

PODCAST: ADMIN PANEL CINEBOOK

PHẦN 4: BUSINESS LOGIC VÀ CHUẨN BỊ THUYẾT TRÌNH

Thời lượng ước tính: 35-40 phút

Microphone icon **PHẦN CUỐI CÙNG**

Xin chào các bạn! Đây là phần cuối cùng của series. Trong ba phần trước, chúng ta đã đi từ kiến trúc tổng quan, qua các thuật toán kỹ thuật, đến thiết kế giao diện. Hôm nay, mình sẽ tổng hợp lại qua góc nhìn business - nghiệp vụ thực tế của rạp chiếu phim.

Và quan trọng hơn, mình sẽ giúp các bạn chuẩn bị cho việc thuyết trình. Những câu hỏi nào có thể gặp? Trả lời như thế nào cho thuyết phục?

Bắt đầu thôi!

Briefcase icon **PHẦN 1: NGHIỆP VỤ RẠP CHIẾU PHIM - "Hiểu để xây dựng"**

Trước khi xây dựng bất kỳ hệ thống nào, chúng ta cần hiểu nghiệp vụ. Rạp chiếu phim hoạt động như thế nào?

Các thực thể chính:

Đầu tiên là Phim. Rạp có kho phim, mỗi phim có thông tin riêng: tên, đạo diễn, diễn viên, thời lượng, thể loại, trạng thái - đang chiếu, sắp chiếu, hay đã ngưng.

Thứ hai là Phòng chiếu. Rạp có nhiều phòng, mỗi phòng có sơ đồ ghế riêng, loại màn hình riêng. Phòng IMAX khác phòng 2D thường.

Thứ ba là Suất chiếu. Đây là sự kết hợp: Phim X chiếu ở Phòng Y, vào Ngày Z, lúc Giờ T. Mỗi suất chiếu có giá vé riêng.

Thứ tư là Khách hàng. Họ đăng ký tài khoản, đặt vé, thanh toán.

Thứ năm là Đơn đặt vé. Kết nối khách hàng với suất chiếu, ghi nhận ghế nào, giá bao nhiêu, trạng thái thanh toán.

Luồng nghiệp vụ chính:

Nghiệp vụ một: Thêm phim mới. Phim mới ra mắt, admin thêm vào hệ thống với trạng thái "Sắp chiếu". Đến ngày ra mắt, đổi thành "Đang chiếu". Hết mùa chiếu, đổi thành "Đã kết thúc".

Nghiệp vụ hai: Xếp lịch chiếu. Admin xem lịch, chọn phòng còn trống, tạo suất chiếu mới. Hệ thống tự động kiểm tra không trùng lịch.

Nghiệp vụ ba: Đặt vé. Khách hàng chọn phim, chọn suất, chọn ghế, thanh toán. Admin theo dõi đơn hàng, xử lý các trường hợp đặc biệt.

Nghệp vụ bốn: Check-in. Khách đến rạp, đưa mã QR. Nhân viên quét, xác nhận, cho vào phòng.

Nghệp vụ năm: Báo cáo. Admin xem doanh thu theo ngày, tuần, tháng. Xem phim nào bán chạy, phim nào ế.

🔒 PHẦN 2: BẢO MẬT - "Khóa chặt hệ thống"

Bảo mật là chủ đề quan trọng mà hội đồng thường hỏi. Hãy cùng hiểu các lớp bảo mật.

Lớp 1: Xác thực - Bạn là ai?

Trước khi làm bất cứ điều gì, hệ thống hỏi "Bạn là ai?" Nếu chưa đăng nhập, bạn bị đuổi về trang login.

Mật khẩu không được lưu dạng gốc. Thay vào đó, hệ thống lưu "hash" - một chuỗi ký tự được tạo từ mật khẩu bằng thuật toán một chiều. Từ hash không thể suy ngược ra mật khẩu gốc. Khi đăng nhập, hệ thống hash mật khẩu nhập vào, so sánh với hash đã lưu.

Lớp 2: Phân quyền - Bạn được làm gì?

Đăng nhập rồi, nhưng bạn có quyền vào trang admin không? User thường không được. Chỉ admin mới được.

Hệ thống kiểm tra trường "role" trong thông tin người dùng. Nếu role là "admin", cho vào. Nếu là "user", từ chối với lỗi 403.

Lớp 3: CSRF - Chống giả mạo yêu cầu

Đây là loại tấn công tinh vi. Kẻ xấu tạo một trang web giả, nhúng một đường link ẩn đến hệ thống của bạn - ví dụ link xóa tài khoản. Nếu admin vô tình vào trang đó trong khi đang đăng nhập hệ thống, trình duyệt sẽ gửi yêu cầu xóa tài khoản với cookie của admin.

Giải pháp là CSRF token. Mỗi form có một token bí mật, do server tạo ra. Khi form được submit, server kiểm tra token. Nếu token không khớp, từ chối. Trang web giả không biết token, nên không thể giả mạo.

Lớp 4: Validation - Kiểm tra dữ liệu

Không bao giờ tin tưởng dữ liệu từ người dùng. Mọi input đều được kiểm tra:

- Trường bắt buộc không được bỏ trống
- Email phải đúng định dạng email
- Số phải là số, không phải chữ
- Link phải là URL hợp lệ
- Giá trị phải nằm trong danh sách cho phép

Nếu dữ liệu không hợp lệ, từ chối ngay, không lưu vào database.

Lớp 5: Chống SQL Injection

SQL Injection là kỹ thuật tấn công bằng cách chèn mã độc vào input. Ví dụ, thay vì nhập tên phim, kẻ xấu nhập một đoạn lệnh xóa toàn bộ database.

Hệ thống ngăn chặn bằng cách không bao giờ ghép trực tiếp input vào câu lệnh SQL. Thay vào đó, dùng "prepared statements" - câu lệnh được chuẩn bị sẵn với các vị trí giữ chỗ. Input được đưa vào riêng, được xử lý an toàn trước.

💡 PHẦN 3: XỬ LÝ TÌNH HUỐNG ĐẶC BIỆT - "Khi mọi thứ không như kế hoạch"

Hệ thống tốt không chỉ hoạt động khi mọi thứ suôn sẻ, mà còn xử lý được các tình huống bất thường.

Tình huống 1: Hai người cùng đặt một ghế

Khách A và khách B cùng chọn ghế C5. Cả hai cùng bấm "Đặt vé" gần như cùng lúc. Chuyện gì xảy ra?

Giải pháp: Khi khách chọn ghế và bấm đặt, hệ thống "giữ" ghế đó ngay lập tức, đổi trạng thái thành "đang giữ". Nếu khách B bấm sau, hệ thống thấy ghế đã bị giữ, báo lỗi "Ghế này vừa được người khác chọn."

Kỹ thuật sử dụng là "locking" ở mức database - chỉ một request được cập nhật ghế tại một thời điểm.

Tình huống 2: Khách chọn ghế rồi bỏ ngang

Khách chọn 3 ghế, ghế chuyển sang "đang giữ". Nhưng khách đóng trình duyệt, không thanh toán. Ghế bị "khóa" mãi mãi?

Giải pháp: Mỗi đơn đặt vé "pending" có thời hạn 15 phút. Sau 15 phút, hệ thống tự động chuyển đơn thành "expired" và trả ghế về trạng thái "available".

Có thể thực hiện bằng cách chạy một job định kỳ mỗi phút, kiểm tra các đơn pending quá hạn.

Tình huống 3: Thanh toán thất bại

Khách bấm thanh toán, nhưng ngân hàng từ chối - thẻ hết tiền, hoặc lỗi mạng.

Giải pháp: Không xóa đơn ngay. Giữ đơn ở trạng thái "pending", cho phép khách thử lại. Hiển thị thông báo lỗi rõ ràng và hướng dẫn. Nếu hết 15 phút vẫn chưa thanh toán được, đơn mới bị hủy.

Tình huống 4: Admin xóa phim đang có suất chiếu

Admin muốn xóa phim, nhưng phim đó đang có suất chiếu, suất chiếu đang có đơn đặt vé.

Giải pháp: Hệ thống kiểm tra trước khi xóa. Nếu phim còn liên quan đến dữ liệu khác, từ chối xóa và thông báo: "Không thể xóa phim này vì đang có suất chiếu. Hãy xóa suất chiếu trước."

Đây gọi là "referential integrity" - toàn vẹn tham chiếu.

Tình huống 5: Hai admin cùng sửa một phim

Admin A mở trang sửa phim "Avatar". Admin B cũng mở. A sửa và lưu. B sửa và lưu. Thay đổi của A bị ghi đè?

Giải pháp đơn giản nhất là "last write wins" - ai lưu sau thì thắng. Chấp nhận được nếu ít admin và ít xảy ra trùng.

Giải pháp tốt hơn là "optimistic locking" - khi lưu, kiểm tra xem dữ liệu có bị thay đổi kể từ lúc mở form không. Nếu có, báo lỗi "Dữ liệu đã được người khác cập nhật, vui lòng tải lại."

✍ PHẦN 4: CHUẨN BỊ THUYẾT TRÌNH - "Nói như chuyên gia"

Bây giờ đến phần thực hành. Mình sẽ đóng vai hội đồng, hỏi các câu hỏi. Các bạn học cách trả lời.

Câu hỏi: "Tại sao bạn chọn Laravel cho dự án này?"

Cách trả lời: "Em chọn Laravel vì ba lý do. Thứ nhất, Laravel có kiến trúc MVC rõ ràng, giúp code dễ tổ chức và bảo trì. Thứ hai, Laravel cung cấp sẵn nhiều tính năng quan trọng như authentication, authorization, validation - em không cần viết lại từ đầu. Thứ ba, Laravel có cộng đồng lớn và documentation tốt, khi gặp vấn đề dễ tìm được hướng giải quyết."

Câu hỏi: "Giải thích về mô hình MVC?"

Cách trả lời: "MVC chia ứng dụng thành ba phần. Model xử lý dữ liệu - đọc, ghi database. View hiển thị giao diện cho người dùng. Controller nhận yêu cầu từ người dùng, điều phối giữa Model và View, rồi trả kết quả. Ví dụ khi admin thêm phim mới: Controller nhận dữ liệu từ form, gọi Model để lưu vào database, rồi gọi View để hiển thị thông báo thành công."

Câu hỏi: "Làm sao bạn bảo vệ trang admin?"

Cách trả lời: "Em dùng hai lớp middleware. Lớp đầu tiên kiểm tra đăng nhập - nếu chưa đăng nhập, chuyển về trang login. Lớp thứ hai kiểm tra quyền - nếu không phải admin, trả về lỗi 403 Forbidden. Tất cả route admin đều được bọc bởi hai middleware này, nên không thể vượt qua."

Câu hỏi: "Transaction là gì và tại sao cần dùng?"

Cách trả lời: "Transaction đảm bảo nhiều thao tác database xảy ra như một khối duy nhất. Hoặc tất cả thành công, hoặc không có gì xảy ra. Ví dụ khi tạo suất chiếu, em cần tạo bản ghi suất chiếu, tạo bảng giá, và tạo trạng thái cho 80 ghế. Nếu lỗi ở ghế thứ 50, transaction sẽ rollback tất cả - không để lại dữ liệu nửa vời."

Câu hỏi: "N+1 Query Problem là gì?"

Cách trả lời: "Đây là vấn đề hiệu năng khi lấy dữ liệu trong vòng lặp. Ví dụ, lấy 100 đơn đặt vé bằng một query, rồi với mỗi đơn, chạy một query nữa để lấy tên khách. Tổng cộng 101 query. Em giải quyết bằng Eager Loading - lấy đơn và thông tin khách trong cùng một câu query, chỉ tốn 2 query."

Câu hỏi: "Hệ thống QR check-in hoạt động như thế nào?"

Cách trả lời: "Khi khách thanh toán thành công, hệ thống tạo một mã QR unique bằng cách hash tổ hợp mã đơn, thông tin ghế, và timestamp. Mã này lưu vào database với trạng thái 'active'. Khi nhân viên quét QR, hệ thống kiểm tra: mã có tồn tại không, còn active không. Nếu hợp lệ, đổi trạng thái thành 'checked', ghi nhận thời gian check-in. Mã đã dùng không thể dùng lại."

Câu hỏi: "Tại sao dùng màu xanh lá cho 'thành công', đỏ cho 'lỗi'?"

Cách trả lời: "Đây là convention quốc tế - quy ước mà mọi người đều hiểu. Xanh lá như đèn giao thông xanh - đi được, OK. Đỏ như đèn đỏ - dừng lại, có vấn đề. Vàng là cảnh báo, chờ đợi. Dùng convention giúp người dùng hiểu ngay mà không cần đọc text."

⌚ PHẦN 5: KỊCH BẢN DEMO - "5 phút ấn tượng"

Khi demo, thời gian quý giá. Đây là kịch bản 5 phút hiệu quả.

Phút đầu - Giới thiệu và Dashboard:

"Đây là hệ thống Admin Panel cho rạp chiếu phim CineBook. Sau khi đăng nhập với tài khoản admin, chúng ta thấy Dashboard với các thống kê quan trọng: tổng user, tổng phim, vé bán hôm nay, doanh thu. Phía dưới là 10 đơn đặt vé gần nhất."

Phút hai - Thêm phim mới:

"Tiếp theo, em sẽ demo thêm phim mới. Click vào Quản lý Phim, rồi Thêm mới. Điền thông tin: tên phim, đạo diễn, thời lượng, chọn thể loại. Hệ thống validate ngay - nếu bỏ trống trường bắt buộc, sẽ báo lỗi. Bấm Lưu, thông báo thành công xuất hiện, phim đã có trong danh sách."

Phút ba - Tạo phòng chiếu với sơ đồ ghế:

"Đây là tính năng em tâm đắc. Vào Quản lý Phòng, Thêm mới. Nhập tên phòng, chọn loại màn hình, nhập số hàng và số ghế. Bấm Xem trước - sơ đồ ghế xuất hiện. Em có thể chọn template Cinema Style để tự động xếp ghế Standard trước, VIP giữa, Couple cuối. Hoặc click từng ghế để tùy chỉnh. Bấm Tạo phòng - 80 ghế được tạo cùng lúc nhờ Transaction."

Phút bốn - Quản lý đơn đặt vé:

"Vào Quản lý Đơn, chúng ta thấy danh sách đơn với các trạng thái khác nhau - badge màu xanh là đã xác nhận, vàng là đang chờ. Có thể lọc theo trạng thái, tìm kiếm theo tên khách. Click vào một đơn để xem chi tiết: thông tin khách, phim, ghế, giá tiền. Admin có thể hủy đơn nếu cần."

Phút năm - QR Check-in và tổng kết:

"Cuối cùng là QR Check-in. Nhân viên nhập mã QR, hệ thống hiển thị thông tin đặt vé để xác nhận. Bấm Check-in, mã được đánh dấu đã sử dụng. Tóm lại, hệ thống được xây dựng theo kiến trúc MVC, bảo mật hai lớp với authentication và authorization, sử dụng Transaction cho dữ liệu nhất quán, và thiết kế UX/UI trực quan. Cảm ơn thầy cô đã lắng nghe, em sẵn sàng trả lời câu hỏi."

💡 PHẦN 6: MẸO THUYẾT TRÌNH - "Bí quyết gây ấn tượng"

Nói chậm, rõ ràng:

Khi hồi hộp, chúng ta hay nói nhanh. Hãy cố gắng nói chậm hơn bình thường. Ngắt câu. Nhấn mạnh từ quan trọng.

Giải thích WHY, không chỉ WHAT:

Không chỉ nói "Em dùng Transaction." Mà nói "Em dùng Transaction để đảm bảo nếu có lỗi, không có dữ liệu nửa vời."

Dùng ví dụ cụ thể:

Thay vì nói trừu tượng, đưa ví dụ: "Ví dụ khi tạo suất chiếu, em cần tạo 3 bản ghi giá và 80 bản ghi trạng thái ghế..."

Thừa nhận hạn chế, đề xuất cải tiến:

Nếu hỏi đồng hỏi về điểm yếu, đừng né tránh. "Hiện tại em chưa có automated testing. Nếu có thêm thời gian, em sẽ viết unit tests và integration tests."

Không bịa khi không biết:

Nếu gặp câu hỏi không biết, nói thẳng: "Em chưa tìm hiểu sâu về vấn đề này, nhưng em sẽ research thêm."
Tốt hơn là bịa sai.

Chuẩn bị dữ liệu demo:

Trước khi demo, đảm bảo database có dữ liệu đẹp: vài bộ phim với poster, vài đơn đặt vé, vài user. Demo với database trống rất nhảm chán.

TÓM TẮT TOÀN BỘ SERIES

Qua bốn phần, chúng ta đã đi từ:

Phần 1 - Kiến trúc:

- MVC pattern chia code thành Model, View, Controller
- Middleware bảo vệ route với authentication và authorization
- Cấu trúc thư mục rõ ràng, dễ bảo trì

Phần 2 - Kỹ thuật:

- Quản lý phòng chiếu với sơ đồ ghế và các template
- Transaction đảm bảo toàn vẹn dữ liệu
- Công thức giá vé linh hoạt
- Vòng đời đơn đặt vé từ pending đến completed
- QR Check-in an toàn với hash SHA-256
- Eager Loading giải quyết N+1 problem

Phần 3 - UX/UI:

- Bốn nguyên tắc: Hiệu quả, Rõ ràng, Nhất quán, Phản hồi
- Dashboard với các thẻ thống kê
- Bảng dữ liệu, form, status badge
- Sơ đồ ghế như rạp thật
- Responsive cho mọi thiết bị

Phần 4 - Business & Thuyết trình:

- Luồng nghiệp vụ rạp chiếu phim
 - Các lớp bảo mật
 - Xử lý tình huống đặc biệt
 - Trả lời câu hỏi phỏng vấn
 - Kịch bản demo 5 phút
 - Mẹo thuyết trình
-

LỜI KẾT

Các bạn thân mến,

Xây dựng một hệ thống Admin Panel không chỉ là viết code. Đó là sự kết hợp của:

- Hiểu nghiệp vụ - biết rạp phim cần gì
- Thiết kế kiến trúc - tổ chức code hợp lý
- Kỹ năng kỹ thuật - database, algorithm, security
- Tư duy thiết kế - UX/UI trực quan
- Khả năng giao tiếp - giải thích cho người khác hiểu

Nếu các bạn nắm vững những gì trong series này, các bạn không chỉ hiểu về Admin Panel CineBook, mà còn có nền tảng để xây dựng bất kỳ hệ thống quản trị nào.

Chúc các bạn thuyết trình thành công!

HẾT SERIES

Danh sách files:

- [Phần 1: Tổng quan Kiến trúc](#)
- [Phần 2: Database và Thuật toán](#)
- [Phần 3: Thiết kế UX/UI](#)
- [Phần 4: Business Logic & Thuyết trình \(file hiện tại\)](#)