

# Đề tài: Xây dựng hệ thống quản lý bán và đặt sách trực tuyến sử dụng ReactJS và MySQL

## Slide 1: Bìa (Cover)

- **Tên đề tài:** *Xây dựng hệ thống quản lý bán và đặt sách trực tuyến sử dụng ReactJS và MySQL*
- **Sinh viên thực hiện:** Nguyễn Thành Đạt – MSSV: 20213104 – Lớp: DCCNTT12.10.11
- **Giảng viên hướng dẫn:** Th.S. Mai Văn Linh
- **Trường:** Đại học Công nghệ Đông Á (Bắc Ninh, 2025)

*(Slide mở đầu, giới thiệu đề tài, thông tin sinh viên và giáo viên hướng dẫn.)*

## Slide 2: Agenda (Nội dung chính)

1. **Bài toán & Bối cảnh:** Đặt vấn đề, thực trạng và mục tiêu của đề tài
2. **Phân tích & Thiết kế hệ thống:** Chức năng chính, mô hình, cơ sở dữ liệu
3. **Công nghệ sử dụng:** Công nghệ chính và tích hợp trong hệ thống
4. **Sản phẩm (Demo):** Minh họa giao diện người dùng và quản trị hệ thống
5. **Kết luận & Kiến nghị:** Kết quả đạt được, hạn chế và hướng phát triển
6. **Q&A:** Giải đáp thắc mắc (Slide cảm ơn kết thúc)

*(Slide Agenda liệt kê các phần chính trong bài trình bày để khán giả nắm được cấu trúc.)*

## Slide 3: Bài toán & Bối cảnh

- **Thực trạng:** Nhiều nhà sách vừa và nhỏ quản lý kho, đơn hàng thủ công (ghi chép tay, Excel) kém hiệu quả, dễ sai sót và chậm trễ <sup>1</sup>. Hệ thống quản lý chuyên nghiệp thì chi phí cao, nên các nhà sách này thiếu công cụ tự động hóa quy trình đặt sách và phân tích kinh doanh <sup>2</sup>.
- **Vấn đề:** Khách hàng mua sách gặp khó khăn khi tìm kiếm thông tin sách (giá cả, tồn kho, mô tả...) và không theo dõi được trạng thái đơn hàng minh bạch, dẫn đến trải nghiệm chưa tốt <sup>3</sup>. Sự bất tiện này làm giảm sự hài lòng và lòng tin của khách đối với nhà sách.

*(Trình bày bối cảnh và vấn đề đặt ra: các hạn chế của phương thức bán sách truyền thống và nhu cầu chuyển đổi số trong bán sách.)*

## Slide 4: Mục tiêu đề tài

- **Xây dựng nền tảng web thân thiện** giúp khách hàng dễ dàng **tìm kiếm, xem thông tin và đặt mua sách trực tuyến** mọi lúc mọi nơi <sup>4</sup>.
- **Cung cấp công cụ cho quản trị viên** quản lý hiệu quả: theo dõi **kho sách, xử lý đơn hàng, quản lý tài khoản khách hàng** và hỗ trợ **phân tích doanh thu** <sup>5</sup>.
- **Ứng dụng công nghệ hiện đại** (ReactJS, Spring Boot, MySQL...) để hệ thống **hoạt động ổn định, bảo mật và có khả năng mở rộng** trong tương lai <sup>6</sup>.

*(Nêu rõ mục tiêu cần đạt: xây dựng hệ thống đáp ứng nhu cầu người dùng và quản trị, đồng thời sử dụng công nghệ mới để đảm bảo hiệu quả và mở rộng được.)*

## Slide 5: Chức năng cho Khách hàng

- **Đăng ký, đăng nhập tài khoản:** Khách hàng tạo tài khoản và đăng nhập an toàn (hỗ trợ quên mật khẩu, xác thực bảo mật).
- **Tìm kiếm sách:** Tìm theo tên sách, tác giả, thể loại; có gợi ý từ khóa và hiển thị kết quả nhanh.
- **Xem chi tiết sách:** Hiển thị đầy đủ thông tin mỗi cuốn sách (tiêu đề, tác giả, nhà XB, năm XB, giá, số lượng tồn, mô tả nội dung, hình ảnh, đánh giá...).
- **Quản lý giỏ hàng:** Thêm sách vào giỏ, cập nhật số lượng, xóa sách; tự động tính tổng tiền (bao gồm phí vận chuyển nếu có) trước khi thanh toán <sup>7</sup> <sup>8</sup> .
- **Thanh toán trực tuyến:** Hỗ trợ thanh toán online qua cổng như Stripe, PayPal hoặc chuyển khoản ngân hàng – đảm bảo **nhANH chóng, an toàn** <sup>9</sup> .
- **Theo dõi đơn hàng:** Xem trạng thái đơn hàng (đang xử lý, đang giao, đã hoàn tất) và lịch sử mua hàng.
- **Đánh giá, nhận xét:** Sau khi nhận sách, khách hàng có thể đánh giá (xếp hạng sao) và bình luận về sách, giúp người dùng khác tham khảo <sup>10</sup> .

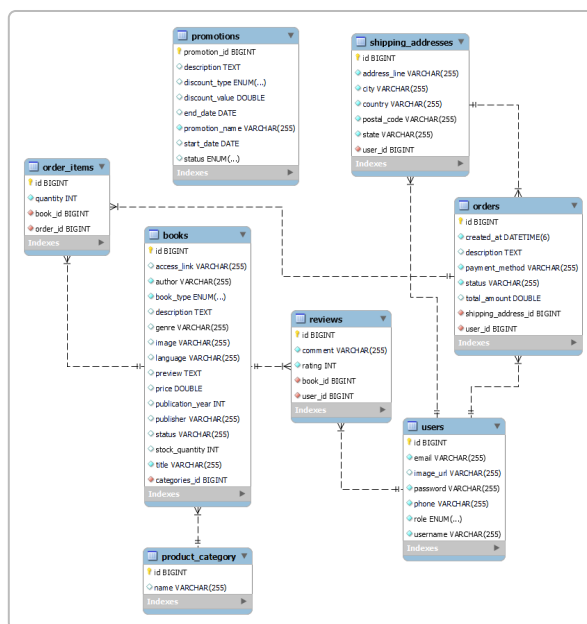
*(Liệt kê các chức năng chính dành cho người dùng (khách hàng) trên hệ thống: từ đăng ký đăng nhập đến tìm kiếm, mua sách và các tiện ích sau mua.)*

## Slide 6: Chức năng cho Quản trị viên

- **Quản lý sản phẩm (sách):** Thêm mới sách, chỉnh sửa thông tin hoặc xóa sách. Thông tin gồm tiêu đề, tác giả, giá bán, số lượng, thể loại, hình ảnh bìa, mô tả nội dung, v.v. <sup>11</sup> .
- **Quản lý đơn hàng:** Xem danh sách đơn, xác nhận đơn hàng, cập nhật trạng thái giao hàng (đã giao, hủy) và xử lý yêu cầu hoàn tiền nếu có <sup>12</sup> <sup>13</sup> .
- **Quản lý kho hàng:** Theo dõi số lượng tồn kho của từng sách; cập nhật khi nhập thêm hoặc bán ra; cảnh báo khi sách sắp hết để kịp thời bổ sung.
- **Quản lý khách hàng:** Xem danh sách tài khoản khách, chỉnh sửa hoặc xóa tài khoản. Có thể phân quyền người dùng (ví dụ nâng một user thành nhân viên quản lý).
- **Quản lý danh mục:** Tạo/sửa/xóa các danh mục thể loại sách (ví dụ: Văn học, Kinh tế, Thiếu nhi...) để phân loại sản phẩm một cách khoa học.
- **Báo cáo thống kê:** Xem báo cáo doanh thu theo ngày/tháng/năm; thống kê sách bán chạy, số đơn hàng hoàn thành, tỷ lệ đơn bị hủy, v.v. giúp đánh giá hoạt động kinh doanh.
- **Cấu hình hệ thống:** Thiết lập các tham số chung như phí vận chuyển, mã khuyến mãi; hệ thống tự động gửi thông báo cho khách (email, SMS) theo cấu hình.

*(Liệt kê các chức năng chính dành cho người quản trị hệ thống (Admin): quản lý sách và danh mục, xử lý đơn hàng, theo dõi kho, quản lý tài khoản khách, xem báo cáo và cấu hình các thiết lập hệ thống.)*

## Slide 7: Phân tích thiết kế – Cơ sở dữ liệu chính



Sơ đồ ERD thiết kế cơ sở dữ liệu: các bảng chính và quan hệ

- **Cơ sở dữ liệu gồm 8 bảng chính** tương ứng với các thực thể trong hệ thống: **books**, **users**, **orders**, **order\_items**, **reviews**, **promotions**, **shipping\_addresses**, **product\_category**. Các bảng được thiết kế với khóa chính, khóa ngoại rõ ràng, ràng buộc quan hệ chặt chẽ (mỗi đơn hàng liên kết người dùng, địa chỉ giao hàng; mỗi dòng order\_items liên kết một đơn hàng và một sách; mỗi review liên kết sách và người dùng, v.v.).
- **Sơ đồ ERD** bên phải thể hiện mối quan hệ giữa các bảng. Ví dụ: bảng **orders** liên kết bảng **users** (mỗi đơn hàng thuộc về một user), liên kết bảng **shipping\_addresses** (thông tin giao hàng), và có các chi tiết đơn hàng ở bảng **order\_items**. Bảng **books** liên kết với **order\_items** (sách có trong các đơn hàng), với **reviews** (đánh giá từ khách), và chứa khóa ngoại trỏ đến bảng **product\_category** (thể loại sách).

(Trình bày thiết kế CSDL: liệt kê các bảng dữ liệu chính và mô tả sơ lược cách các bảng liên kết với nhau. Sơ đồ ERD minh họa cấu trúc và quan hệ giữa các bảng, đảm bảo hệ thống lưu trữ dữ liệu nhất quán và hỗ trợ tốt các chức năng đã đề ra.)

## Slide 8: Kiến trúc & Công nghệ chính

- **Frontend – ReactJS**: Sử dụng ReactJS (thư viện JavaScript) để xây dựng giao diện người dùng dạng **SPA** (Single Page Application) tương tác và mượt mà. ReactJS cho phép tạo component tái sử dụng, cập nhật giao diện linh hoạt nhờ Virtual DOM, giúp ứng dụng chạy nhanh và trải nghiệm tốt.
- **Backend – Spring Boot**: Sử dụng Spring Boot (framework Java) để phát triển **RESTful API** trên phía server. Spring Boot tích hợp sẵn Spring Security để quản lý xác thực/phân quyền, dùng JPA/Hibernate để tương tác cơ sở dữ liệu MySQL, giúp lập trình viên tập trung vào logic nghiệp vụ (cấu hình đã được Spring Boot tự động hóa) <sup>14</sup> <sup>15</sup>.
- **Database – MySQL**: Sử dụng MySQL (hệ quản trị CSDL quan hệ) để lưu trữ dữ liệu có cấu trúc của hệ thống. MySQL phù hợp với bài toán quản lý bán sách nhờ hiệu năng cao, truy vấn SQL mạnh mẽ và hỗ trợ các ràng buộc quan hệ (giữa sách, đơn hàng, người dùng...) <sup>16</sup> <sup>17</sup>.

- **Kiến trúc hệ thống:** Hệ thống tuân theo mô hình **client-server**. Phía **client** là ứng dụng web ReactJS chạy trên trình duyệt, chịu trách nhiệm giao diện và gọi API. Phía **server** là ứng dụng Spring Boot triển khai các API RESTful, xử lý logic nghiệp vụ và tương tác với **database MySQL** để truy xuất hoặc lưu trữ dữ liệu <sup>18</sup>. (Mô hình triển khai đảm bảo phân tách rõ ràng giữa giao diện và xử lý, giúp hệ thống dễ mở rộng và bảo trì.)

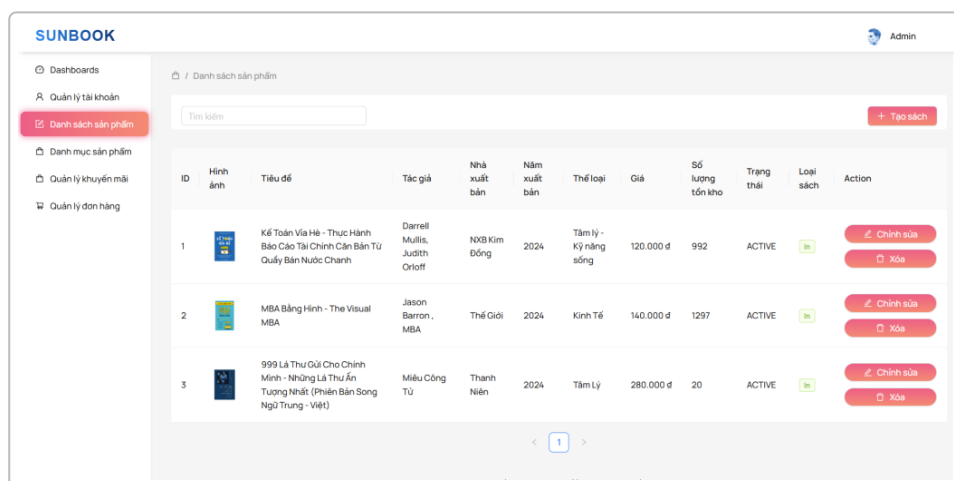
(Trình bày các công nghệ chính sử dụng trong đồ án và kiến trúc tổng thể: Giao diện web xây dựng bằng ReactJS; Server backend dùng Spring Boot cung cấp API và kết nối MySQL. Mô hình client-server giúp ứng dụng linh hoạt, có thể mở rộng thành mobile app sau này bằng cách tái sử dụng API.)

## Slide 9: Các công nghệ tích hợp

- **Redux:** Sử dụng Redux trên frontend để **quản lý trạng thái** ứng dụng một cách tập trung. Nhờ Redux, các thành phần giao diện React có thể truy xuất và cập nhật trạng thái (ví dụ: thông tin giỏ hàng, thông tin người dùng đăng nhập) một cách nhất quán, giảm thiểu lỗi và thuận tiện cho việc debug.
- **JWT (JSON Web Token):** Sử dụng JWT để **xác thực và phân quyền** người dùng khi truy cập các API bảo mật. Sau khi đăng nhập, mỗi request từ client sẽ kèm một JWT token mã hóa thông tin người dùng và quyền hạn, server dựa vào token này để xác minh và cho phép truy cập tài nguyên <sup>19</sup> <sup>20</sup>. Cơ chế này tăng cường bảo mật và tránh truy cập trái phép.
- **Stripe/PayPal API:** Tích hợp **cổng thanh toán trực tuyến** như **Stripe, PayPal** giúp khách hàng thanh toán đơn hàng ngay trên website một cách tiện lợi. Hệ thống kết nối đến API của cổng thanh toán để xử lý giao dịch, đảm bảo thông tin thanh toán được mã hóa và an toàn. (Do yêu cầu thị trường Việt Nam, đề tài đã thử nghiệm tích hợp cổng VNPAY sandbox tương tự Stripe/PayPal để mô phỏng thanh toán thẻ nội địa.)
- **Cloudinary:** Sử dụng Cloudinary (dịch vụ lưu trữ media trên mây) để quản lý **hình ảnh bìa sách**. Thay vì lưu file ảnh trên server, ứng dụng tải ảnh lên Cloudinary và lưu URL, giúp **giảm tải cho server** và người dùng tải ảnh nhanh hơn từ CDN của Cloudinary <sup>21</sup> <sup>18</sup>.

(Trình bày các công nghệ, thư viện tích hợp nhằm hoàn thiện hệ thống: Redux để quản lý state trên giao diện React; JWT để bảo mật API; tích hợp cổng thanh toán trực tuyến (Stripe/PayPal hoặc tương đương) cho chức năng mua hàng; Cloudinary để lưu trữ và phân phối ảnh bìa sách, tối ưu hiệu suất. Các công nghệ này giúp hệ thống đầy đủ tính năng, bảo mật và hiệu quả hơn.)

## Slide 10: Demo: Giao diện Quản trị viên (Admin)

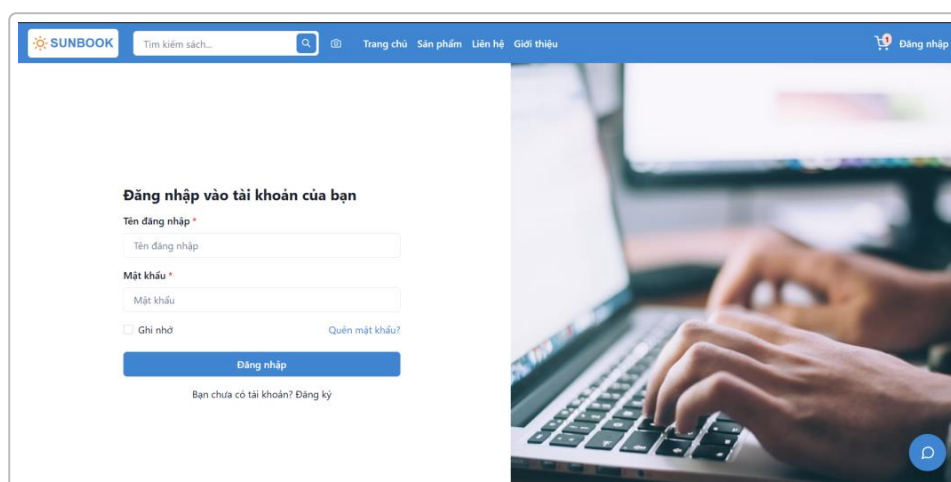


#### Giao diện trang quản trị danh sách sản phẩm (Admin)

- Hình trên là màn hình **"Danh sách sản phẩm"** trong trang quản trị. Giao diện quản trị được thiết kế hiện đại và trực quan với thanh menu bên trái để điều hướng giữa các chức năng (Quản lý tài khoản, Danh sách sản phẩm, Danh mục, Khuyến mãi, Đơn hàng...).
- Trang danh sách sản phẩm liệt kê các sách hiện có trong hệ thống cùng các thông tin quan trọng (tiêu đề, tác giả, nhà XB, năm XB, thể loại, giá, số lượng tồn kho, trạng thái kinh doanh...). Quản trị viên có thể dễ dàng **chỉnh sửa** thông tin sách hoặc **xóa** sách thông qua các nút hành động (Edit, Delete) màu sắc nổi bật ở bên phải mỗi dòng. Nút **" + Tạo sách "** cho phép thêm sách mới.
- Giao diện Admin đảm bảo tính đồng nhất và dễ sử dụng: các bảng dữ liệu rõ ràng, có ô tìm kiếm, phân trang (nếu dữ liệu nhiều) và các nút chức năng được tô màu nổi bật (màu xanh/đỏ) để thu hút sự chú ý, giúp Admin thao tác nhanh và chính xác.

*(Demo tính năng quản trị: minh họa một màn hình tiêu biểu trong trang Admin – ở đây là quản lý danh sách sách. Sinh viên trình bày cách admin sử dụng giao diện này để theo dõi và thao tác trên dữ liệu sách.)*

## Slide 11: Demo: Giao diện Khách hàng (Client)



#### Giao diện trang Đăng nhập dành cho khách hàng

- Hình trên là giao diện **đăng nhập** của người dùng. Khách hàng cần nhập tên đăng nhập (hoặc email) và mật khẩu để truy cập vào hệ thống. Giao diện được thiết kế **đơn giản, trực quan** với tông màu sáng, ô nhập liệu rõ ràng và nút **"Đăng nhập"** nổi bật. Cũng có liên kết **"Quên mật khẩu?"** để hỗ trợ người dùng khi cần và tùy chọn **"Đăng ký"** cho người dùng mới chưa có tài khoản.
- Thanh điều hướng (navbar) phía trên hiển thị logo **SUNBOOK** của trang web, thanh tìm kiếm sách (cho phép khách hàng gõ từ khóa tìm sách một cách nhanh chóng) và các mục chính như Trang chủ, Sản phẩm, Liên hệ, Giới thiệu. Góc phải thanh menu có biểu tượng giỏ hàng và nút **"Đăng nhập"** (hoặc tên user khi đã đăng nhập). Thiết kế này đảm bảo ngay cả **người dùng lần đầu** cũng dễ dàng hiểu cách sử dụng.
- Sau khi đăng nhập thành công, khách hàng sẽ có thể thực hiện các chức năng đã liệt kê (tìm kiếm sách, xem chi tiết, cho sách vào giỏ và thanh toán...). Giao diện người dùng của các trang đó cũng được thiết kế nhất quán với phong cách đơn giản, thân thiện, tập trung vào nội dung sách (ví dụ: trang danh sách sản phẩm hiển thị bìa sách, tên sách, giá và nút mua; trang chi tiết sách hiển thị ảnh bìa lớn, thông tin chi tiết và nút thêm vào giỏ hàng).

*(Demo tính năng khách hàng: minh họa giao diện trang đăng nhập và mô tả tổng quan trải nghiệm người dùng trên website bán sách. Nhấn mạnh sự thân thiện, dễ sử dụng và các thành phần chính trên giao diện người dùng.)*

## Slide 12: Kết luận

- **Hoàn thành hệ thống:** Đề tài đã **xây dựng thành công một website bán sách trực tuyến** đáp ứng đầy đủ các chức năng đề ra. Hệ thống được phát triển trên **nền tảng công nghệ hiện đại** (ReactJS, Spring Boot, MySQL), kiến trúc rõ ràng, phù hợp xu hướng, giúp ứng dụng chạy ổn định và có thể mở rộng trong tương lai <sup>22</sup> <sup>23</sup> .
- **Chức năng đáp ứng yêu cầu:** Hệ thống đã triển khai được **các tính năng thiết yếu của một website thương mại điện tử** về sách: từ quản lý sản phẩm, giỏ hàng, **xử lý thanh toán trực tuyến** cho đến gửi **email thông báo** cho khách hàng khi đặt hàng thành công. Tất cả hoạt động **trôi chảy, hiệu quả**, góp phần số hóa quy trình bán sách truyền thống. (Ví dụ: *tích hợp thành công cổng thanh toán online giúp khách mua sách thuận tiện hơn nhiều so với trước đây.*)
- **Tích hợp dịch vụ bên ngoài:** Hệ thống có khả năng **tương tác với dịch vụ bên thứ ba** để nâng cao tiện ích. Cụ thể, đã tích hợp **Google Books API** để tự động lấy dữ liệu sách (thông tin mô tả, bìa sách...) cho hệ thống, giúp việc nhập liệu sách mới nhanh chóng và chính xác hơn <sup>24</sup> <sup>25</sup> . Điều này thể hiện **tính linh hoạt và mở rộng** của hệ thống – có thể dễ dàng bổ sung thêm nguồn dữ liệu hoặc dịch vụ mới trong tương lai.
- **Kiểm thử và chất lượng:** Hệ thống đã trải qua **quá trình kiểm thử toàn diện**: kiểm thử chức năng đảm bảo mỗi yêu cầu hoạt động đúng; kiểm thử hiệu năng cho thấy hệ thống đáp ứng tốt với lượng người dùng vừa phải; đặc biệt **kiểm thử bảo mật** (sử dụng các công cụ quét OWASP) cho kết quả **không phát hiện lỗ hổng nghiêm trọng** <sup>26</sup> <sup>27</sup> . Nhờ đó, hệ thống vận hành **ổn định, tin cậy và an toàn**, đáp ứng được các tiêu chí đề ra về trải nghiệm người dùng và bảo mật dữ liệu (quan trọng đối với thông tin nhạy cảm như thanh toán, tài khoản khách hàng).

*(Tóm tắt những kết quả chính đạt được: hệ thống đã hoàn thiện và chạy tốt, đáp ứng mục tiêu đề ra. Nêu bật việc ứng dụng đúng công nghệ, triển khai đủ chức năng, tích hợp thêm dịch vụ để tăng tính năng, và đã kiểm thử kỹ lưỡng để đảm bảo chất lượng hệ thống.)*

## Slide 13: Hạn chế

- **Tích hợp công nghệ phức tạp:** Việc tích hợp nhiều công nghệ (frontend, backend, DB, cổng thanh toán...) đòi hỏi nhiều thời gian nghiên cứu và cấu hình. Trong quá trình phát triển đã gặp khó khăn khi cấu hình tương tác giữa các thành phần (đặc biệt với lập trình viên chưa có nhiều kinh nghiệm DevOps). **Môi trường sandbox của cổng thanh toán** đôi khi xảy ra lỗi không tương thích quy trình thực tế, dẫn đến phải điều chỉnh nhiều lần và có thể gây gián đoạn nếu khách hàng thanh toán đúng lúc gặp lỗi.
- **Hiệu năng chưa tối ưu ở tải cao:** Hệ thống chạy mượt với lượng người dùng vừa phải, nhưng khi **số lượng người dùng đồng thời tăng >50** thì bắt đầu có dấu hiệu chậm lại. Điều này cho thấy backend Spring Boot và cơ sở dữ liệu MySQL hiện tại **chưa tối ưu cho tải lớn**, tiềm ẩn rủi ro nếu triển khai thực tế có nhiều người dùng hoặc các đợt khuyến mãi thu hút lượng truy cập cao <sup>28</sup> <sup>29</sup> .
- **Thiếu một số tính năng nâng cao:** Do giới hạn thời gian, đề án **chưa kịp phát triển các tính năng nâng cao** như hệ thống **gợi ý sách thông minh** dựa trên lịch sử/tương tác của người dùng (để cá nhân hóa trải nghiệm mua sắm), hoặc tích hợp **mạng xã hội** để người dùng chia sẻ sở thích. Ngoài ra, hiện tại khách hàng **chưa thể đánh giá, nhận xét sách trực tiếp trên website** (mặc dù hệ thống có bảng reviews, tính năng này chưa được triển khai đầy đủ trên giao diện) <sup>30</sup> . Những thiếu hụt này nếu được bổ sung sẽ tăng tính cạnh tranh cho sản phẩm so với các trang thương mại điện tử khác.
- **Giới hạn về địa lý/ngôn ngữ:** Hệ thống **chưa hỗ trợ đa ngôn ngữ** (mới chỉ có tiếng Việt) và **chưa có tùy biến theo vùng** (ví dụ đơn vị tiền tệ, địa chỉ). Điều này khiến sản phẩm **chỉ phù hợp cho thị trường trong nước**, nếu muốn mở rộng ra quốc tế sẽ cần cải tiến. Việc không có đa

ngôn ngữ cũng làm giảm trải nghiệm đối với người dùng không rành tiếng Việt, giới hạn tệp khách hàng.

*(Trung thực chỉ ra những hạn chế của đồ án: tích hợp nhiều công nghệ gây phức tạp, hiệu năng cần cải thiện cho quy mô lớn, một số tính năng nâng cao chưa kịp làm, và phạm vi áp dụng còn hạn chế ở thị trường nội địa. Những hạn chế này là mục tiêu để tiếp tục hoàn thiện trong tương lai.)*

## Slide 14: Hướng phát triển

- **Mở rộng phương thức thanh toán:** Tích hợp thêm các phương thức thanh toán phổ biến khác tại Việt Nam như **ví điện tử Momo, ZaloPay** và hỗ trợ **thẻ quốc tế (Visa, Mastercard)**. Điều này giúp khách hàng có **nhiều lựa chọn thanh toán**, tăng **tỷ lệ chuyển đổi đơn hàng**. Bên cạnh đó, xây dựng **cơ chế tự động thử lại giao dịch** khi kết nối thanh toán bị lỗi hoặc mạng không ổn định, để khách không phải thao tác lại từ đầu, cải thiện trải nghiệm người dùng.
- **Tối ưu hiệu năng hệ thống:** Triển khai các biện pháp nâng cao hiệu suất cho server: áp dụng **caching** (ví dụ tích hợp Redis) để lưu tạm kết quả các truy vấn thường xuyên, giảm tải cho database; đồng thời có thể triển khai **load balancing** (cân bằng tải) và **mở rộng theo chiều ngang** (horizontal scaling) – chạy nhiều instance server song song – nhằm phục vụ nhiều người dùng đồng thời hơn mà không làm giảm tốc độ.
- **Ứng dụng công nghệ DevOps:** Đưa hệ thống lên các nền tảng **container** như Docker, quản lý bằng **Kubernetes** để dễ dàng mở rộng và triển khai linh hoạt trên môi trường thực tế. Song song, tích hợp các công cụ **giám sát và phân tích log** (ví dụ: **ELK Stack** – Elasticsearch, Logstash, Kibana) để theo dõi sức khỏe hệ thống theo thời gian thực, nhanh chóng phát hiện và xử lý các sự cố về hiệu năng, lỗi hệ thống... đảm bảo dịch vụ ổn định.
- **Mở rộng nền tảng và phạm vi:** Phát triển thêm **ứng dụng di động (mobile app)** cho hệ sinh thái SunBook nhằm tiếp cận nhiều khách hàng hơn (đáp ứng xu hướng người dùng thích mua sắm trên điện thoại). Đồng thời, bổ sung **đa ngôn ngữ** và tùy chỉnh nội dung theo khu vực (ví dụ: hỗ trợ tiếng Anh, đơn vị tiền tệ quốc tế) – bước chuẩn bị để **mở rộng thị trường ra quốc tế** trong tương lai.

*(Đề xuất những hướng phát triển tiếp theo cho đồ án: tích hợp thêm phương thức thanh toán và cải thiện quy trình thanh toán; tối ưu hiệu năng bằng kỹ thuật caching, scaling; áp dụng container hóa và công cụ quản trị để sẵn sàng triển khai thực tế; phát triển ứng dụng di động và hỗ trợ đa ngôn ngữ để mở rộng tầm ảnh hưởng của hệ thống.)*

## Slide 15: Q&A – Kết thúc

- **Q&A (Questions & Answers):** Sẵn sàng giải đáp các câu hỏi của hội đồng chấm và khán giả về nội dung đồ án.
- *Xin chân thành cảm ơn!* (Kết thúc phần trình bày, **cảm ơn quý thầy cô và các bạn** đã lắng nghe)

*(Slide cuối cùng, thường để lại lời cảm ơn và chuyển sang phần hỏi đáp. Sinh viên kết thúc trình bày và sẵn sàng trả lời các câu hỏi.)*