIMARY KEY AUTO_INCREMENT,
    title       VARCHAR(255) NOT NULL,
    language    VARCHAR(100),
    director   VARCHAR(255),
    cast        TEXT,
    duration    INT NOT NULL,           -- Thời lượng (phút)
    release_date DATE NOT NULL,
    age_rating  VARCHAR(10),           -- P, T13, T16, T18
    status      ENUM('now_showing', 'coming_soon', 'ended'),
    poster_url  VARCHAR(500),
    trailer_url VARCHAR(500),
    description TEXT,
    rating_avg  DECIMAL(2,1) DEFAULT 0 -- Điểm trung bình (0-5)
);
```

Bảng genres và movie_genres

```
-- Thể loại phim
CREATE TABLE genres (
    id          INT PRIMARY KEY AUTO_INCREMENT,
    name        VARCHAR(100) NOT NULL,
    description TEXT
);

-- Bảng trung gian (Many-to-Many)
CREATE TABLE movie_genres (
    movie_id    INT,
    genre_id    INT,
    PRIMARY KEY (movie_id, genre_id),
    FOREIGN KEY (movie_id) REFERENCES movies(id),
    FOREIGN KEY (genre_id) REFERENCES genres(id)
);
```

Bảng showtimes

```
CREATE TABLE showtimes (
    id          INT PRIMARY KEY AUTO_INCREMENT,
    movie_id    INT NOT NULL,
    room_id    INT NOT NULL,
    show_date   DATE NOT NULL,
```

```

    show_time  TIME NOT NULL,
    FOREIGN KEY (movie_id) REFERENCES movies(id),
    FOREIGN KEY (room_id) REFERENCES rooms(id)
);

```

Bảng showtime_prices

```

CREATE TABLE showtime_prices (
    id          INT PRIMARY KEY AUTO_INCREMENT,
    showtime_id  INT NOT NULL,
    seat_type_id INT NOT NULL,
    price        DECIMAL(10,2) NOT NULL,
    FOREIGN KEY (showtime_id) REFERENCES showtimes(id),
    FOREIGN KEY (seat_type_id) REFERENCES seat_types(id)
);

```

Bảng reviews

```

CREATE TABLE reviews (
    id          INT PRIMARY KEY AUTO_INCREMENT,
    user_id     INT NOT NULL,
    movie_id    INT NOT NULL,
    rating      INT NOT NULL,           -- 1-5 sao
    comment     TEXT,
    created_at  TIMESTAMP,
    updated_at  TIMESTAMP,
    FOREIGN KEY (user_id) REFERENCES users(id),
    FOREIGN KEY (movie_id) REFERENCES movies(id)
);

```

3. Models và Relationships

File: [app/Models/Movie.php](#)

```

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;

class Movie extends Model
{
    protected $fillable = [
        'title', 'language', 'director', 'cast', 'duration',
        'release_date', 'age_rating', 'status', 'poster_url',
    ];
}

```

```
'trailer_url', 'description', 'rating_avg'  
];  
  
// Quan hệ nhiều-nhiều với Genre  
public function genres()  
{  
    return $this->belongsToMany(Genre::class, 'movie_genres');  
}  
  
// Quan hệ một-nhiều với Review  
public function reviews()  
{  
    return $this->hasMany(Review::class);  
}  
  
// Quan hệ một-nhiều với Showtime  
public function showtimes()  
{  
    return $this->hasMany>Showtime::class);  
}  
  
// Accessor: Lấy danh sách thể loại dạng chuỗi  
public function getGenresStringAttribute()  
{  
    return $this->genres->pluck('name')->implode(', ');  
}  
  
// Method: Cập nhật điểm trung bình từ reviews  
public function updateAverageRating()  
{  
    $avg = $this->reviews()->avg('rating') ?? 0;  
    $this->update(['rating_avg' => round($avg, 1)]);  
}  
}
```

File: [app/Models>Showtime.php](#)

```
<?php  
  
namespace App\Models;  
  
use Illuminate\Database\Eloquent\Model;  
  
class Showtime extends Model  
{  
    public $timestamps = false;  
  
    protected $fillable = ['movie_id', 'room_id', 'show_date', 'show_time'];  
  
    protected $casts = [  
        'show_date' => 'date',
```

```
    'show_time' => 'datetime:H:i',
];

public function movie()
{
    return $this->belongsTo(Movie::class);
}

public function room()
{
    return $this->belongsTo(Room::class);
}

public function showtimePrices()
{
    return $this->hasMany(ShowtimePrice::class);
}

public function showtimeSeats()
{
    return $this->hasMany(ShowtimeSeat::class);
}

public function bookings()
{
    return $this->hasMany(Booking::class);
}
}
```

File: app/Models/Genre.php

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;

class Genre extends Model
{
    public $timestamps = false;

    protected $fillable = ['name', 'description'];

    public function movies()
    {
        return $this->belongsToMany(Movie::class, 'movie_genres');
    }
}
```

File: app/Models/Review.php

```

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;

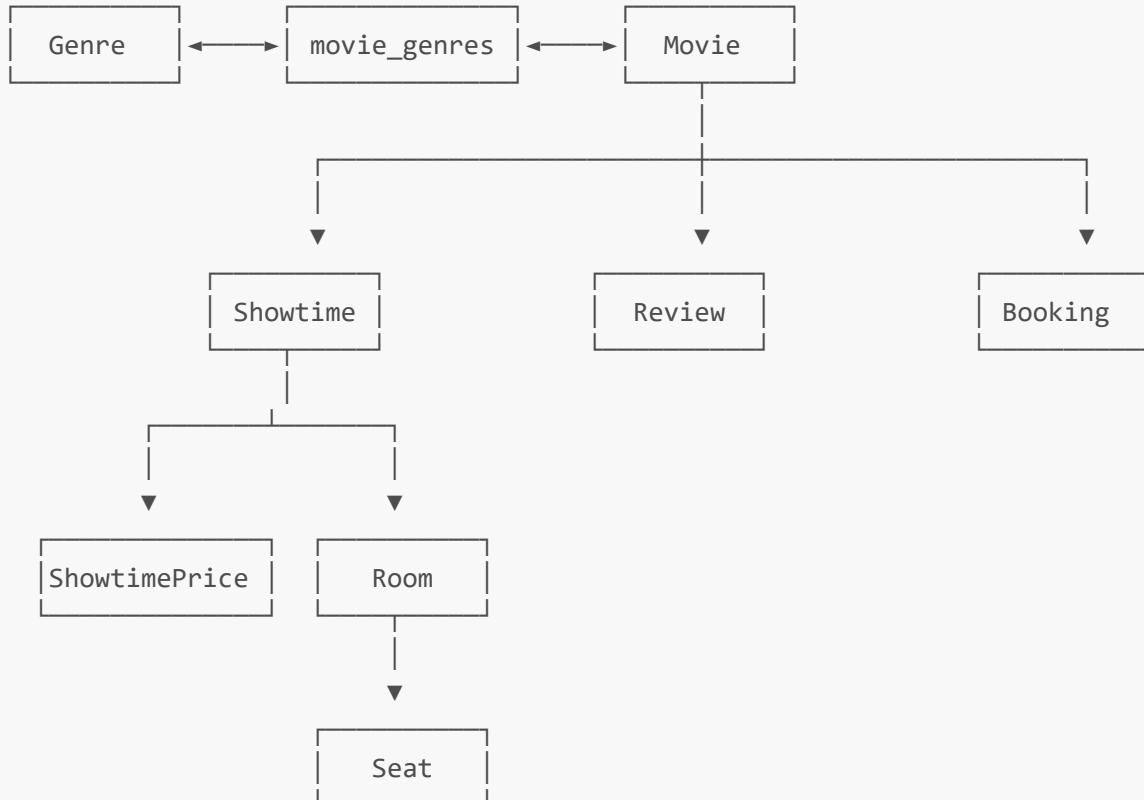
class Review extends Model
{
    protected $fillable = ['user_id', 'movie_id', 'rating', 'comment'];

    public function user()
    {
        return $this->belongsTo(User::class);
    }

    public function movie()
    {
        return $this->belongsTo(Movie::class);
    }
}

```

Sơ đồ quan hệ



4. Controllers

4.1. MovieController (Frontend)

File: app/Http/Controllers/MovieController.php

```
<?php

namespace App\Http\Controllers;

use App\Models\Movie;
use App\Models\Review;
use App\Models\Booking;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\DB;
use Carbon\Carbon;

class MovieController extends Controller
{
    /**
     * Trang chủ - Hiển thị 6 phim nổi bật
     */
    public function homepage()
    {
        // Lấy 6 phim đang chiếu, sắp xếp theo rating
        $movies = DB::table('movies')
            ->where('status', 'now_showing')
            ->orderByDesc('rating_avg')
            ->limit(6)
            ->get();

        // Gắn thể loại vào mỗi phim
        $this->attachGenresToMovies($movies);

        return view('homepage', compact('movies'));
    }

    /**
     * Danh sách tất cả phim
     */
    public function index()
    {
        $movies = DB::table('movies')
            ->orderByDesc('release_date')
            ->get();

        $this->attachGenresToMovies($movies);

        return view('index', compact('movies'));
    }

    /**
     * Chi tiết phim
     * - Hiển thị thông tin phim
     */
```

```
* - Danh sách reviews
* - Kiểm tra quyền review của user
*/
public function show($id)
{
    $movie = Movie::with(['genres', 'reviews.user'])->findOrFail($id);

    $canReview = false;
    $userId = session('user_id');

    if ($userId) {
        $user = \App\Models\User::find($userId);

        // Admin có thể review bất kỳ phim nào
        if ($user && $user->role === 'admin') {
            $canReview = !Review::where('user_id', $userId)
                ->where('movie_id', $id)
                ->exists();
        } else {
            // User thường phải đã xem phim (có booking đã thanh toán và suất
            // chiếu đã qua)
            $hasWatched = Booking::where('user_id', $userId)
                ->where('payment_status', 'paid')
                ->whereHas('showtime', function ($query) use ($id) {
                    $query->where('movie_id', $id)
                        ->where(function ($q) {
                            $q->where('show_date', '<', Carbon::today())
                                ->orWhere(function ($q2) {
                                    $q2->where('show_date', '=', Carbon::today())
                                        ->where('show_time', '<', Carbon::now()
                                            ->format('H:i:s')));
                                });
                            });
                })
                ->exists();
        }
        $alreadyReviewed = Review::where('user_id', $userId)
            ->where('movie_id', $id)
            ->exists();

        $canReview = $hasWatched && !$alreadyReviewed;
    }
}

return view('movie_details', compact('movie', 'canReview'));
}

/**
 * Phim đang chiếu
 */
public function nowShowing()
{
    $movies = DB::table('movies')
```

```

->where('status', 'now_showing')
->orderByDesc('rating_avg')
->get();

$this->attachGenresToMovies($movies);

return view('movie.now_showing', compact('movies'));
}

/**
 * Phim sắp chiếu
 */
public function upcomingMovies()
{
    $movies = DB::table('movies')
        ->where('status', 'coming_soon')
        ->orderBy('release_date')
        ->get();

$this->attachGenresToMovies($movies);

return view('movie.upcoming_movies', compact('movies'));
}

/**
 * Helper: Gắn thẻ loại vào danh sách phim
 */
private function attachGenresToMovies($movies)
{
    foreach ($movies as $movie) {
        $genres = DB::table('movie_genres')
            ->join('genres', 'movie_genres.genre_id', '=', 'genres.id')
            ->where('movie_genres.movie_id', $movie->id)
            ->pluck('genres.name')
            ->toArray();

        $movie->genres_list = implode(', ', $genres);
    }
}
}

```

4.2. ShowtimeController

File: app/Http/Controllers>ShowtimeController.php

```

<?php

namespace App\Http\Controllers;

use App\Models\Movie;
use App\Models>Showtime;

```

```
use Illuminate\Http\Request;

class ShowtimeController extends Controller
{
    /**
     * Hiển thị tất cả suất chiếu của một phim
     */
    public function showtimes($id)
    {
        $movie = Movie::findOrFail($id);

        // Lấy các suất chiếu từ hôm nay trở đi
        $showtimes = Showtime::with(['room.screenType'])
            ->where('movie_id', $id)
            ->where('show_date', '>=', now()->toDateString())
            ->orderBy('show_date')
            ->orderBy('show_time')
            ->get();

        return view('movie.showtimes', compact('movie', 'showtimes'));
    }
}
```

4.3. AdminMovieController

File: app/Http/Controllers/Admin/AdminMovieController.php

```
<?php

namespace App\Http\Controllers\Admin;

use App\Http\Controllers\Controller;
use App\Models\Movie;
use App\Models\Genre;
use Illuminate\Http\Request;

class AdminMovieController extends Controller
{
    /**
     * Danh sách phim (Admin)
     */
    public function index()
    {
        $movies = Movie::with('genres')
            ->orderByDesc('id')
            ->paginate(20);

        return view('admin.movies.index', compact('movies'));
    }

    /**

```

```
* Form tạo phim mới
*/
public function create()
{
    $genres = Genre::orderBy('name')->get();
    return view('admin.movies.create', compact('genres'));
}

/**
 * Lưu phim mới
 */
public function store(Request $request)
{
    $validated = $request->validate([
        'title' => 'required|string|max:255',
        'duration' => 'required|integer|min:1',
        'release_date' => 'required|date',
        'status' => 'required|in:now_showing,coming_soon,ended',
        'genres' => 'nullable|array',
        'genres.*' => 'exists:genres,id',
        'poster_url' => 'nullable|url',
        'trailer_url' => 'nullable|url',
        'director' => 'nullable|string|max:255',
        'cast' => 'nullable|string',
        'language' => 'nullable|string|max:100',
        'age_rating' => 'nullable|string|max:10',
        'description' => 'nullable|string',
    ]);
}

$movie = Movie::create($validated);

// Sync genres (Many-to-Many)
if (!empty($validated['genres'])) {
    $movie->genres()->sync($validated['genres']);
}

return redirect()->route('admin.movies.index')
    ->with('success', 'Movie created successfully!');
}

/**
 * Form chỉnh sửa phim
 */
public function edit(Movie $movie)
{
    $genres = Genre::orderBy('name')->get();
    $selectedGenres = $movie->genres->pluck('id')->toArray();

    return view('admin.movies.edit', compact('movie', 'genres',
    'selectedGenres'));
}

/**
 * Cập nhật phim
*/
```

```

    */
    public function update(Request $request, Movie $movie)
    {
        $validated = $request->validate([
            'title' => 'required|string|max:255',
            'duration' => 'required|integer|min:1',
            'release_date' => 'required|date',
            'status' => 'required|in:now_showing,coming_soon,ended',
            'genres' => 'nullable|array',
            'genres.*' => 'exists:genres,id',
            // ... các trường khác
        ]);

        $movie->update($validated);

        // Sync genres
        $movie->genres()->sync($validated['genres'] ?? []);

        return redirect()->route('admin.movies.index')
            ->with('success', 'Movie updated successfully!');
    }

    /**
     * Xóa phim
     */
    public function destroy(Movie $movie)
    {
        // Detach all genres trước khi xóa
        $movie->genres()->detach();
        $movie->delete();

        return redirect()->route('admin.movies.index')
            ->with('success', 'Movie deleted successfully!');
    }
}

```

5. Views

5.1. Homepage ([resources/views/homepage.blade.php](#))

Chức năng: Hiển thị 6 phim nổi bật trên trang chủ

```

@extends('layouts.main')

@section('content')
<div class="container py-5">
    <!-- Hero Section -->
    <div class="hero-section text-center mb-5">
        <h1>Welcome to CineBook</h1>
        <p>Book your favorite movies now!</p>

```

```
<!-- Search Bar -->
<form action="{{ route('search') }}" method="GET" class="search-form">
    <input type="text" name="query" placeholder="Search movies...">
    <button type="submit"><i class="bi bi-search"></i></button>
</form>
</div>

<!-- Featured Movies -->
<section class="featured-movies">
    <div class="section-header d-flex justify-content-between">
        <h2>Now Showing</h2>
        <a href="{{ route('movies.index') }}">View All</a>
    </div>

    <div class="row">
        @foreach($movies as $movie)
        <div class="col-md-4 col-lg-2 mb-4">
            <div class="movie-card">
                <!-- Poster -->
                <div class="movie-poster">
                    title }}"/>
                </div>

                <!-- Overlay với buttons -->
                <div class="movie-overlay">
                    <a href="{{ route('movies.show', $movie->id) }}" class="btn btn-details">
                        View Details
                    </a>
                    <a href="{{ route('movies.showtimes', $movie->id) }}" class="btn btn-book">
                        Book Now
                    </a>
                </div>
            </div>
        </div>
    </div>

    <!-- Movie Info -->
    <div class="movie-info">
        <h5 class="movie-title">{{ $movie->title }}</h5>
        <p class="movie-genres">{{ $movie->genres_list }}</p>
        <div class="movie-meta">
            <span class="duration">
                <i class="bi bi-clock"></i> {{ $movie->duration }} min
            </span>
            <span class="rating">
                <i class="bi bi-star-fill"></i> {{ $movie->rating_avg }}
            </span>
        </div>
    </div>
</div>
```

```

        @endforeach
    </div>
</section>
</div>
@endsection

```

5.2. Movie Details ([resources/views/movie_details.blade.php](#))

Chức năng: Hiển thị chi tiết phim, reviews, và form đánh giá

```

@extends('layouts.main')

@section('content')


![{{ $movie->title }}]({{ $movie->poster_url }})



@if($movie->status === 'now_showing')

Book Now

@endif

@if($movie->trailer_url)

Watch Trailer

@endif



# {{ $movie->title }}


@if($movie->status === 'now_showing')
Now Showing
@elseif($movie->status === 'coming_soon')
Coming Soon
@else
Ended
@endif


```

```
<!-- Rating -->
<div class="rating-display my-3">
    @for($i = 1; $i <= 5; $i++)
        @if($i <= $movie->rating_avg)
            <i class="bi bi-star-fill text-warning"></i>
        @else
            <i class="bi bi-star text-warning"></i>
        @endif
    @endfor
    <span class="ms-2">{{ $movie->rating_avg }}/5</span>
</div>

<!-- Movie Details -->
<table class="table table-borderless">
    <tr>
        <th>Director:</th>
        <td>{{ $movie->director }}</td>
    </tr>
    <tr>
        <th>Cast:</th>
        <td>{{ $movie->cast }}</td>
    </tr>
    <tr>
        <th>Duration:</th>
        <td>{{ $movie->duration }} minutes</td>
    </tr>
    <tr>
        <th>Language:</th>
        <td>{{ $movie->language }}</td>
    </tr>
    <tr>
        <th>Age Rating:</th>
        <td>{{ $movie->age_rating }}</td>
    </tr>
    <tr>
        <th>Release Date:</th>
        <td>{{ $movie->release_date->format('d/m/Y') }}</td>
    </tr>
    <tr>
        <th>Genres:</th>
        <td>
            @foreach($movie->genres as $genre)
                <span class="badge bg-info">{{ $genre->name }}</span>
            @endforeach
        </td>
    </tr>
</table>

<!-- Synopsis -->
<h4>Synopsis</h4>
<p>{{ $movie->description }}</p>
</div>
</div>
```

```
<!-- Reviews Section -->
<section class="reviews-section mt-5">
    <h3>Reviews ({{ $movie->reviews->count() }})</h3>

    <!-- Review Form (nếu user có quyền) -->
    @if($canReview)
        <div class="review-form card p-4 mb-4">
            <h5>Write a Review</h5>
            <form action="{{ route('reviews.store') }}" method="POST">
                @csrf
                <input type="hidden" name="movie_id" value="{{ $movie->id }}>

                <!-- Star Rating -->
                <div class="mb-3">
                    <label>Your Rating:</label>
                    <div class="star-rating">
                        @for($i = 5; $i >= 1; $i--)
                            <input type="radio" name="rating" value="{{ $i }}"
id="star{{ $i }}">
                            <label for="star{{ $i }}"><i class="bi bi-star-fill">
</i></label>
                    @endfor
                </div>
            </div>

                <!-- Comment -->
                <div class="mb-3">
                    <textarea name="comment" class="form-control" rows="4"
placeholder="Share your thoughts...">
maxlength="1000"></textarea>
                </div>

                <button type="submit" class="btn btn-primary">Submit
Review</button>
            </form>
        </div>
    @endif

    <!-- Reviews List -->
    <div class="reviews-list">
        @forelse($movie->reviews as $review)
            <div class="review-item card p-3 mb-3">
                <div class="d-flex justify-content-between">
                    <div>
                        <strong>{{ $review->user->name }}</strong>
                        <div class="stars">
                            @for($i = 1; $i <= 5; $i++)
                                @if($i <= $review->rating)
                                    <i class="bi bi-star-fill text-warning"></i>
                                @else
                                    <i class="bi bi-star text-warning"></i>
                                @endif
                            @endfor
                        </div>
                    <div>
                        <strong>{{ $review->rating }}</strong>
                        <p>{{ $review->content }}</p>
                    </div>
                </div>
            </div>
        @forelse
    </div>
```

```
        </div>
    </div>
    <small class="text-muted">{{ $review->created_at->diffForHumans() }}</small>
    </div>
    <p class="mt-2">{{ $review->comment }}</p>

    <!-- Edit/Delete buttons cho owner hoặc admin -->
    @if(session('user_id') == $review->user_id || session('user_role') == 'admin')
        <div class="review-actions">
            <a href="{{ route('reviews.edit', $review->id) }}" class="btn btn-sm btn-outline-primary">Edit</a>
            <form action="{{ route('reviews.destroy', $review->id) }}" method="POST" class="d-inline">
                @csrf
                @method('DELETE')
                <button type="submit" class="btn btn-sm btn-outline-danger" onclick="return confirm('Delete this review?')">Delete</button>
            </form>
        </div>
        @endif
    </div>
    @empty
        <p class="text-muted">No reviews yet. Be the first to review!</p>
    @endforelse
</div>
</section>
</div>

<!-- Trailer Modal -->
<div class="modal fade" id="trailerModal">
    <div class="modal-dialog modal-lg">
        <div class="modal-content">
            <div class="modal-header">
                <h5>{{ $movie->title }} - Trailer</h5>
                <button type="button" class="btn-close" data-bs-dismiss="modal">
            </button>
            </div>
            <div class="modal-body">
                <div class="ratio ratio-16x9">
                    <iframe src="{{ $movie->trailer_url }}" allowfullscreen>
                </div>
            </div>
        </div>
    </div>
</div>
@endsection
```

5.3. Now Showing ([resources/views/movie/now_showing.blade.php](#))

```
@extends('layouts.main')

@section('content')


<div class="d-flex justify-content-between align-items-center mb-4">
        <h2>Now Showing</h2>
        <a href="{{ route('upcoming') }}" class="btn btn-outline-primary">
            View Upcoming Movies
        </a>
    </div>

    <div class="row">
        @forelse($movies as $movie)
            <div class="col-md-3 mb-4">
                <div class="movie-card">
                    <div class="movie-poster position-relative">
                        title }}"/>
                        <span class="badge bg-success position-absolute top-0 end-0 m-2">
                            Now Showing
                        </span>
                    </div>
                    <div class="movie-info p-3">
                        <h5>{{ $movie->title }}</h5>
                        <p class="text-muted small">{{ $movie->genres_list }}</p>
                        <div class="d-flex justify-content-between">
                            <span><i class="bi bi-clock"></i> {{ $movie->duration }} min</span>
                            <span><i class="bi bi-star-fill text-warning"></i> {{ $movie->rating_avg }}</span>
                        </div>
                        <a href="{{ route('movies.show', $movie->id) }}" class="btn btn-primary w-100 mt-2">
                            View Details
                        </a>
                    </div>
                </div>
            </div>
        @empty
            <div class="col-12">
                <p class="text-muted text-center">No movies currently showing.</p>
            </div>
        @endforelse
    </div>


```

5.4. Showtimes ([resources/views/movie/showtimes.blade.php](#))

```
@extends('layouts.main')

@section('content')


<!-- Movie Header -->
    <div class="movie-header d-flex mb-4">
        title }}"
style="width: 100px;">
        <div class="ms-3">
            <h2>{{ $movie->title }}</h2>
            <p>{{ $movie->duration }} min | {{ $movie->language }}</p>
        </div>
    </div>

    <h3>Available Showtimes</h3>

    @if($showtimes->count() > 0)
        <div class="table-responsive">
            <table class="table table-hover">
                <thead>
                    <tr>
                        <th>Date</th>
                        <th>Time</th>
                        <th>Room</th>
                        <th>Screen Type</th>
                        <th>Action</th>
                    </tr>
                </thead>
                <tbody>
                    @foreach($showtimes as $showtime)
                    <tr>
                        <td>{{ $showtime->show_date->format('D, d M Y') }}</td>
                        <td>{{ Carbon\Carbon::parse($showtime->show_time)-
>format('H:i') }}</td>
                        <td>{{ $showtime->room->name }}</td>
                        <td>{{ $showtime->room->screenType->name }}</td>
                        <td>
                            <a href="{{ route('booking.seatmap', $showtime->id) }}"
                                class="btn btn-sm btn-primary">
                                Select Seats
                            </a>
                        </td>
                    </tr>
                    @endforeach
                </tbody>
            </table>
        </div>
    @else
        <div class="alert alert-info">
            No showtimes available for this movie.
        </div>
    @endif


```

```
<a href="{{ route('movies.show', $movie->id) }}" class="btn btn-outline-secondary mt-3">
    <i class="bi bi-arrow-left"></i> Back to Movie Details
</a>
</div>
@endsection
```

6. Routes

File: routes/web.php

```
<?php

use Illuminate\Support\Facades\Route;
use App\Http\Controllers\MovieController;
use App\Http\Controllers>ShowtimeController;
use App\Http\Controllers\BookingController;
use App\Http\Controllers\SearchController;
use App\Http\Controllers\ReviewController;
use App\Http\Controllers\Admin\AdminMovieController;
use App\Http\Controllers\Admin\AdminShowtimeController;

/*
| -----
| Public Routes - Movie Display
| -----
*/

// Homepage
Route::get('/', [MovieController::class, 'homepage'])->name('homepage');

// Movie List
Route::get('/index', [MovieController::class, 'index'])->name('movies.index');

// Movie Categories
Route::get('/now-showing', [MovieController::class, 'nowShowing'])->name('nowShowing');
Route::get('/upcoming-movies', [MovieController::class, 'upcomingMovies'])->name('upcoming');

// Movie Details
Route::get('/movies/{id}', [MovieController::class, 'show'])->name('movies.show');

// Movie Showtimes
Route::get('/movies/{id}/showtimes', [ShowtimeController::class, 'showtimes'])->name('movies.showtimes');

// Search
Route::get('/search', [SearchController::class, 'search'])->name('search');
```

```

/*
| -----
| Protected Routes - Booking (requires login)
| -----
*/

Route::middleware(['auth.check'])->group(function () {
    // Seat Selection
    Route::get('/showtimes/{showtime_id}/seats', [BookingController::class,
'seatMap'])
        ->name('booking.seatmap');

    // Book Seats
    Route::post('/showtimes/{showtime_id}/book', [BookingController::class,
'bookSeats'])
        ->name('booking.book');

    // Booking Confirmation
    Route::get('/booking/confirm/{booking_id}', [BookingController::class,
'confirmBooking'])
        ->name('booking.confirm');

    // Reviews
    Route::post('/reviews', [ReviewController::class, 'store'])
        ->name('reviews.store');
    Route::get('/reviews/{id}/edit', [ReviewController::class, 'edit'])
        ->name('reviews.edit');
    Route::put('/reviews/{id}', [ReviewController::class, 'update'])
        ->name('reviews.update');
    Route::delete('/reviews/{id}', [ReviewController::class, 'destroy'])
        ->name('reviews.destroy');
});

/*
| -----
| Admin Routes
| -----
*/

Route::prefix('admin')->middleware(['auth.check', 'admin'])->name('admin.')->group(function () {

    // Movies Management
    Route::get('/movies', [AdminMovieController::class, 'index'])
        ->name('movies.index');
    Route::get('/movies/create', [AdminMovieController::class, 'create'])
        ->name('movies.create');
    Route::post('/movies', [AdminMovieController::class, 'store'])
        ->name('movies.store');
    Route::get('/movies/{movie}/edit', [AdminMovieController::class, 'edit'])
        ->name('movies.edit');
    Route::put('/movies/{movie}', [AdminMovieController::class, 'update'])
        ->name('movies.update');
    Route::delete('/movies/{movie}', [AdminMovieController::class, 'destroy'])
        ->name('movies.destroy');
});

```

```

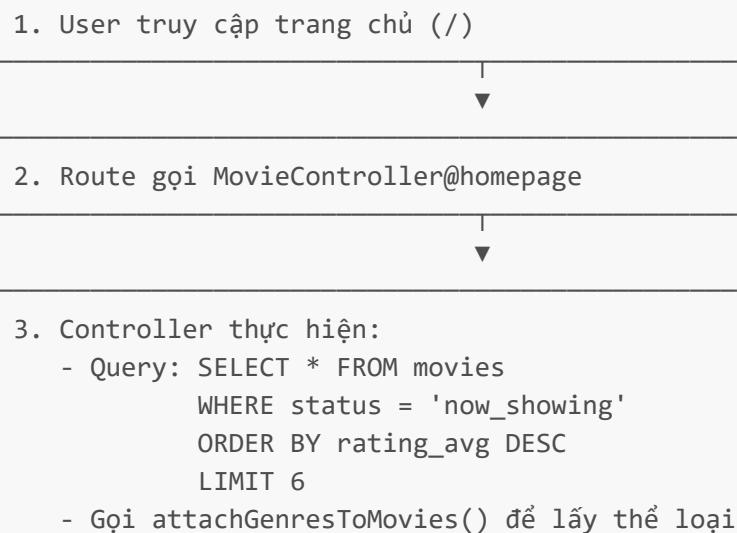
>name('movies.destroy');

    // Showtimes Management
    Route::get('/showtimes', [AdminShowtimeController::class, 'index'])-
>name('showtimes.index');
    Route::get('/showtimes/create', [AdminShowtimeController::class, 'create'])-
>name('showtimes.create');
    Route::post('/showtimes', [AdminShowtimeController::class, 'store'])-
>name('showtimes.store');
    Route::get('/showtimes/{showtime}/edit', [AdminShowtimeController::class,
'edit'])->name('showtimes.edit');
    Route::put('/showtimes/{showtime}', [AdminShowtimeController::class,
'update'])->name('showtimes.update');
    Route::delete('/showtimes/{showtime}', [AdminShowtimeController::class,
'destroy'])->name('showtimes.destroy');
});

```

7. Luồng hoạt động chi tiết

7.1. Luồng hiển thị phim trên Homepage



7.2. Luồng xem chi tiết phim

1. User click "View Details" trên movie card
URL: /movies/{id}



2. Route gọi MovieController@show(\$id)



3. Controller thực hiện:

a) Lấy phim với genres và reviews:
`$movie = Movie::with(['genres', 'reviews.user'])->findOrFail($id);`

b) Kiểm tra quyền review của user:

- Nếu chưa login: \$canReview = false
- Nếu là admin: có thể review nếu chưa review
- Nếu là user: phải có booking đã thanh toán
VÀ suất chiếu đã qua VÀ chưa review



4. Return view('movie_details', compact('movie', 'canReview'))



5. View movie_details.blade.php:

- Hiển thị poster, thông tin chi tiết
- Nút Book Now (nếu now_showing)
- Nút Watch Trailer
- Danh sách reviews
- Form review (nếu \$canReview = true)

7.3. Luồng kiểm tra quyền review

Điều kiện để user được phép review phim:

ADMIN:

1. User có role = 'admin'
2. Chưa review phim này trước đó
→ Được phép review

USER THƯỜNG:

1. Đã có booking với payment_status = 'paid'
2. Booking đó thuộc về showtime của phim này
3. Showtime đã diễn ra (show_date < today
HOẶC show_date = today VÀ show_time < now)
4. Chưa review phim này trước đó
→ Được phép review

8. Logic phân loại phim

8.1. Ba trạng thái phim

Status	Mô tả	Hiển thị	Có thể đặt vé
now_showing	Đang chiếu	Homepage, Now Showing	Có
coming_soon	Sắp chiếu	Upcoming Movies	Không
ended	Đã kết thúc	Chỉ trong admin	Không

8.2. Query theo trạng thái

```
// Phim đang chiếu (Homepage, Now Showing)
Movie::where('status', 'now_showing')
    ->orderByDesc('rating_avg')
    ->get();

// Phim sắp chiếu (Upcoming)
Movie::where('status', 'coming_soon')
    ->orderBy('release_date')
    ->get();

// Tất cả phim (Admin)
Movie::orderByDesc('id')
    ->paginate(20);
```

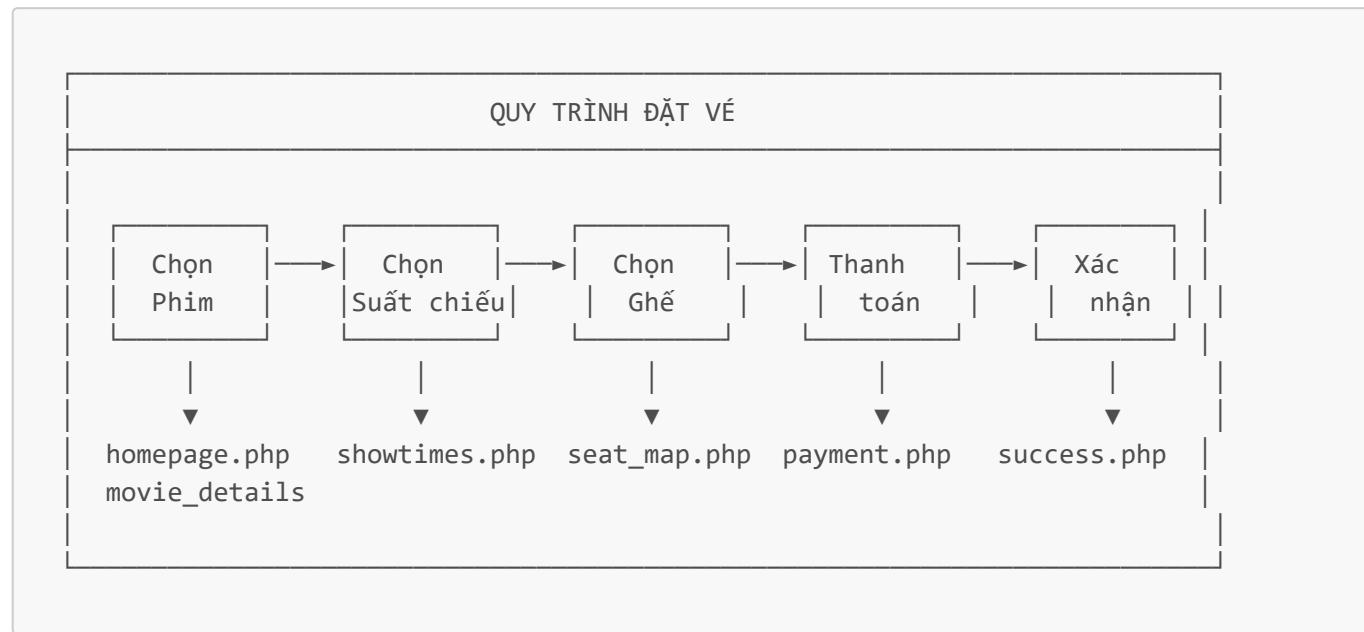
8.3. Hiển thị badge theo trạng thái

```
@if($movie->status === 'now_showing')
    <span class="badge bg-success">Now Showing</span>
@elseif($movie->status === 'coming_soon')
    <span class="badge bg-warning text-dark">Coming Soon</span>
@else
```

```
<span class="badge bg-secondary">Ended</span>
@endif
```

9. Quy trình đặt vé

9.1. Sơ đồ tổng quan



9.2. Chi tiết từng bước

Bước 1: Chọn phim

- User xem danh sách phim trên Homepage hoặc Now Showing
- Click "View Details" để xem chi tiết
- Click "Book Now" để đặt vé

Bước 2: Chọn suất chiếu

- Hiển thị tất cả suất chiếu của phim
- Thông tin: Ngày, Giờ, Phòng, Loại màn hình
- User chọn suất chiếu phù hợp

Bước 3: Chọn ghế

- Hiển thị sơ đồ ghế của phòng chiếu
- Màu sắc phân biệt:
 - Xanh dương: Standard
 - Vàng: VIP
 - Đỏ/Hồng: Couple
 - Xám: Đã đặt
- User click để chọn ghế
- Hiển thị tổng tiền real-time

Bước 4: Thanh toán

- Xác nhận thông tin đặt vé
- Chọn phương thức thanh toán
- Xử lý thanh toán (mock)

Bước 5: Xác nhận

- Hiển thị mã booking
- Tạo QR code cho mỗi vé
- Gửi email xác nhận (nếu có)

9.3. Tính giá vé

```
// Công thức tính giá vé
$ticketPrice = $seatType->base_price          // Giá cơ bản theo loại ghế
            + $screenType->price           // Phụ phí loại màn hình
            + $showtimePrice->price;      // Giá điều chỉnh theo suất chiếu

// Ví dụ:
// Standard seat: 80,000 VND
// 3D screen: +40,000 VND
// Prime time adjustment: +10,000 VND
// Total: 130,000 VND
```

10. Các file liên quan

Models

- app/Models/Movie.php
- app/Models/Genre.php
- app/Models>Showtime.php
- app/Models>ShowtimePrice.php
- app/Models>ShowtimeSeat.php
- app/Models/Room.php
- app/Models/Seat.php
- app/Models/SeatType.php
- app/Models/ScreenType.php
- app/Models/Review.php
- app/Models/Booking.php
- app/Models/BookingSeat.php

Controllers

- app/Http/Controllers/MovieController.php
- app/Http/Controllers>ShowtimeController.php
- app/Http/Controllers/BookingController.php

- app/Http/Controllers/SearchController.php
- app/Http/Controllers/ReviewController.php
- app/Http/Controllers/Admin/AdminMovieController.php
- app/Http/Controllers/Admin/AdminShowtimeController.php

Views (Frontend)

- resources/views/homepage.blade.php
- resources/views/index.blade.php
- resources/views/movie_details.blade.php
- resources/views/movie/now_showing.blade.php
- resources/views/movie/upcoming_movies.blade.php
- resources/views/movie/showtimes.blade.php
- resources/views/booking/seat_map.blade.php
- resources/views/booking/confirm.blade.php
- resources/views/booking/success.blade.php
- resources/views/search_results.blade.php

Views (Admin)

- resources/views/admin/movies/index.blade.php
- resources/views/admin/movies/create.blade.php
- resources/views/admin/movies/edit.blade.php
- resources/views/admin/showtimes/index.blade.php
- resources/views/admin/showtimes/create.blade.php
- resources/views/admin/showtimes/edit.blade.php

Routes

- routes/web.php

Database

- mySQL/data.sql
-

Ghi chú thêm

Timestamps trong Models

Một số model không sử dụng timestamps (`created_at`, `updated_at`):

```
public $timestamps = false;
```

Các model cần setting này:

- Seat
- ShowtimePrice

- `ShowtimeSeat`
- `Genre`
- `Showtime`

Helper Functions

- `attachGenresToMovies($movies)`: Gắn thể loại vào collection phim khi dùng `DB::table()`
- `updateAverageRating()`: Cập nhật điểm trung bình của phim từ reviews

Session Variables

- `session('user_id')`: ID người dùng đang đăng nhập
- `session('user_role')`: Role của người dùng ('admin' hoặc 'user')

Tài liệu được tạo tự động bởi Claude Code - Cập nhật: {{ date('Y-m-d') }}