TRƯỜNG ĐẠI HỌC CÔNG NGHỆ SÀI GÒN

KHOA CÔNG NGHỆ THÔNG TIN

BỘ MÔN ĐỒ ÁN CHUYÊN NGÀNH



TÊN ĐỀ TÀI: XÂY DỰNG ỨNG DỤNG WEB LUYỆN THI TIẾNG ANH (EnglishMaster)

Giảng viên hướng dẫn: Thầy TRẦN QUỐC TRƯỜNG

Sinh viên thực hiện:

Họ và tên: Đặng Trường Thi

MSSV: DH52201479

Lớp: D22\_TH10

TP. HỒ CHÍ MINH, THÁNG 11 NĂM 2025

Giáo viên chấm điểm: .....................................................

Điểm số: .....................................................

**LỜI CẢM ƠN**

Trước tiên, em xin gửi lời cảm ơn chân thành đến Ban Giám hiệu Trường Đại học Công nghệ Sài Gòn cùng tập thể giảng viên Khoa Công nghệ Thông tin đã tạo điều kiện thuận lợi để em hoàn thành chương trình học và thực hiện đồ án tốt nghiệp.

Em xin bày tỏ lòng biết ơn sâu sắc đến Thầy ThS. Trần Quốc Trường – giảng viên hướng dẫn – người đã tận tình hướng dẫn, chỉ bảo và hỗ trợ em trong suốt quá trình nghiên cứu. Sự định hướng và kinh nghiệm quý báu của Thầy đã giúp em vượt qua nhiều khó khăn để hoàn thiện đề tài.

Em cũng xin cảm ơn các thầy cô trong Khoa Công nghệ Thông tin đã trang bị cho em nền tảng kiến thức vững chắc trong suốt thời gian học tập.

Cuối cùng, em xin tri ân gia đình, bạn bè và những người thân yêu đã luôn động viên, chia sẻ và tạo điều kiện để em hoàn thành đồ án này.

Tp. Hồ Chí Minh, ngày 15 tháng 11 năm 2025 Sinh viên Đặng Trường Thi

**Mục Lục**

TÓM TẮT ĐỀ TÀI ...............................................................................................1

1. CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI ...............................................................2

1.1. Giới thiệu chung ......................................................................................... 2

1.2. Lý do chọn đề tài ........................................................................................ 2

1.3. Mục tiêu của đề tài ..................................................................................... 2

1.4. Yêu cầu hệ thống ........................................................................................ 3

1.4.1. Yêu cầu chức năng ............................................................................. 3

1.4.2. Yêu cầu phi chức năng ....................................................................... 5

2. CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG ............................ 6

2.1. Sơ đồ Use Case............................................................................................. 6

2.2. Biểu đồ phân rã chức năng (D0, D1) ........................................................ 7

2.3. Biểu đồ thực thể - mối quan hệ (ERD) ..................................................... 9

3. CHƯƠNG 3: HIỆN THỰC ĐỀ TÀI ............................................................ 10

3.1. Chức năng Đăng ký .................................................................................. 10

3.1.1. Mô tả chức năng ............................................................................... 10

3.1.2. Giao diện đăng ký ............................................................................. 11

3.1.3. Hiện thực xử lý nghiệp vụ (Backend) ............................................. 11

3.2. Chức năng Đăng nhập ............................................................................. 13

3.2.1. Mô tả chức năng ............................................................................... 13

3.2.2. Giao diện đăng nhập ........................................................................ 13

3.2.3. Hiện thực xử lý nghiệp vụ (Backend) ............................................. 14

3.3. Chức năng Xem danh sách khóa học ..................................................... 16

4.3.1. Mô tả chức năng ............................................................................... 16

4.3.2. Giao diện hiển thị ..............................................................................16

4.3.3. Hiện thực xử lý nghiệp vụ (Backend) ............................................. 17

3.4. Chức năng Quản lý người dùng (Admin) .............................................. 19

3.4.1. Mô tả chức năng ............................................................................... 19

3.4.2. Giao diện quản trị người dùng ........................................................ 20

3.4.3. Hiện thực xử lý nghiệp vụ (Backend) ..............................................21

3.5. Chức năng Quản lý khóa học (Admin) .................................................... 23

3.5.1. Mô tả chức năng ............................................................................... 23

3.5.2. Giao diện quản trị ............................................................................ 23

3.5.3. Hiện thực xử lý nghiệp vụ (Backend) ............................................. 24

3.6. Chức năng Quản lý đơn hàng (Admin) .................................................... 26

3.6.1. Mô tả chức năng ............................................................................... 26

3.6.2. Giao diện quản trị đơn hàng ........................................................... 27

3.6.3. Hiện thực xử lý nghiệp vụ (Backend) ............................................. 28

4. CHƯƠNG 4: KẾT LUẬN ............................................................................. 30

4.1. Kết quả đạt được ........................................................................................ 30

4.1.1. Về mặt kỹ thuật ................................................................................ 30

4.1.2. Về mặt chức năng ............................................................................. 31

4.1.3. Về mặt trải nghiệm người dùng ...................................................... 32

4.1.4. Về mặt ứng dụng thực tiễn .............................................................. 33

4.2. Hạn chế của đề tài ...................................................................................... 34

5. CHƯƠNG 5: HƯỚNG PHÁT TRIỂN CỦA ĐỀ TÀI ................................. 35

5.1. Nâng cấp tính năng quản trị ..................................................................... 35

5.1.1. Hệ thống báo cáo thông minh ......................................................... 35

5.1.2. Quản lý nội dung học tập ................................................................. 36

5.1.3. Tự động hóa tương tác và chăm sóc học viên ................................ 36

5.2. Mở rộng trải nghiệm người dùng ............................................................. 37

5.2.1. Hệ thống tìm kiếm và lọc thông minh ............................................ 37

5.2.2. Cá nhân hóa trải nghiệm học viên .................................................. 37

5.2.3. Tương tác cộng đồng ........................................................................ 38

5.2.4. Học tập tương tác ............................................................................. 39

5.3. Tích hợp công nghệ tiên tiến ...................................................................... 39

5.3.1. Thanh toán Online ........................................................................... 39

5.3.2. Phát triển Mobile App ..................................................................... 40

5.3.3. Trí tuệ nhân tạo (AI) và Học máy (Machine Learning) ............... 40

5.3.4. Triển khai trên Cloud ...................................................................... 41

5.3.5. Bảo mật nâng cao ............................................................................. 41

5.3.6. Phân tích dữ liệu (Analytics) ........................................................... 42

5.4. Mở rộng mô hình kinh doanh .................................................................... 43

5.4.1. Đa dạng hóa sản phẩm ..................................................................... 43

5.4.2. Chương trình đối tác ........................................................................ 43

5.4.3. Mở rộng thị trường .......................................................................... 44

6. GIAO DIỆN BỔ SUNG.................................................................................. 40

TÀI LIỆU THAM KHẢO ................................................................................. 45

**TÓM TẮT ĐỀ TÀI**

Tên đề tài: Xây dựng ứng dụng web luyện thi tiếng Anh (EnglishMaster).

Mục tiêu: Phát triển nền tảng web hỗ trợ luyện thi các chứng chỉ tiếng Anh quốc tế (TOEIC, IELTS, TOEFL) với bài tập tương tác, đề thi thử, chấm điểm tự động và theo dõi tiến độ học tập cá nhân hóa.

Phạm vi: Tập trung vào 3 chứng chỉ chính với đầy đủ 4 kỹ năng (Nghe, Nói, Đọc, Viết).

Công nghệ sử dụng:

* Frontend: HTML5, CSS3, JavaScript, Bootstrap.
* Backend: Node.js, Express.js.
* Database: MongoDB
* Công cụ hỗ trợ: Git

Kết quả mong đợi: Ứng dụng hoàn chỉnh, thân thiện, giúp người dùng cải thiện kỹ năng tiếng Anh hiệu quả.

**1. CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI**

**1.1. Giới thiệu chung**

Trong bối cảnh toàn cầu hóa, tiếng Anh là ngôn ngữ thiết yếu cho sự phát triển cá nhân và nghề nghiệp. Các chứng chỉ như TOEIC, IELTS, TOEFL ngày càng quan trọng đối với sinh viên và người lao động tại Việt Nam.

Ứng dụng web EnglishMaster được phát triển nhằm cung cấp giải pháp học tập trực tuyến toàn diện, với môi trường luyện tập tương tác và đánh giá năng lực tiếng Anh hiệu quả.

**1.2. Lý do chọn đề tài**

**1.2.1. Nhu cầu thực tế**

* Số lượng người luyện thi tiếng Anh tại Việt Nam tăng mạnh.
* Cần các nền tảng học tập linh hoạt, tiết kiệm chi phí và thời gian.
* Thiếu các ứng dụng trực tuyến chất lượng cao với nội dung chuẩn hóa.

**1.2.2. Lợi ích công nghệ**

* Áp dụng công nghệ web hiện đại để tạo trải nghiệm người dùng tốt.
* Sử dụng framework mã nguồn mở để phát triển nhanh chóng.
* Dễ dàng mở rộng và bảo trì hệ thống.

**1.2.3. Đóng góp cá nhân**

* Áp dụng kiến thức lập trình full-stack vào thực tế.
* Rèn luyện kỹ năng phát triển ứng dụng web hoàn chỉnh.
* Tích lũy kinh nghiệm dự án thực tiễn.

**1.3. Mục tiêu của đề tài**

**1.3.1. Mục tiêu tổng quát**

Xây dựng ứng dụng web luyện thi tiếng Anh hoàn chỉnh, hiệu quả và thân thiện với người dùng.

**1.3.2. Mục tiêu cụ thể**

* Phát triển hệ thống quản lý người dùng và phân quyền.
* Xây dựng kho bài tập, đề thi đa dạng.
* Triển khai chấm điểm tự động và theo dõi tiến độ.
* Thiết kế giao diện trực quan, responsive.
* Triển khai ứng dụng lên môi trường production.

**1.4 Yêu cầu hệ thống**

**1.4.1. Yêu cầu chức năng**

**1.4.1.1. Phân hệ Người dùng (Học viên)**

**1.4.1.1.1. Quản lý tài khoản**

- Đăng ký: Cho phép người dùng tạo tài khoản mới với các thông tin cá nhân (Họ tên, Email, Mật khẩu).

- Đăng nhập: Xác thực người dùng truy cập hệ thống thông qua Email và Mật khẩu.

- Đăng xuất: Cho phép người dùng thoát khỏi phiên làm việc an toàn.

- Quản lý hồ sơ: Xem và cập nhật thông tin cá nhân.

- Lịch sử học tập: Xem danh sách khóa học đã đăng ký, kết quả các bài thi thử đã làm.

**1.4.1.1.2. Khóa học trực tuyến**

- Tìm kiếm & Lọc: Tìm kiếm khóa học theo tên, lọc theo danh mục, trình độ, mức giá.

- Xem chi tiết: Hiển thị đầy đủ thông tin khóa học (giới thiệu, giảng viên, lộ trình, học phí, đánh giá).

- Học tập: Truy cập nội dung bài học (Video, Văn bản) đối với các khóa đã đăng ký thành công.

- Đánh giá: Gửi đánh giá và bình luận cho các khóa học đã tham gia.

**1.4.1.1.3. Thương mại điện tử (Mua khóa học)**

- Giỏ hàng: Thêm khóa học vào giỏ, cập nhật số lượng (nếu áp dụng), xóa khóa học khỏi giỏ.

- Thanh toán: Thực hiện đặt hàng và thanh toán (Hỗ trợ các phương thức cơ bản: Chuyển khoản, COD, v.v.).

- Gợi ý khóa học: Hiển thị các khóa học liên quan hoặc đề xuất dựa trên khóa học đang xem.

**1.4.1.1.4. Luyện thi & Tài nguyên**

- Làm bài thi thử: Thực hiện các bài thi trắc nghiệm (TOEIC, IELTS, TOEFL) có tính giờ.

- Xem kết quả: Hiển thị điểm số, đáp án đúng/sai và lời giải chi tiết ngay sau khi nộp bài.

- Tài liệu tham khảo: Truy cập kho từ vựng và ngữ pháp miễn phí.

**1.4.1.2. Phân hệ Quản trị viên (Admin)**

**1.4.1.2.1. Bảng điều khiển (Dashboard)**

- Xem thống kê tổng quan: Số lượng người dùng, Số lượng khóa học, Doanh thu, Đơn hàng mới, Liên hệ mới.

- Biểu đồ doanh thu theo năm.

**1.4.1.2.2. Quản lý nội dung đào tạo**

- Quản lý Danh mục: Thêm, sửa, xóa các danh mục khóa học (TOEIC, IELTS, Giao tiếp...).

- Quản lý Khóa học: Tạo mới, cập nhật thông tin, thay đổi trạng thái (ẩn/hiện) khóa học.

- Quản lý Bài học: Thêm video bài giảng, nội dung chi tiết cho từng khóa học.

- Quản lý Tài nguyên:Đăng tải và quản lý các bài viết về Ngữ pháp và Từ vựng.

**1.4.1.2.3. Quản lý Đề thi & Ngân hàng câu hỏi**

- Quản lý Đề thi: Tạo đề thi mới, thiết lập thời gian, điểm số.

- Ngân hàng câu hỏi: Thêm, sửa, xóa câu hỏi trắc nghiệm; phân loại theo trình độ.

- Soạn đề: Thêm/bớt câu hỏi vào đề thi từ ngân hàng câu hỏi có sẵn.

**1.4.1.2.4. Quản lý bán hàng & Người dùng**

- Quản lý Đơn hàng: Xem danh sách đơn hàng, xem chi tiết đơn hàng, cập nhật trạng thái đơn hàng (đang xử lý, hoàn thành, hủy) và tự động kích hoạt khóa học khi hoàn thành.

- Quản lý Người dùng: Xem danh sách học viên, thêm mới, sửa thông tin, khóa/xóa tài khoản.

- Quản lý Liên hệ: Xem và xử lý các tin nhắn liên hệ từ khách hàng.

- Quản lý Đánh giá: Phê duyệt hoặc ẩn các đánh giá của học viên.

**1.4.2. Yêu cầu phi chức năng**

**1.4.2.1. Hiệu năng (Performance)**

- Thời gian phản hồi: Hệ thống phản hồi các thao tác cơ bản của người dùng (tải trang, chuyển trang) trong vòng dưới 3 giây trong điều kiện mạng ổn định.

- Khả năng chịu tải: Hệ thống hoạt động ổn định với số lượng người dùng truy cập đồng thời vừa phải (đáp ứng quy mô trung tâm đào tạo).

**1.4.2.2. Bảo mật (Security)**

- Mã hóa mật khẩu: Mật khẩu người dùng phải được mã hóa (hashing) trước khi lưu vào cơ sở dữ liệu (sử dụng Bcrypt hoặc tương đương).

- Xác thực & Phân quyền:

- Cơ chế phiên làm việc (Session-based Authentication) để quản lý đăng nhập.

- Phân quyền rõ ràng giữa Người dùng thường và Quản trị viên; chặn truy cập trái phép vào các trang quản trị (Admin routes protection).

- Bảo vệ dữ liệu: Dữ liệu thanh toán và thông tin cá nhân cần được xử lý an toàn.

**1.4.2.3. Tính khả dụng & Tin cậy (Availability & Reliability)**

- Hệ thống hoạt động liên tục 24/7.

- Dữ liệu được lưu trữ bền vững trên Cơ sở dữ liệu (MongoDB), đảm bảo tính toàn vẹn (ACID cho các giao dịch quan trọng nếu cần).

**1.4.2.4. Giao diện & Trải nghiệm người dùng (UI/UX)**

- Responsive Design: Giao diện hiển thị tốt và tương thích trên nhiều thiết bị: Máy tính cá nhân (PC/Laptop), Máy tính bảng (Tablet) và Điện thoại di động (Mobile).

- Thẩm mỹ: Giao diện hiện đại, màu sắc hài hòa, dễ nhìn, phù hợp với lĩnh vực giáo dục.

- Dễ sử dụng: Các chức năng được bố trí logic, dễ tìm kiếm và thao tác.

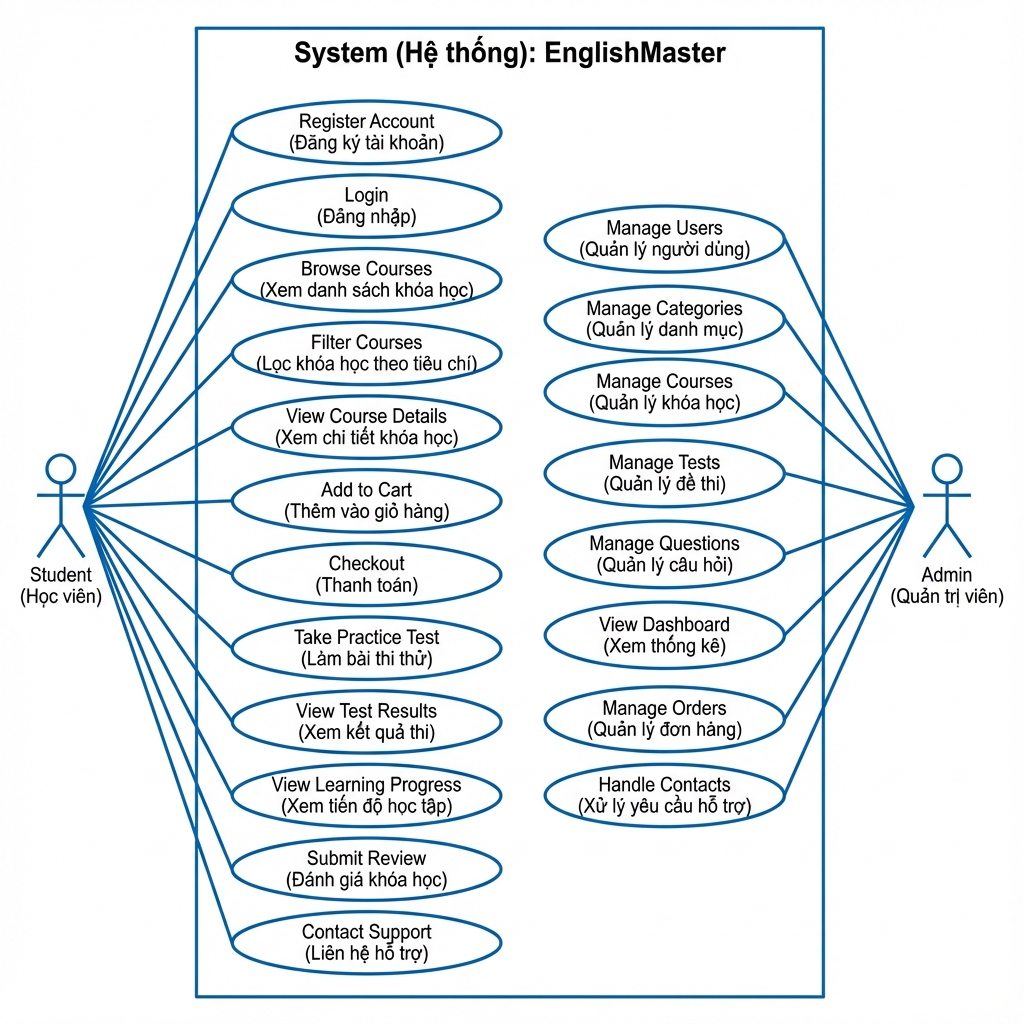
**1.4.2.5. Khả năng tương thích & Mở rộng**

- Trình duyệt: Tương thích với các trình duyệt web phổ biến hiện nay (Chrome, Firefox, Edge, Safari).

- Công nghệ:Xây dựng trên nền tảng Node.js và Express, dễ dàng mở rộng thêm các tính năng mới (API, Mobile App integration) trong tương lai.

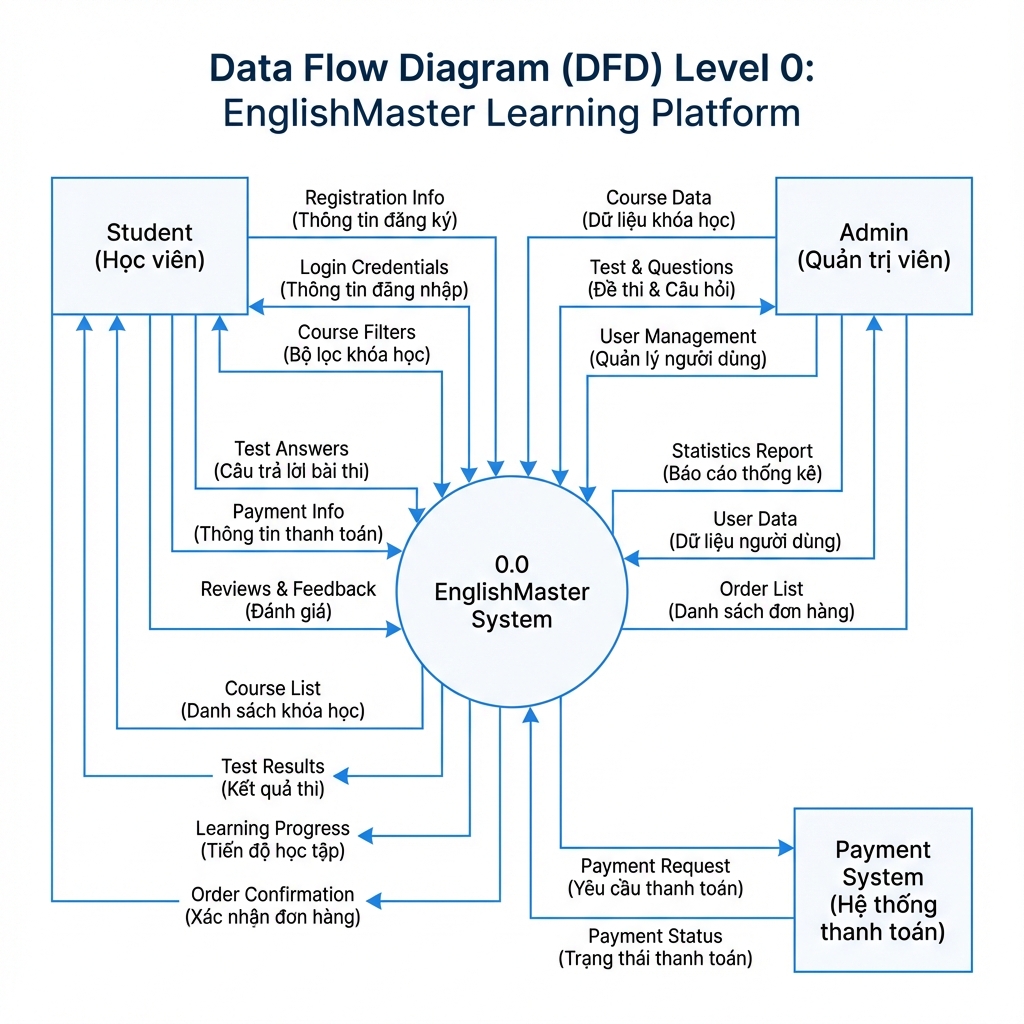
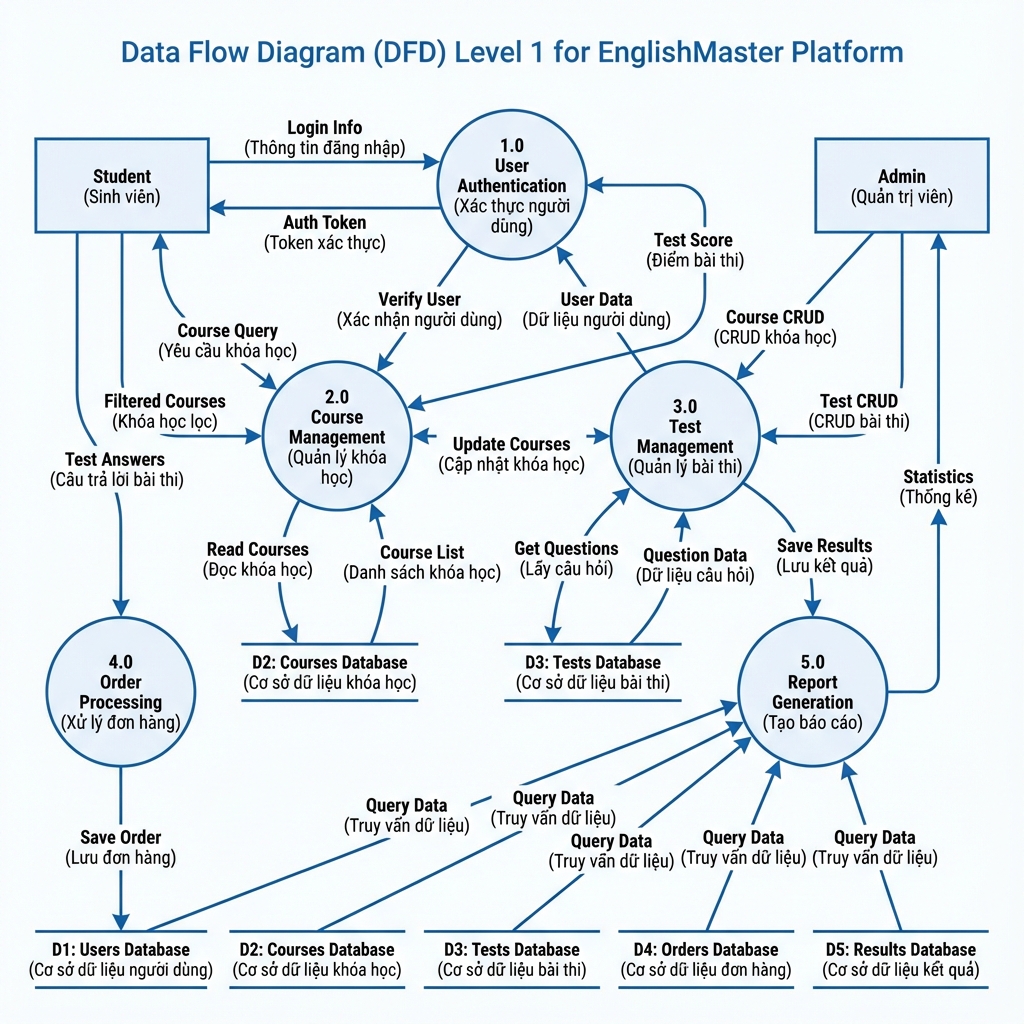
**2. CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG**

**2.1. Sơ đồ USE CASE**

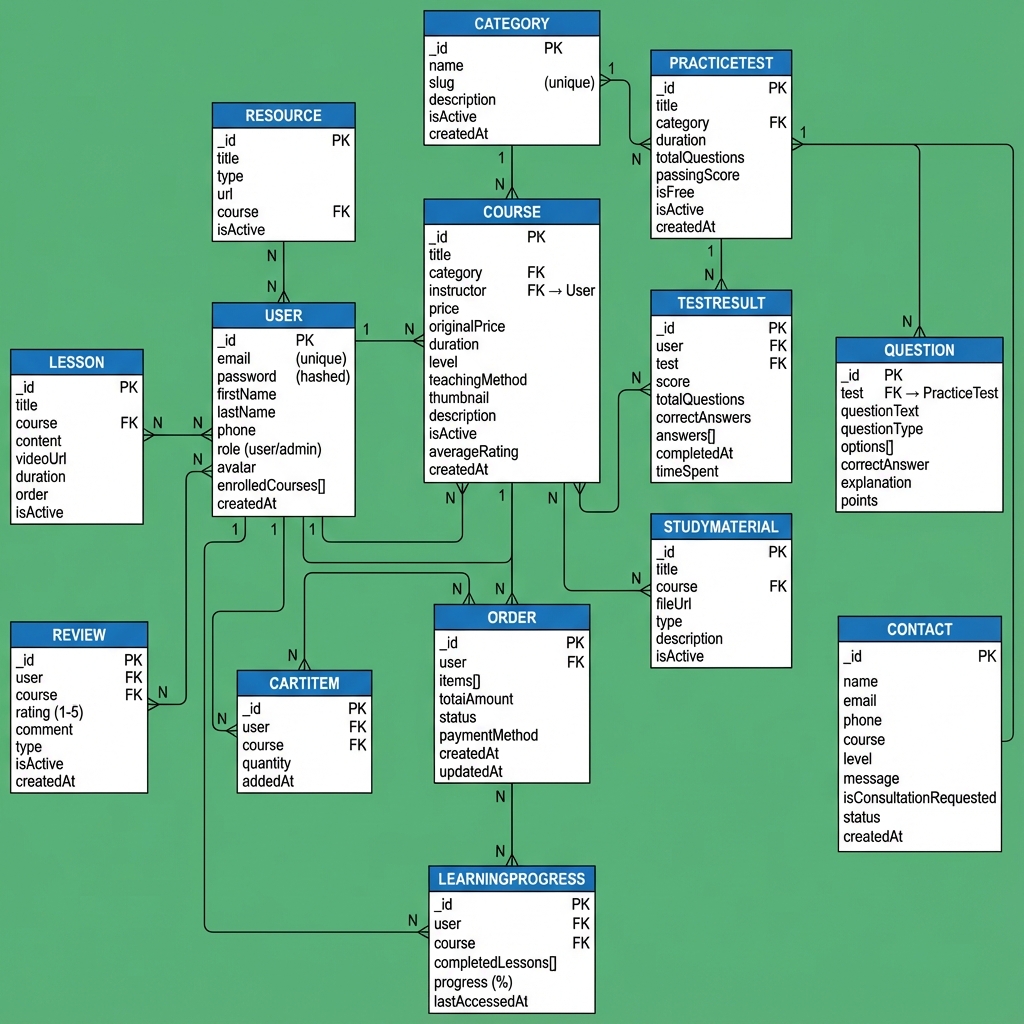
****

**2.2. Biểu đồ phân rã chức năng (D0, D1)**

**2.2.1. Biểu đồ DFD mức 0**

**2.2.2. Biểu đồ DFD mức 1**

**2.3. Biểu đồ thực thể - mối quan hệ(ERD):**



**CHƯƠNG 3: HIỆN THỰC ĐỀ TÀI**

**3.1. CHỨC NĂNG ĐĂNG KÝ**

**3.1.1. MÔ TẢ CHỨC NĂNG**

**Chức năng đăng ký cho phép người dùng mới tạo tài khoản để tham gia vào hệ thống học tiếng Anh trực tuyến EnglishMaster. Thông qua việc cung cấp các thông tin cá nhân cơ bản, người dùng sẽ có danh tính định danh để thực hiện các thao tác đăng ký khóa học, làm bài thi và quản lý tiến trình học tập cá nhân.**

**3.1.2. GIAO DIỆN ĐĂNG KÝ**

**Giao diện bao gồm các trường nhập liệu thiết yếu như: Họ (Last Name), Tên (First Name), Email và Mật khẩu (Password). Hệ thống yêu cầu người dùng phải điền đầy đủ các thông tin bắt buộc trước khi khởi tạo tài khoản. Giao diện được thiết kế hiện đại với màu sắc chủ đạo xanh dương, tạo cảm giác chuyên nghiệp và thân thiện.**

**3.1.3. HIỆN THỰC XỬ LÝ NGHIỆP VỤ (BACKEND)**

**Quy trình xử lý đăng ký được thực hiện nghiêm ngặt qua các bước kiểm tra và bảo mật dữ liệu:**

**Kiểm tra tính hợp lệ của dữ liệu (Validation):**

**- Hệ thống thực hiện kiểm tra phía Server-side để đảm bảo các trường firstName, lastName, email và password không được để trống.**

**- Nếu thiếu thông tin, hệ thống sẽ tạo mảng errors và sử dụng req.flash('registerErrors', error) để lưu trữ các thông báo lỗi, sau đó hiển thị lại trang đăng ký kèm theo các thông báo lỗi cụ thể cho người dùng.**

**Kiểm tra tài khoản tồn tại:**

**- Trước khi tạo mới, hệ thống sử dụng phương thức User.findOne({email: req.body.email}) để kiểm tra xem Email đã được sử dụng trong cơ sở dữ liệu MongoDB hay chưa.**

**- Trong trường hợp Email đã tồn tại, một thông báo lỗi "Email is exist!" sẽ được gửi qua req.flash và điều hướng người dùng quay lại trang đăng ký.**

**Cấp quyền người dùng (Role Assignment):**

**- Mặc định hệ thống sẽ gán vai trò là "user" cho các tài khoản đăng ký mới nếu không có thông tin vai trò cụ thể được gửi lên, đảm bảo an toàn phân quyền.**

**- Hệ thống hỗ trợ hai loại vai trò chính: "user" (người học) và "admin" (quản trị viên).**

**Mã hóa mật khẩu (Security):**

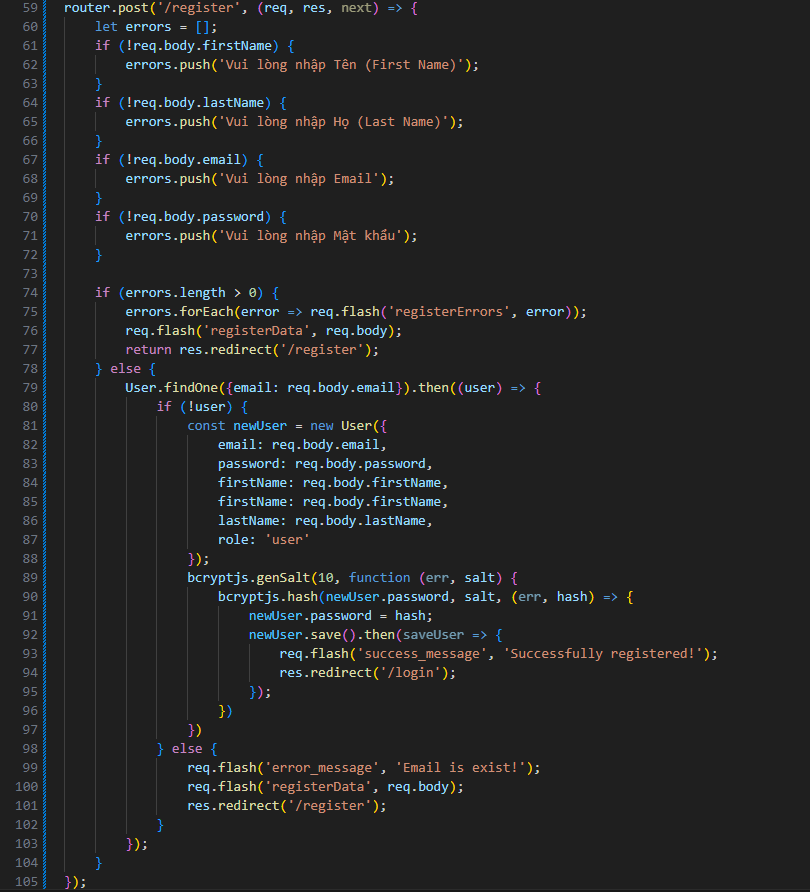
**- Để đảm bảo an toàn thông tin, mật khẩu không được lưu trực tiếp vào cơ sở dữ liệu.**

**- Hệ thống sử dụng thư viện bcryptjs để tạo mã muối (salt) với độ phức tạp là 10 (genSalt(10)), sau đó thực hiện băm mật khẩu (hashing) trước khi lưu vào cơ sở dữ liệu MongoDB.**

**- Quá trình này đảm bảo rằng ngay cả khi cơ sở dữ liệu bị xâm nhập, mật khẩu gốc của người dùng vẫn được bảo vệ an toàn.**

**Hoàn tất đăng ký:**

**- Sau khi lưu người dùng thành công (newUser.save()), hệ thống sẽ gửi thông báo "Successfully registered!" thông qua flash message và tự động điều hướng người dùng sang trang đăng nhập để bắt đầu phiên làm việc.**

****

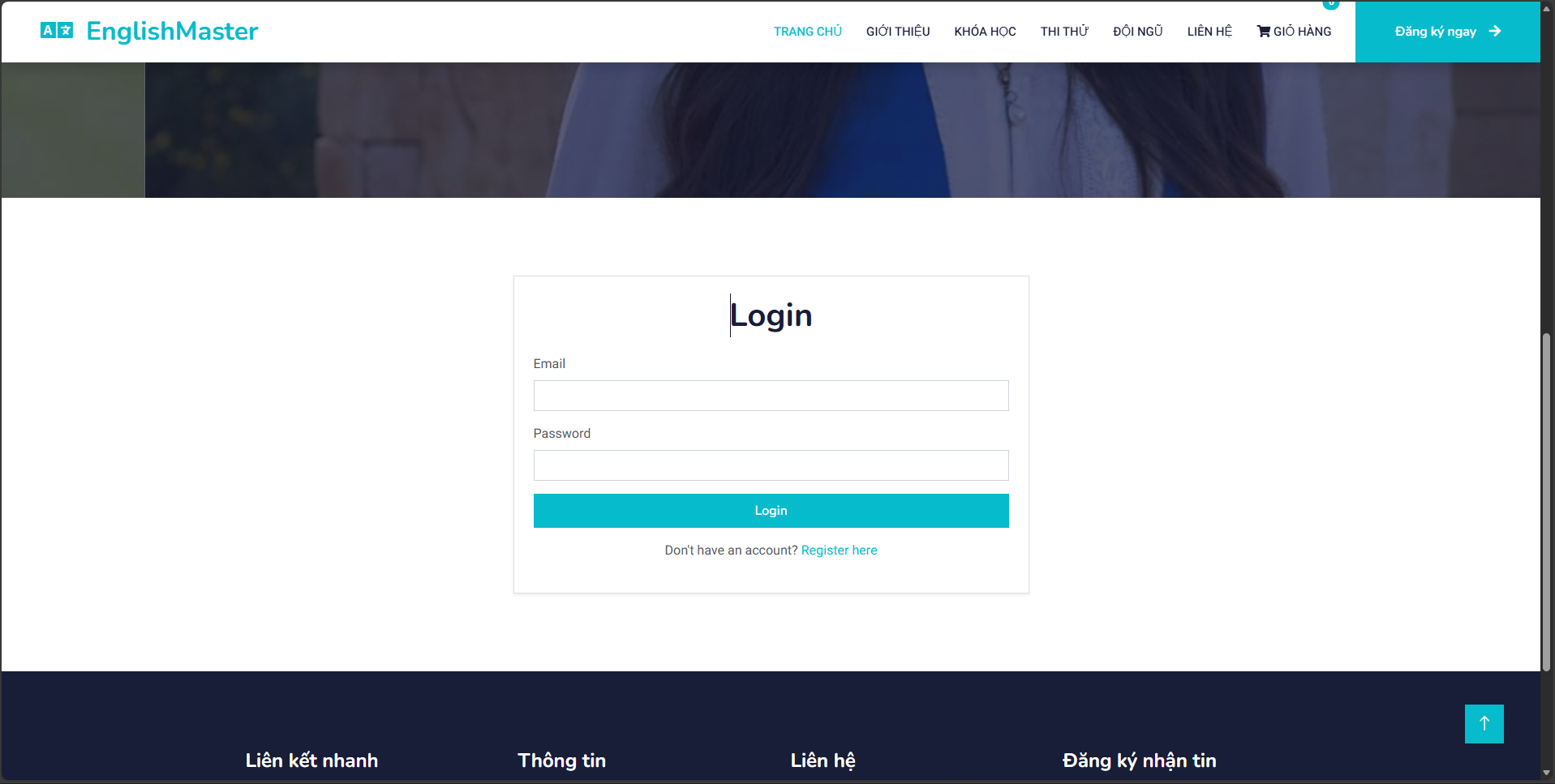
**3.2. CHỨC NĂNG ĐĂNG NHẬP**

**3.2.1. MÔ TẢ CHỨC NĂNG**

**Chức năng đăng nhập cho phép người dùng truy cập vào hệ thống bằng tài khoản đã được đăng ký trước đó. Sau khi xác thực thành công, hệ thống sẽ kiểm tra vai trò (Role) để điều hướng người dùng: nếu là "admin" sẽ được truy cập trang quản trị, ngược lại sẽ trở về trang chủ dành cho người học.**

**3.2.2. GIAO DIỆN ĐĂNG NHẬP**

**Giao diện được thiết kế tối giản với form nhập liệu bao gồm Email và Mật khẩu. Hệ thống tích hợp thông báo lỗi ngay trên giao diện nếu người dùng để trống thông tin hoặc nhập sai tài khoản. Giao diện cũng cung cấp liên kết nhanh đến trang đăng ký cho người dùng mới.**

**3.2.3. HIỆN THỰC XỬ LÝ NGHIỆP VỤ (BACKEND)**

**Hệ thống sử dụng thư viện Passport.js kết hợp với passport-local để xử lý quá trình xác thực. Quy trình hiện thực cụ thể như sau:**

**Cấu hình chiến lược xác thực (Local Strategy):**

**- Sử dụng trường email làm định danh duy nhất (usernameField: 'email') để tìm kiếm người dùng trong cơ sở dữ liệu thông qua phương thức User.findOne({email: email}).**

**- Sử dụng thư viện bcryptjs với phương thức compare() để so sánh mật khẩu người dùng nhập vào với mật khẩu đã được mã hóa (hash) lưu trong database nhằm đảm bảo tính bảo mật.**

**- Nếu email không tồn tại hoặc mật khẩu không khớp, hệ thống sẽ trả về thông báo lỗi tương ứng.**

**Xử lý yêu cầu Đăng nhập (Login Route):**

**- Kiểm tra dữ liệu đầu vào: Hệ thống kiểm tra xem Email và Password có bị bỏ trống hay không trước khi gửi yêu cầu xác thực. Nếu thiếu thông tin, hệ thống sẽ hiển thị thông báo "Vui lòng nhập email và mật khẩu" và điều hướng về trang login.**

**- Xác thực và Điều hướng: Sử dụng hàm passport.authenticate với chiến lược 'local'. Nếu thông tin hợp lệ, hệ thống gọi req.logIn để thiết lập phiên làm việc (session) và điều hướng dựa trên vai trò: res.redirect(user.role === "admin" ? "/admin" : "/").**

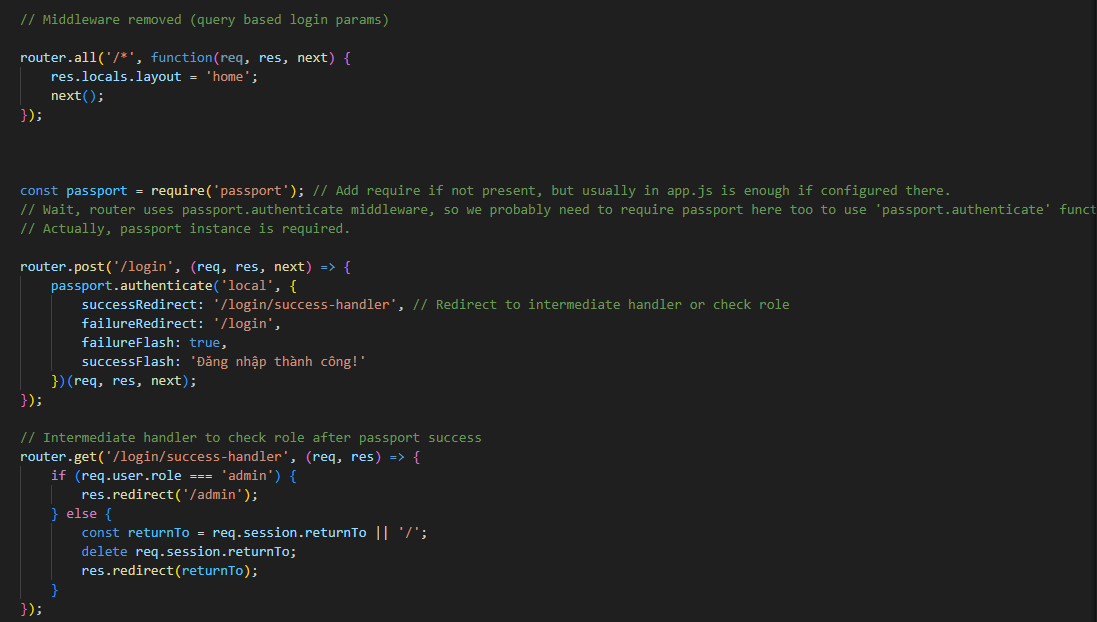
**Quản lý phiên làm việc (Session Management):**

**- serializeUser: Lưu trữ ID của người dùng vào session để duy trì trạng thái đăng nhập giữa các yêu cầu HTTP khác nhau. Điều này giúp hệ thống không cần xác thực lại người dùng ở mỗi request.**

**- deserializeUser: Sử dụng ID từ session để truy vấn thông tin chi tiết của người dùng từ database thông qua User.findById, giúp hệ thống luôn nhận biết được danh tính và quyền hạn của người dùng hiện tại.**

**Xử lý Đăng xuất (Logout):**

**- Phương thức req.logOut được sử dụng để hủy bỏ phiên làm việc của người dùng, xóa thông tin session và điều hướng họ về trang chủ.**

****

**3.3. CHỨC NĂNG XEM DANH SÁCH KHÓA HỌC**

**3.3.1. MÔ TẢ CHỨC NĂNG**

**Chức năng này cho phép người học duyệt xem toàn bộ các khóa học tiếng Anh đang được cung cấp trên hệ thống và xem thông tin chi tiết của từng khóa học. Hệ thống cung cấp các bộ lọc thông minh giúp người dùng tìm kiếm khóa học phù hợp với nhu cầu, ngân sách và trình độ của mình.**

**3.3.2. GIAO DIỆN HIỂN THỊ**

**Trang danh sách khóa học: Hiển thị dưới dạng lưới (Grid Layout) bao gồm hình ảnh thumbnail, tên khóa học, giá tiền (có hiển thị giá gốc nếu có giảm giá), tên giảng viên, đánh giá trung bình và các thông tin cơ bản khác.**

**Bộ lọc nâng cao: Hệ thống cung cấp sidebar hoặc thanh lọc bao gồm:**

**- Lọc theo khoảng giá (minPrice - maxPrice)**

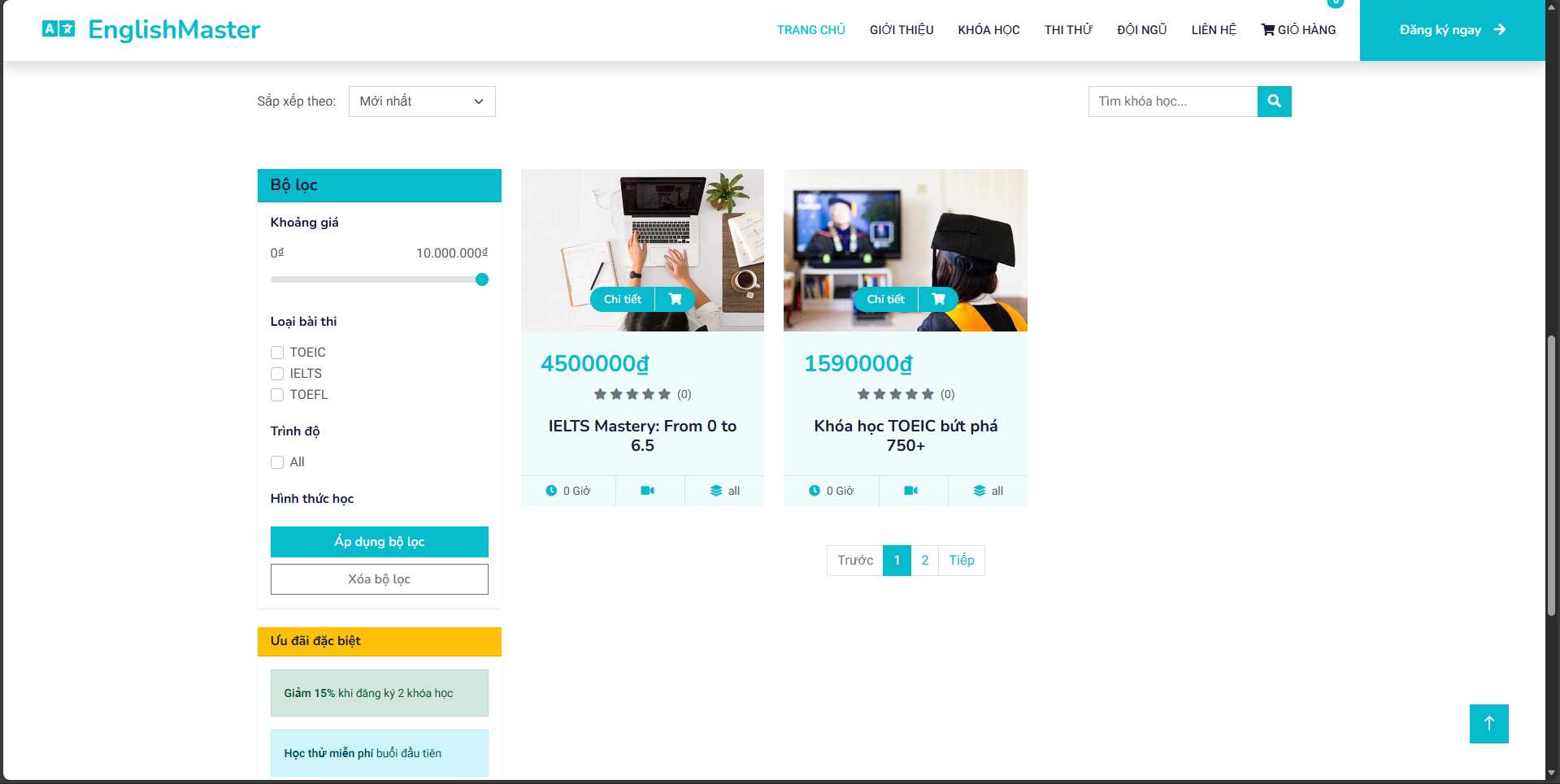
**- Lọc theo danh mục (Category): TOEIC, IELTS, Giao tiếp, Tiếng Anh thiếu nhi...**

**- Lọc theo trình độ (Level): Beginner, Intermediate, Advanced, All levels**

**- Lọc theo hình thức học (Teaching Method): Online, Offline, Hybrid**

**- Tìm kiếm theo từ khóa (Search)**

**- Sắp xếp (Sort): Mới nhất, Giá tăng dần, Giá giảm dần, Phổ biến nhất**

**3.3.3. HIỆN THỰC XỬ LÝ NGHIỆP VỤ (BACKEND)**

**Hệ thống sử dụng các truy vấn nâng cao của Mongoose để tối ưu hóa việc lấy dữ liệu từ nhiều Collection khác nhau:**

**Xử lý lấy danh sách khóa học (Route GET /courses):**

**Xây dựng Query động (Dynamic Query Building):**

**- Hệ thống khởi tạo query cơ bản với điều kiện { isActive: true } để chỉ hiển thị các khóa học đang hoạt động.**

**- Xử lý lọc theo khoảng giá: Nếu có minPrice hoặc maxPrice, hệ thống sẽ thêm điều kiện query.price = { $gte: minPrice, $lte: maxPrice } vào MongoDB query.**

**- Xử lý lọc theo danh mục: Chuyển đổi chuỗi category thành mảng và sử dụng toán tử $in của MongoDB để tìm các khóa học thuộc bất kỳ danh mục nào được chọn.**

**- Xử lý lọc theo trình độ và hình thức học tương tự với toán tử $in.**

**- Xử lý tìm kiếm: Sử dụng Regular Expression với cờ 'i' (case-insensitive) để tìm kiếm khóa học theo tiêu đề: query.title = { $regex: search, $options: 'i' }.**

**Xử lý sắp xếp (Sorting):**

**- Mặc định sắp xếp theo thời gian tạo mới nhất: { createdAt: -1 }**

**- Nếu sort === 'price-asc': sắp xếp theo giá tăng dần { price: 1 }**

**- Nếu sort === 'price-desc': sắp xếp theo giá giảm dần { price: -1 }**

**- Nếu sort === 'popularity': sắp xếp theo đánh giá trung bình { averageRating: -1 }**

**Truy vấn đồng thời (Parallel Queries):**

**- Sử dụng Promise.all để lấy dữ liệu từ ba bộ sưu tập cùng một lúc:**

**\* Course.find(query).sort(sortQuery).populate('instructor').lean()**

**\* Category.find({ isActive: true }).lean()**

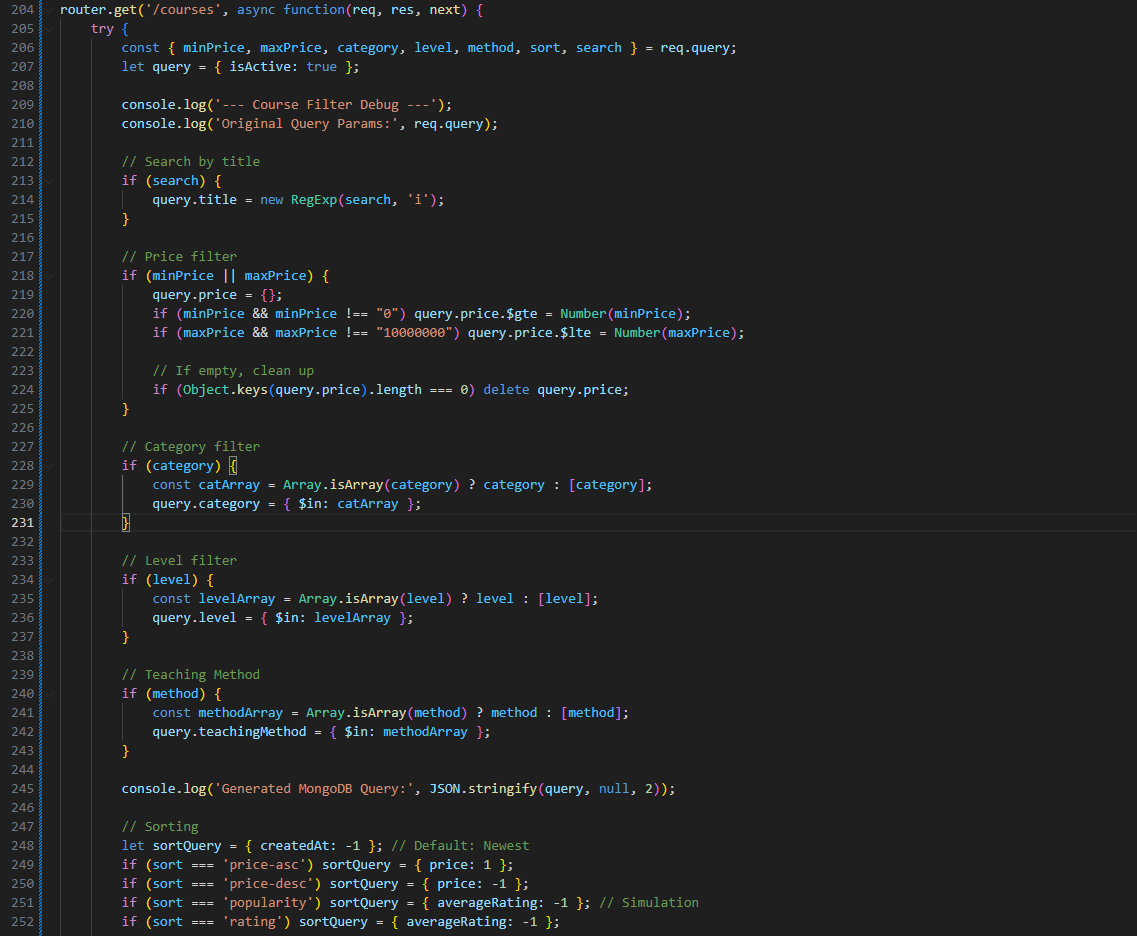
**\* Các giá trị enum cho levels và methods**

**- Việc truy vấn song song này giúp giảm thời gian phản hồi của trang web.**

**Tối ưu hiệu năng:**

**- Sử dụng phương thức .lean() để chuyển đổi kết quả từ tài liệu Mongoose sang đối tượng JavaScript thuần túy, giúp tăng tốc độ xử lý dữ liệu và giảm bộ nhớ sử dụng.**

**- Sử dụng .populate('instructor') để lấy thông tin giảng viên liên kết với khóa học, tránh phải thực hiện query riêng biệt.**

****

****

**3.4. CHỨC NĂNG QUẢN LÝ NGƯỜI DÙNG (ADMIN)**

**3.4.1. MÔ TẢ CHỨC NĂNG**

**Chức năng quản lý người dùng giúp quản trị viên kiểm soát toàn bộ danh sách tài khoản tham gia hệ thống. Admin có quyền xem thông tin, chỉnh sửa phân quyền (Role) giữa người dùng thông thường (user) và quản trị viên (admin), hoặc vô hiệu hóa các tài khoản vi phạm chính sách của hệ thống.**

**3.4.2. GIAO DIỆN QUẢN TRỊ NGƯỜI DÙNG**

**Trang danh sách người dùng:**

**- Hiển thị bảng tổng hợp gồm: STT, Họ tên, Email, Số điện thoại, Vai trò và Trạng thái.**

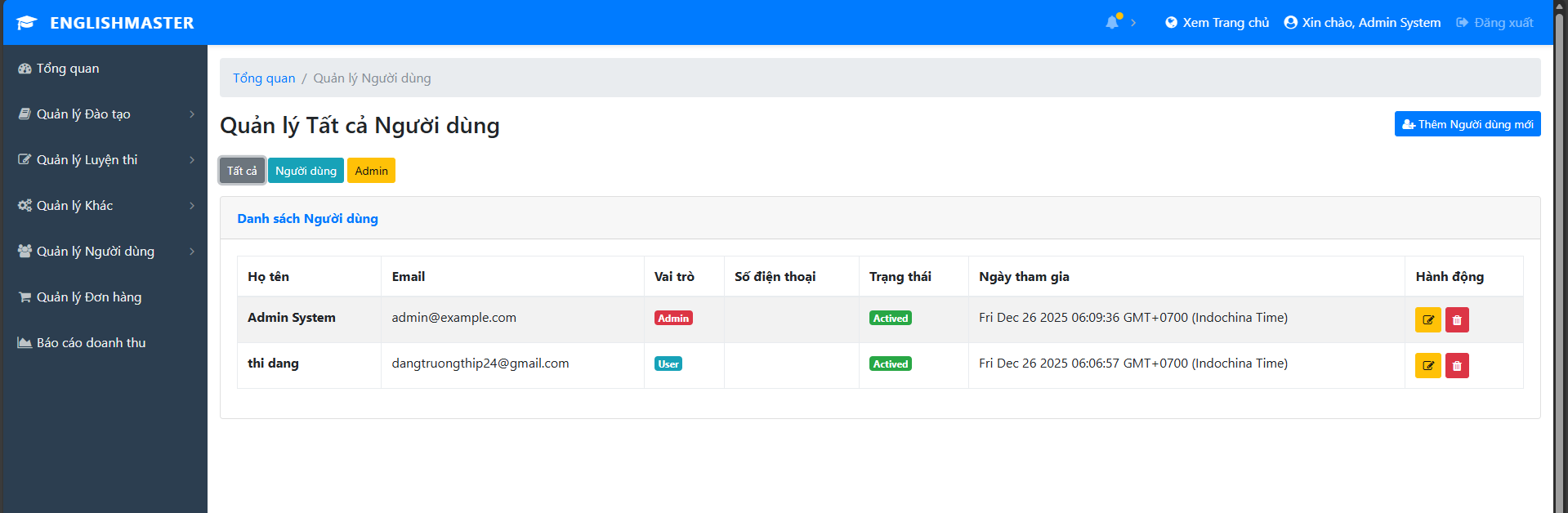
**- Các vai trò được phân biệt bằng nhãn màu (badge): Admin (màu đỏ), User (màu xanh).**

**- Trạng thái hoạt động được hiển thị bằng biểu tượng: Active (dấu tick xanh), Inactive (dấu X đỏ).**

**- Các nút thao tác: Sửa (Edit), Xóa (Delete) cho mỗi người dùng.**

**- Nút "Thêm người dùng mới" ở đầu trang.**

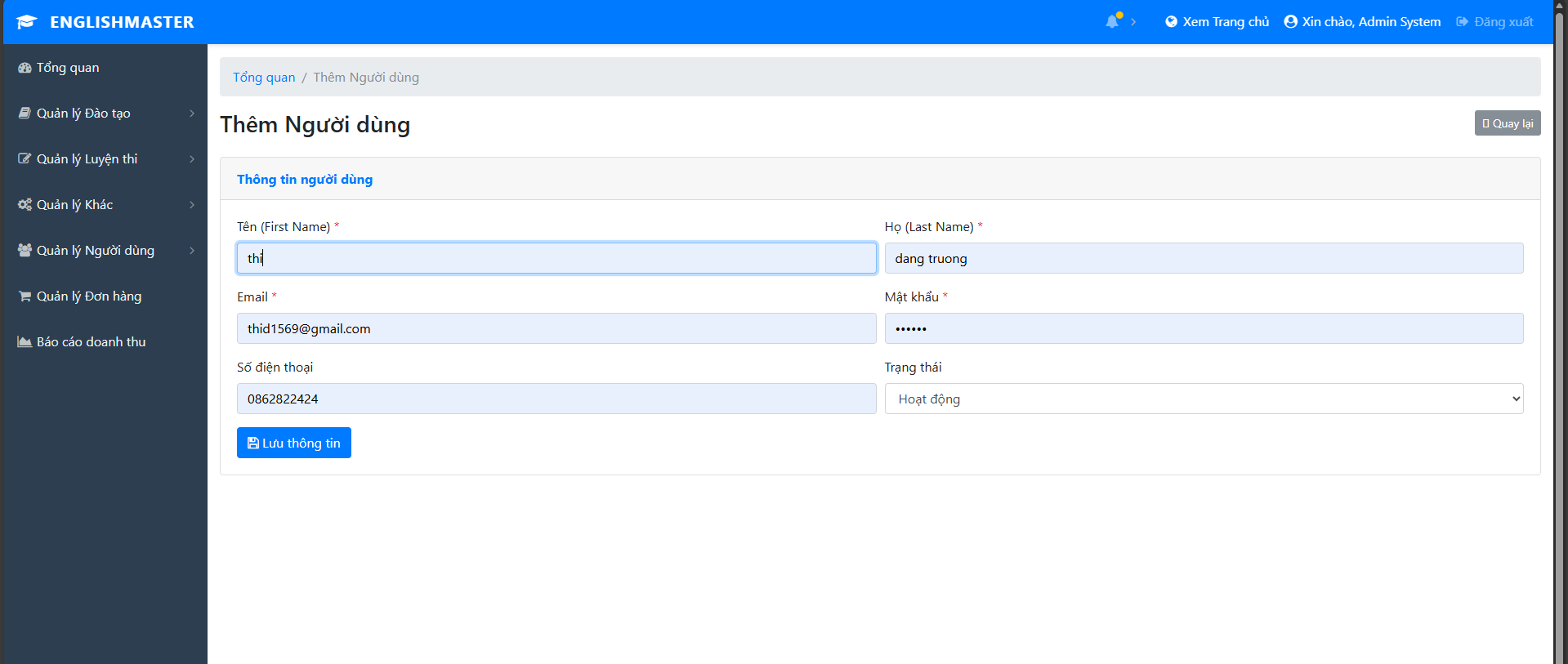
**- Bộ lọc theo vai trò: Tất cả / Chỉ Admin / Chỉ User.**

**Trang thêm/sửa người dùng:**

**- Form nhập liệu bao gồm: Họ (Last Name), Tên (First Name), Email, Mật khẩu, Số điện thoại.**

**- Dropdown chọn vai trò: User / Admin.**

**- Checkbox trạng thái hoạt động: Active / Inactive.**

**3.4.3. HIỆN THỰC XỬ LÝ NGHIỆP VỤ (BACKEND)**

**Middleware phân quyền (isAdmin):**

**- Kiểm tra req.session.isAuthenticated và req.session.user.role === 'admin'**

**- Nếu không phải admin, điều hướng về trang chủ với thông báo lỗi**

**- Middleware này được áp dụng cho tất cả routes trong admin**

**Xử lý hiển thị danh sách (GET /admin/users):**

**- Hệ thống hỗ trợ lọc theo vai trò qua query parameter: req.query.role**

**- Sử dụng User.find(query).sort({ createdAt: -1 }) để truy xuất danh sách người dùng**

**- Dữ liệu được xử lý qua .map() để bổ sung: stt, fullName, isAdmin, isUser**

**Chức năng thêm mới (POST /admin/users/add):**

**- Validation: Kiểm tra các trường bắt buộc**

**- Kiểm tra email trùng lặp: User.findOne({ email: email })**

**- Mã hóa mật khẩu: bcryptjs.genSalt(10) và bcryptjs.hash()**

**- Tạo User mới và lưu vào database**

**Chức năng cập nhật (PUT /admin/users/edit/:id):**

**- Tìm người dùng: User.findById(req.params.id)**

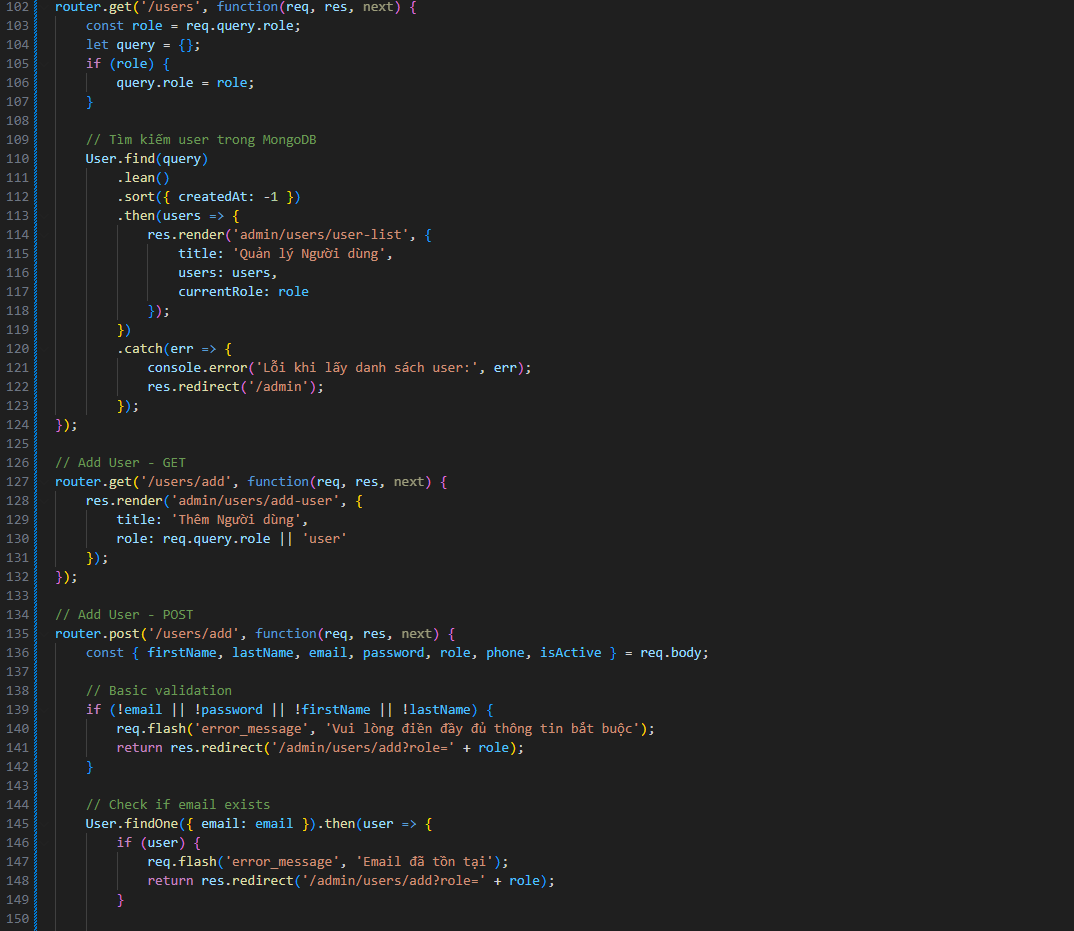
**- Cập nhật các trường: firstName, lastName, email, phone, role, isActive**

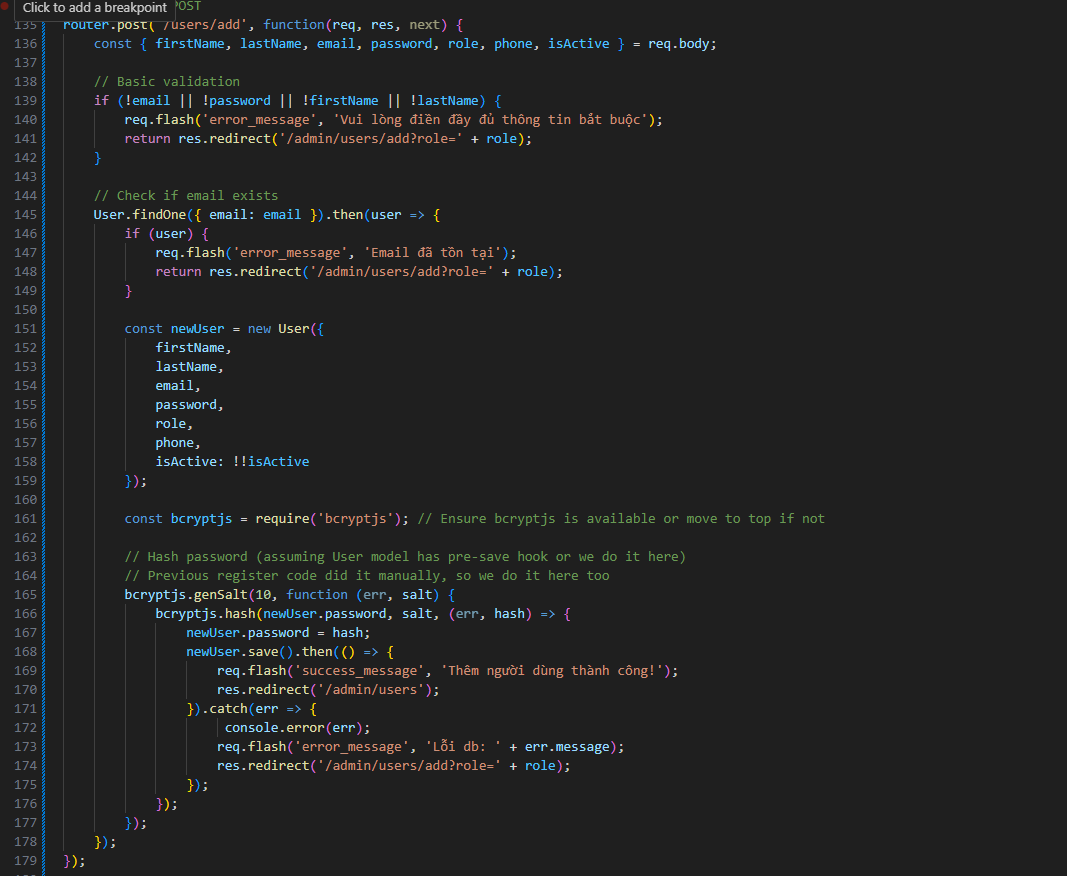
**- Xử lý mật khẩu: chỉ cập nhật nếu người dùng nhập mật khẩu mới**

**Chức năng xóa (POST /admin/users/delete/:id):**

**- Sử dụng User.findByIdAndDelete(req.params.id)**

**- Toàn bộ quá trình được đặt trong khối try...catch**

****

**3.5. CHỨC NĂNG QUẢN LÝ KHÓA HỌC (ADMIN)**

**3.5.1. MÔ TẢ CHỨC NĂNG**

**Chức năng quản lý khóa học cho phép Admin thực hiện đầy đủ các thao tác CRUD đối với danh mục khóa học của hệ thống. Admin có thể kiểm soát nội dung đào tạo, cập nhật giá cả, thông tin giảng viên và phân loại khóa học vào các danh mục tương ứng.**

**3.5.2. GIAO DIỆN QUẢN TRỊ**

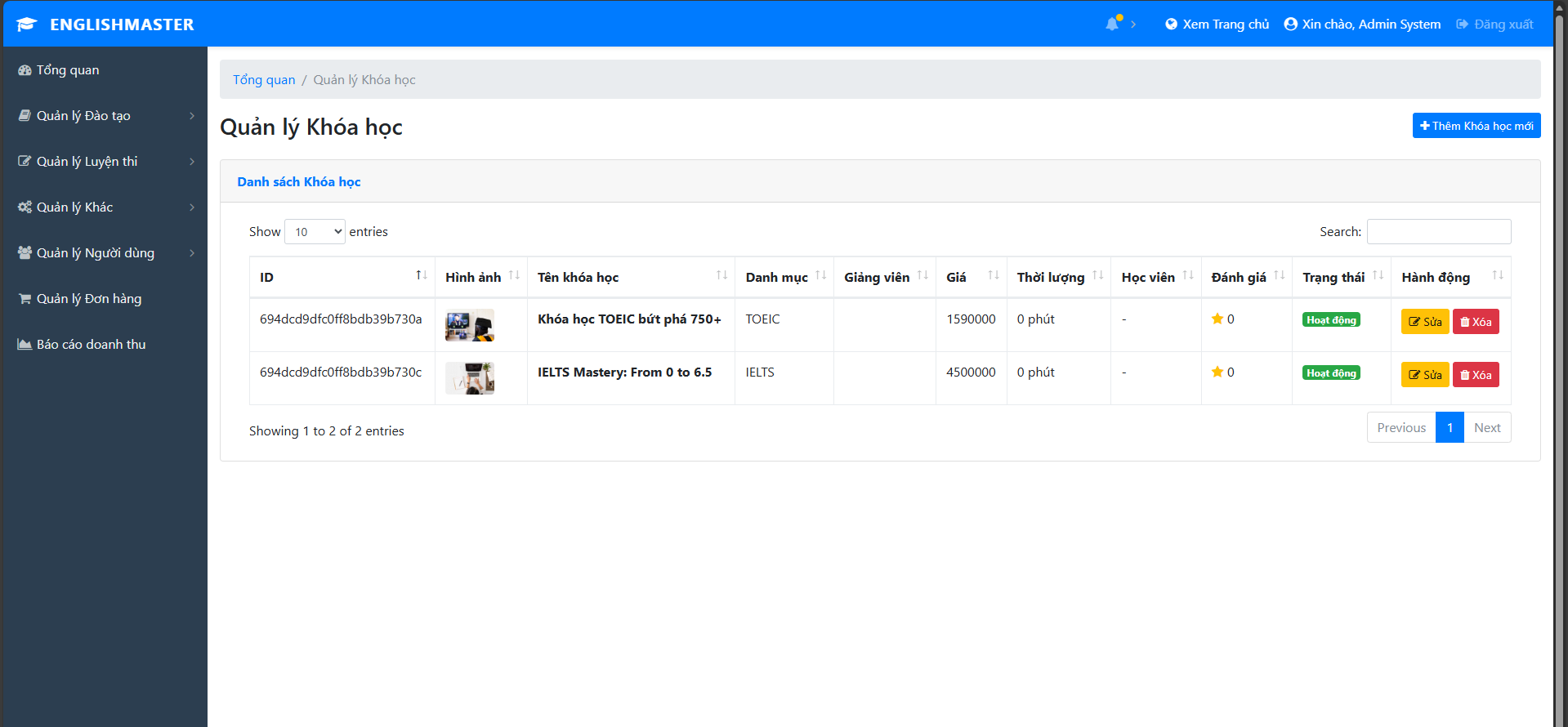
**Trang danh sách khóa học:**

**- Hiển thị bảng với các cột: STT, Hình ảnh, Tên khóa học, Danh mục, Giảng viên, Giá, Trạng thái**

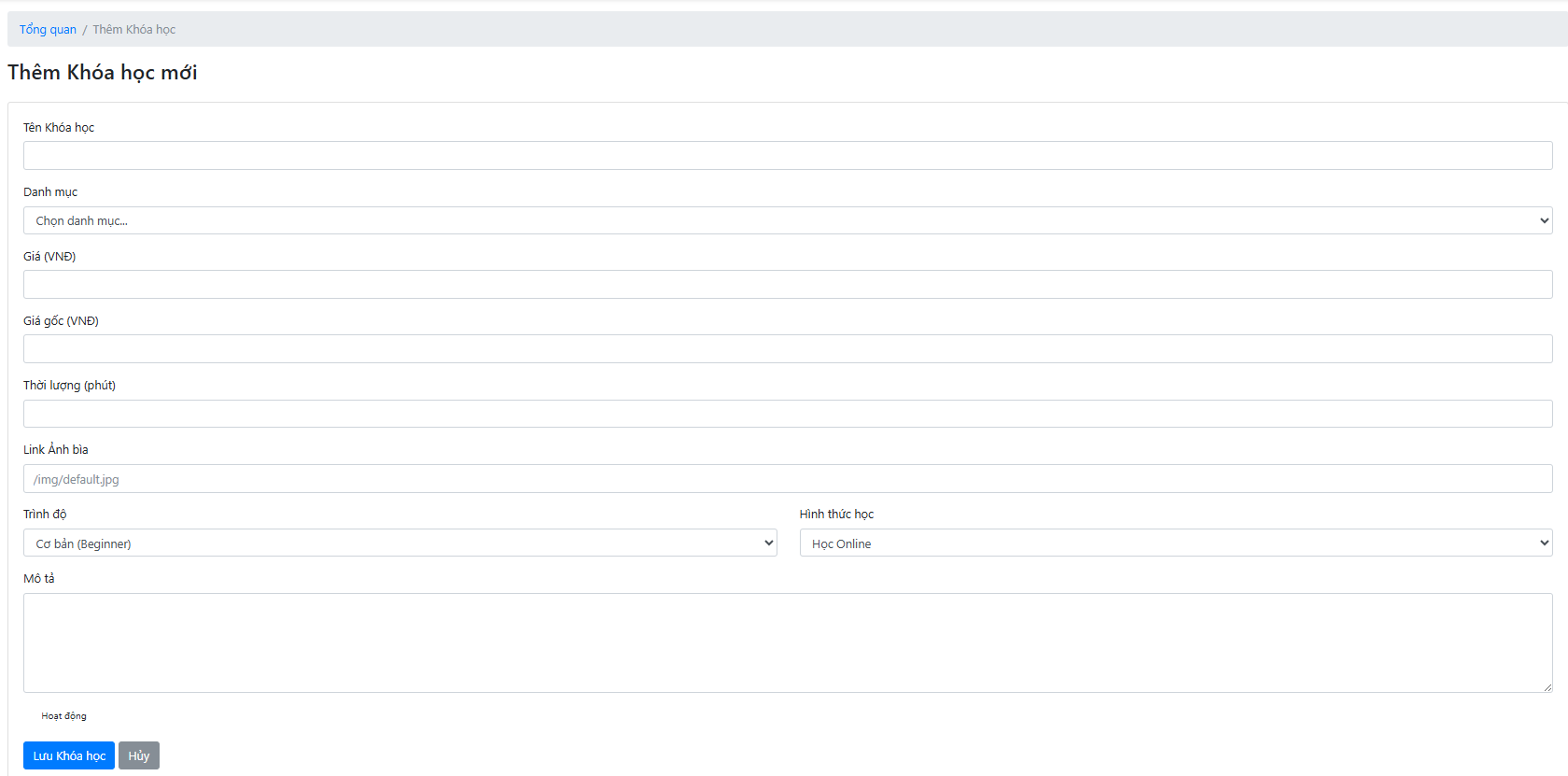
**- Trạng thái được phân biệt bằng nhãn màu (Đang bán, Ngừng bán)**

**- Các nút thao tác: Xem chi tiết, Sửa, Xóa**

**- Nút "Thêm khóa học mới" ở đầu trang**

**Trang thêm/sửa khóa học:**

**- Form nhập liệu: Tên khóa học, Mô tả, Danh mục, Giảng viên, Giá bán, Giá gốc, Thời lượng, Số bài học, Trình độ, Hình thức học, Hình ảnh, Trạng thái**

**3.5.3. HIỆN THỰC XỬ LÝ NGHIỆP VỤ (BACKEND)**

**Xử lý hiển thị danh sách (GET /admin/product):**

**- Truy vấn: Course.find().sort({ createdAt: -1 })**

**- Sử dụng .populate('category') và .populate('instructor')**

**- Xử lý dữ liệu: thêm stt, formattedPrice, isActive, isInactive**

**Chức năng thêm mới (POST /admin/product/add):**

**- Validation: Kiểm tra các trường bắt buộc**

**- Xử lý giá trị mặc định: originalPrice, thumbnail, level, teachingMethod**

**- Tạo Course mới và lưu vào database**

**Chức năng cập nhật (PUT /admin/product/edit/:id):**

**- Tìm khóa học: Course.findById(req.params.id)**

**- Cập nhật các trường từ req.body**

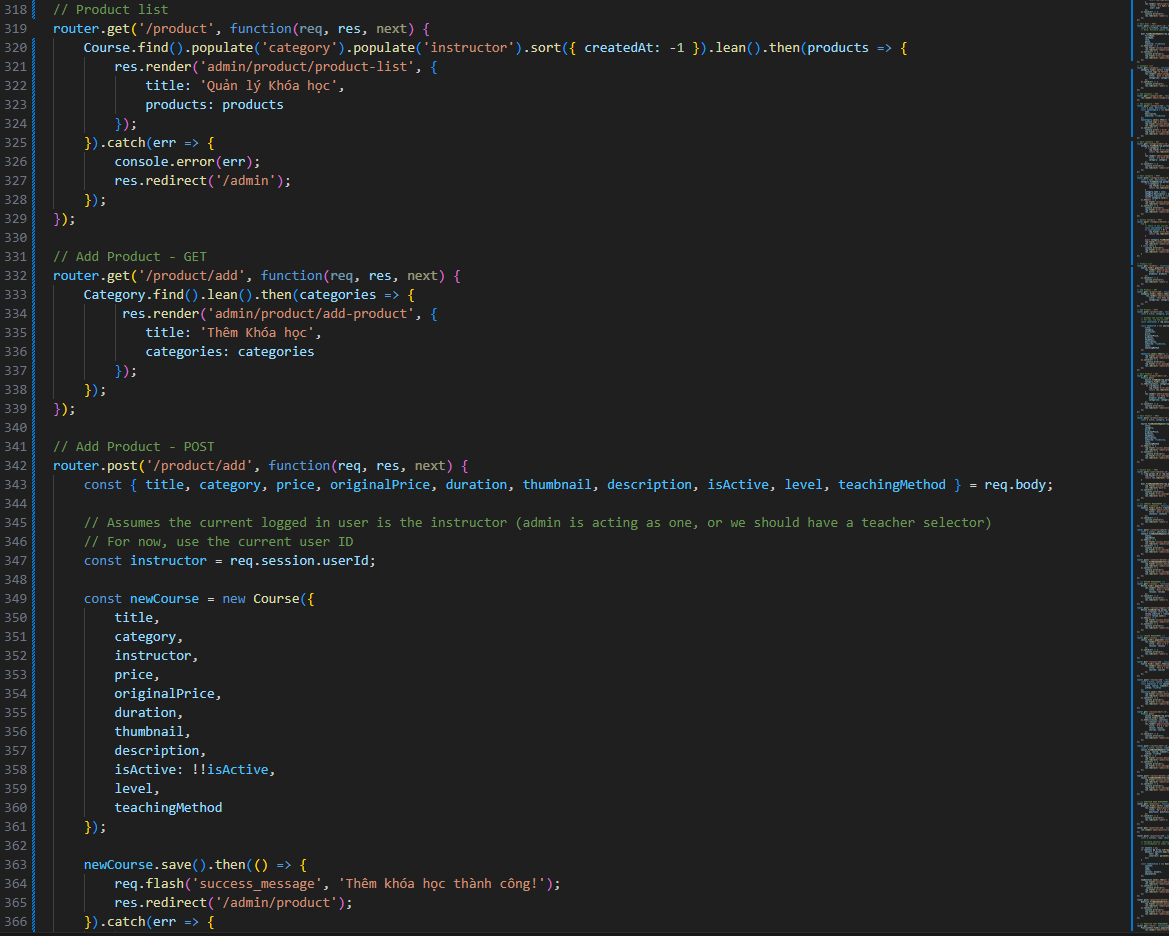
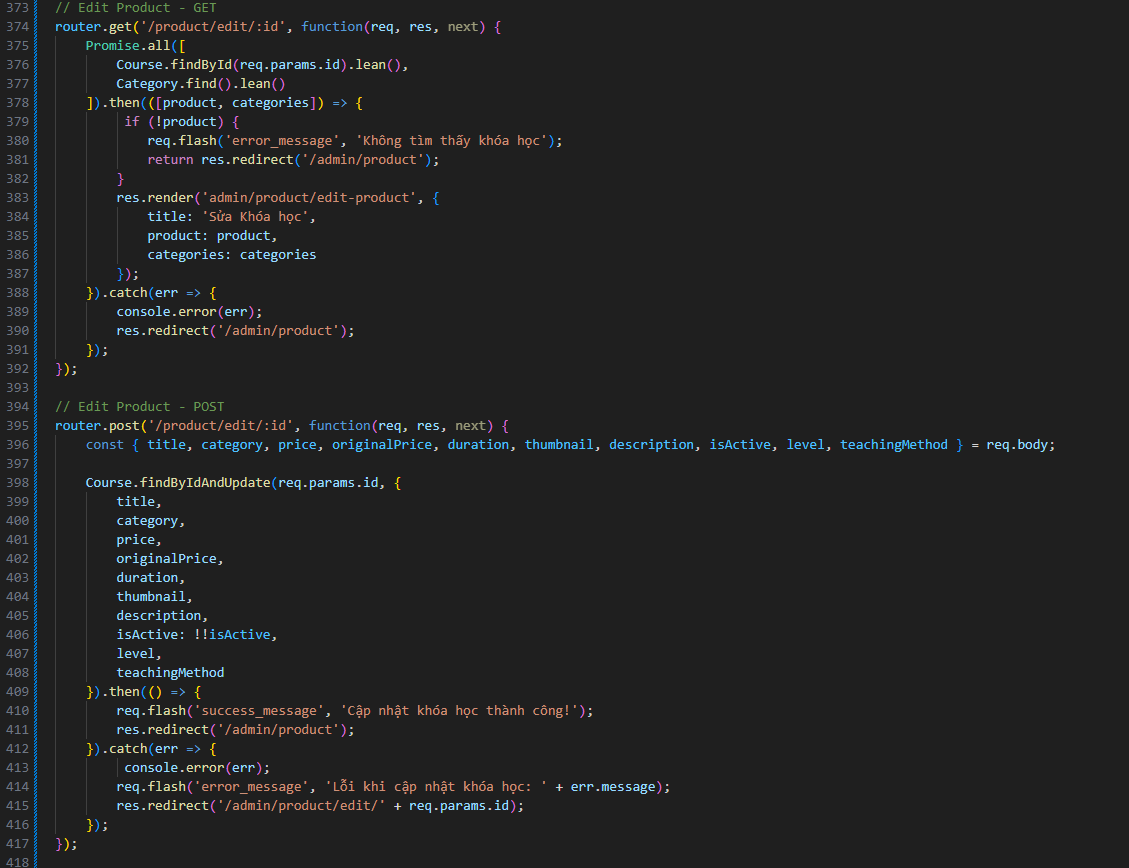
**- Xử lý chuyển đổi kiểu dữ liệu: Number, Boolean**

**Chức năng xóa (POST /admin/product/delete/:id):**

**- Kiểm tra ràng buộc: Order.countDocuments({ 'items.course': req.params.id })**

**- Nếu có đơn hàng, không cho phép xóa**

**- Nếu không có ràng buộc: Course.findByIdAndDelete()**

**** 

**3.6. CHỨC NĂNG QUẢN LÝ ĐƠN HÀNG (ADMIN)**

**3.6.1. MÔ TẢ CHỨC NĂNG**

**Chức năng quản lý đơn hàng cho phép Admin theo dõi và điều phối toàn bộ hoạt động giao dịch của hệ thống. Admin có thể kiểm tra danh sách đơn hàng, xác nhận thanh toán, cập nhật trạng thái đơn hàng hoặc xử lý các yêu cầu hủy đơn từ khách hàng.**

**3.6.2. GIAO DIỆN QUẢN TRỊ ĐƠN HÀNG**

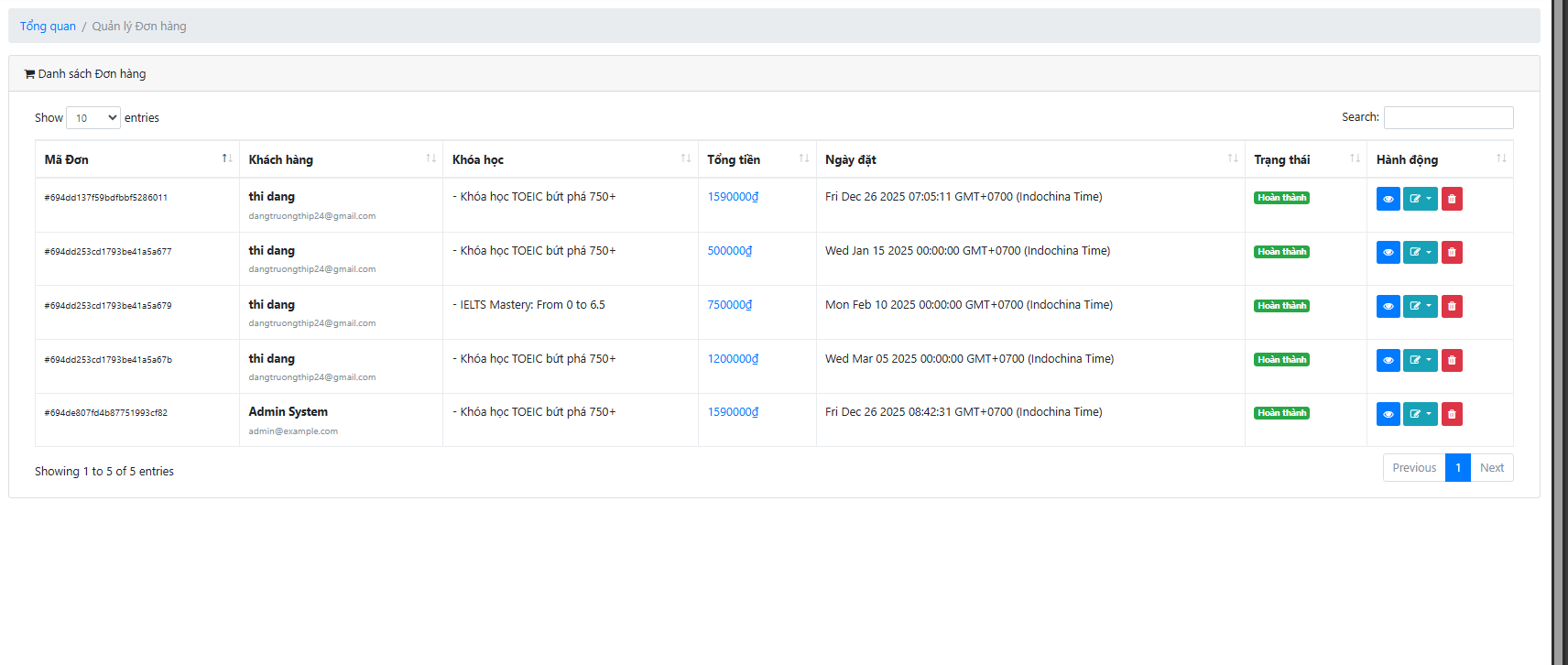
**Trang danh sách đơn hàng:**

**- Hiển thị bảng: Mã đơn hàng, Tên khách hàng, Email, Ngày đặt, Tổng tiền, Phương thức thanh toán, Trạng thái**

**- Hệ thống trạng thái: Pending (vàng), Completed (xanh), Cancelled (đỏ)**

**- Các nút thao tác: Xem chi tiết, Cập nhật trạng thái, Xóa**

**- Bộ lọc theo trạng thái và khoảng thời gian**

****

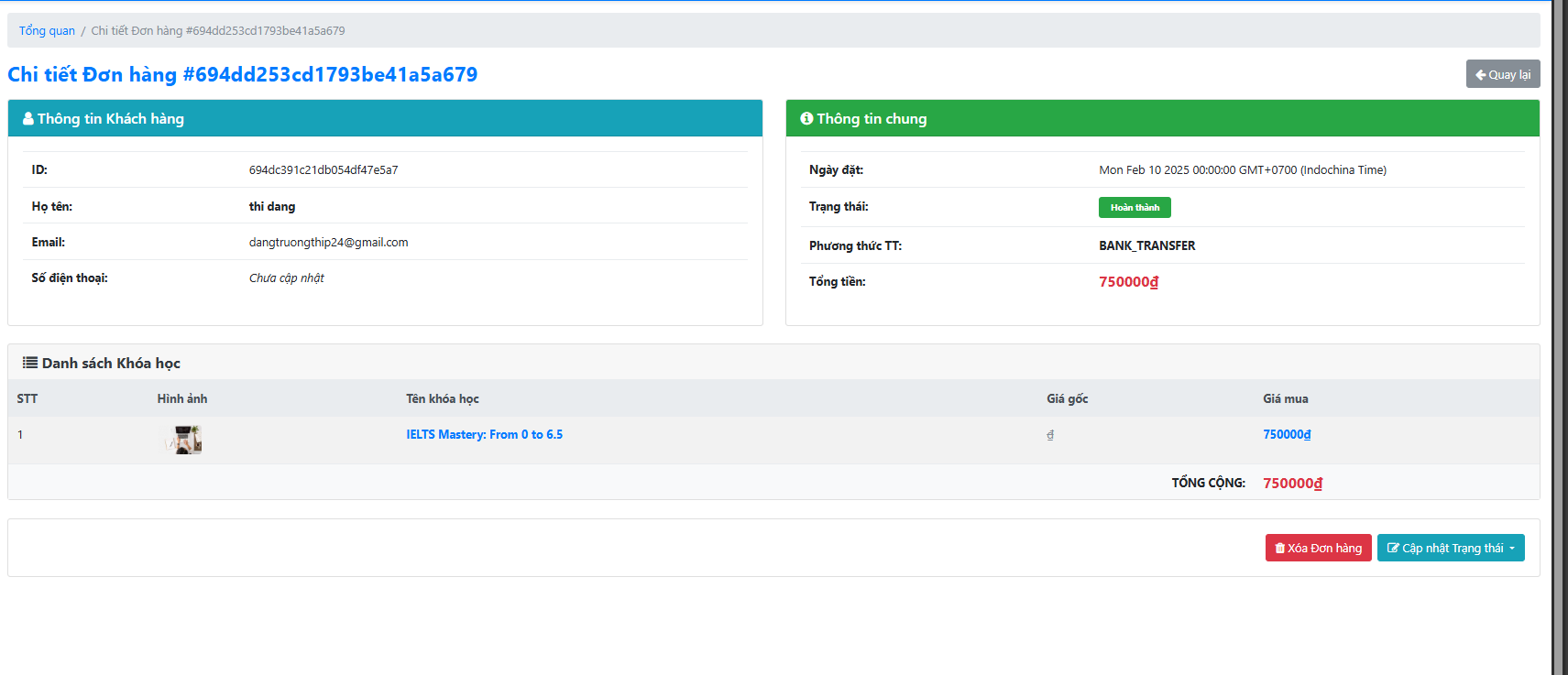
**Trang chi tiết đơn hàng:**

**- Thông tin khách hàng**

**- Danh sách khóa học trong đơn**

**- Tổng tiền, phương thức thanh toán, trạng thái**

**- Form cập nhật trạng thái**

**3.6.3. HIỆN THỰC XỬ LÝ NGHIỆP VỤ (BACKEND)**

**Xác thực và Định dạng dữ liệu (GET /admin/order):**

**- Truy vấn với populate lồng nhau: .populate('user').populate('items.course')**

**- Định dạng: ngày tháng (toLocaleDateString), tiền tệ (toLocaleString)**

**- Tính toán thống kê: tổng số đơn, đơn chờ xử lý, tổng doanh thu**

**Xử lý lọc và tìm kiếm:**

**- Lọc theo trạng thái: query.status = req.query.status**

**- Lọc theo khoảng thời gian: query.createdAt = { $gte, $lte }**

**- Tìm kiếm theo tên khách hàng hoặc email**

**Cập nhật trạng thái (PUT /admin/order/edit/:id):**

**- Tìm đơn hàng: Order.findById(req.params.id)**

**- Cập nhật trạng thái: order.status = req.body.status**

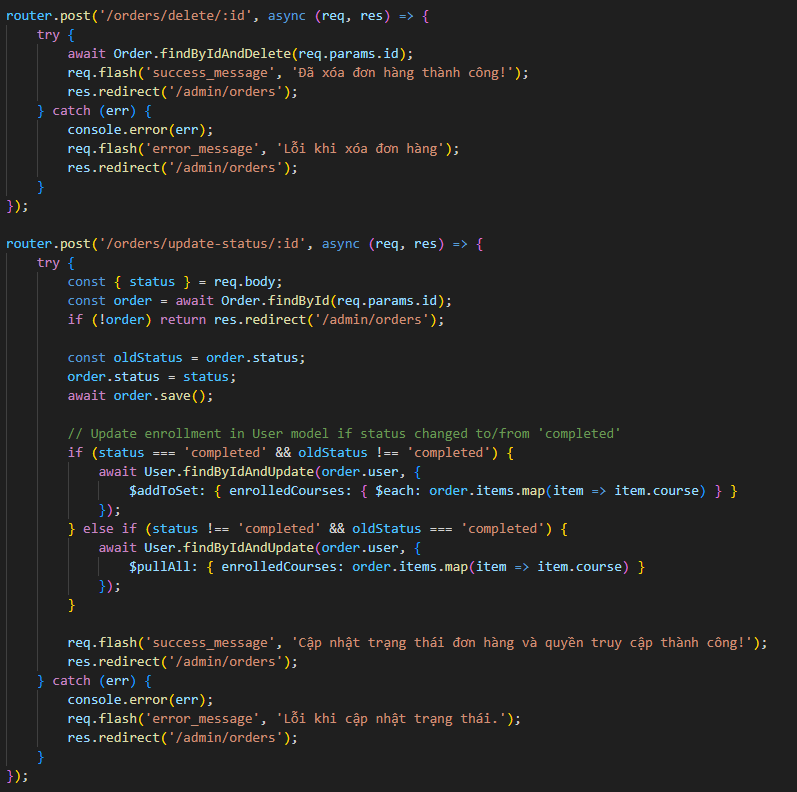
**- Logic nghiệp vụ khi chuyển trạng thái completed/cancelled**

**Xóa đơn hàng (POST /admin/order/delete/:id):**

**- Kiểm tra điều kiện: chỉ cho phép xóa đơn cancelled**

**- Xóa: Order.findByIdAndDelete(req.params.id)**

****

**CHƯƠNG 4: KẾT LUẬN**

**4.1. KẾT QUẢ ĐẠT ĐƯỢC**

**Sau quá trình nghiên cứu, tìm hiểu công nghệ và triển khai thực hiện đề tài "Xây dựng Website Học tiếng Anh trực tuyến EnglishMaster", em đã hoàn thành các mục tiêu đề ra ban đầu và đạt được những kết quả cụ thể như sau:**

**4.1.1.Về mặt kỹ thuật:**

**Kiến trúc hệ thống:**

**- Xây dựng thành công ứng dụng web dựa trên môi trường Node.js và framework Express**

**- Áp dụng kiến trúc MVC (Model-View-Controller) giúp tách biệt rõ ràng giữa các tệp định nghĩa dữ liệu (Models), logic xử lý (Routes/Controllers) và giao diện hiển thị (Views)**

**Cơ sở dữ liệu:**

**- Thiết kế và kết nối thành công với cơ sở dữ liệu MongoDB thông qua thư viện Mongoose**

**- Đảm bảo lưu trữ linh hoạt thông tin khóa học, người dùng, đơn hàng và đề thi**

**- Tối ưu hóa khả năng truy xuất dữ liệu thông qua các phương thức find(), save(), findById() và populate()**

**Bảo mật:**

**- Tích hợp thành công Passport.js với chiến lược xác thực cục bộ (Local Strategy)**

**- Đảm bảo mật khẩu người dùng được mã hóa an toàn bằng thư viện bcryptjs (với salt rounds là 10)**

**- Triển khai phân quyền truy cập (Role-based access control) để ngăn chặn người dùng thường vào trang quản trị**

**Giao diện:**

**- Sử dụng công cụ mẫu Handlebars kết hợp với Bootstrap và CSS tùy chỉnh**

**- Tạo ra giao diện hiện đại, sang trọng, nhất quán qua cơ chế Layout chung**

**- Đảm bảo tính tương thích (Responsive) trên các thiết bị máy tính và tablet**

**4.1.2.Về mặt chức năng:**

**Phân quyền hệ thống:**

**- Hoàn thiện luồng quy trình từ tìm hiểu thông tin khóa học, xem chi tiết đến đăng ký/đăng nhập và thực hiện đặt mua khóa học**

**- Hệ thống dựa trên thông tin phiên làm việc (session) của người dùng để cá nhân hóa trải nghiệm**

**Quản lý toàn diện:**

**- Triển khai thành công bảng điều khiển thống kê trực quan (Dashboard)**

**- Các chức năng quản lý cốt lõi (CRUD) cho Người dùng, Khóa học, Danh mục, Đơn hàng, Đề thi và Câu hỏi**

**- Cho phép vận hành hệ thống một cách hiệu quả và chuyên nghiệp**

**4.1.3 Về mặt trải nghiệm người dùng:**

**Trang chủ:**

**- Hiển thị ấn tượng với khu vực Banner (Hero Section) hiện đại**

**- Cung cấp thông điệp chào mừng và giới thiệu về hệ thống**

**- Hiển thị các khóa học phổ biến và đánh giá từ học viên**

**Trang danh sách khóa học:**

**- Cho phép người dùng duyệt qua toàn bộ hệ thống khóa học theo dạng lưới (Grid Layout) đẹp mắt**

**- Đi kèm bộ lọc thông minh theo giá, danh mục, trình độ, hình thức học**

**- Thông tin giá tiền được định dạng chuẩn và các nút chức năng trực quan**

**Trang chi tiết khóa học:**

**- Cung cấp đầy đủ các thông số: mô tả, giá cả, giảng viên, thời lượng, số bài học**

**- Hiển thị các đánh giá từ học viên đã tham gia**

**- Nút "Thêm vào giỏ hàng" để bắt đầu quy trình đăng ký**

**Quy trình đặt mua khóa học:**

**- Tích hợp giỏ hàng thông minh, tự động nhận diện thông tin người dùng đang đăng nhập**

**- Hỗ trợ nhiều phương thức thanh toán: Chuyển khoản, COD, MoMo, VNPay**

**- Tự động cập nhật danh sách khóa học đã đăng ký vào hồ sơ người dùng**

**Hệ thống làm bài thi:**

**- Phân biệt rõ ràng đề thi miễn phí và trả phí**

**- Tự động chấm điểm và lưu kết quả vào hồ sơ học tập**

**- Hiển thị chi tiết đáp án đúng và giải thích sau khi hoàn thành**

**Hệ thống điều hướng:**

**- Thanh Menu tự động thay đổi trạng thái hiển thị dựa trên phiên làm việc**

**- Tối ưu hóa tính cá nhân hóa cho trải nghiệm người dùng**

**4.1.4.Về mặt ứng dụng thực tiễn:**

**Số hóa quy trình quản lý:**

**- Tạo ra một nền tảng số hóa hiện đại giúp thay thế phương pháp quản lý thông tin khóa học và danh sách đăng ký thủ công**

**- Hệ thống cung cấp công cụ cho phép quản trị viên cập nhật thông tin khóa học, giá cả và quản lý đơn hàng một cách chính xác, nhanh chóng**

**Tiếp cận dịch vụ linh hoạt:**

**- Giúp học viên dễ dàng tiếp cận và tìm hiểu toàn bộ hệ thống khóa học tiếng Anh**

**- Người dùng có thể truy cập mọi lúc mọi nơi thông qua thiết bị di động hoặc máy tính**

**- Không cần phải gọi điện hay đến trực tiếp để kiểm tra thông tin khóa học**

**Tối ưu hóa quy trình giao dịch:**

**- Hệ thống hỗ trợ người dùng chủ động kiểm tra giá cả, nội dung khóa học và giảng viên**

**- Thực hiện quy trình đăng ký khóa học trực tuyến chỉ với vài thao tác đơn giản**

**- Tiết kiệm thời gian cho cả học viên và nhân viên quản lý**

**4.2. HẠN CHẾ CỦA ĐỀ TÀI**

**Mặc dù đã hoàn thiện các chức năng cốt lõi cho việc vận hành một website học tiếng Anh trực tuyến, nhưng do giới hạn về thời gian và kiến thức, đề tài vẫn còn tồn tại một số hạn chế:**

**Quy trình thanh toán:**

**- Hiện tại hệ thống mới chỉ dừng lại ở việc tạo đơn đặt mua khóa học trực tuyến**

**- Chức năng tích hợp cổng thanh toán trực tuyến (như VNPay, Momo) chưa được hiện thực hóa hoàn chỉnh**

**- Chưa có quy trình tự động xác nhận thanh toán và kích hoạt khóa học**

**Tương tác thời gian thực:**

**- Website chưa có tính năng thông báo thời gian thực (Real-time notification)**

**- Chưa gửi thông báo đến quản trị viên khi có học viên vừa đăng ký khóa học mới**

**- Chưa gửi thông báo nhắc lịch học hoặc deadline bài tập cho học viên qua Email/SMS**

**Quản lý nội dung học tập:**

**- Hệ thống hiện tại chưa có chức năng quản lý chi tiết nội dung bài học (video, tài liệu, bài tập)**

**- Chưa có công cụ theo dõi tiến trình học tập của từng học viên**

**- Chưa có hệ thống chứng chỉ tự động khi hoàn thành khóa học**

**Thống kê báo cáo chuyên sâu:**

**- Mặc dù giao diện trang Dashboard quản trị đã có các thành phần hiển thị dữ liệu mẫu**

**- Hệ thống vẫn chưa có các báo cáo phân tích sâu dựa trên dữ liệu thực tế**

**- Chưa có báo cáo về doanh thu thực nhận, tỷ lệ hoàn thành khóa học, xu hướng đăng ký**

**Tối ưu hóa nền tảng di động:**

**- Website hiển thị tốt trên trình duyệt máy tính**

**- Chưa được tối ưu hóa giao diện hoàn chỉnh cho các thiết bị di động nhỏ (Mobile Responsive)**

**- Chưa xây dựng ứng dụng di động (Mobile App) riêng biệt**

**Hệ thống tương tác học viên - giảng viên:**

**- Chưa có chức năng chat trực tiếp giữa học viên và giảng viên**

**- Chưa có diễn đàn thảo luận hoặc hệ thống hỏi đáp**

**- Chưa có tính năng đánh giá chi tiết từng bài học**

**Quản lý ngân hàng câu hỏi:**

**- Hệ thống quản lý câu hỏi còn đơn giản**

**- Chưa hỗ trợ nhiều loại câu hỏi phức tạp (nghe, nói, viết)**

**- Chưa có công cụ tạo đề thi tự động từ ngân hàng câu hỏi**

**CHƯƠNG 5: HƯỚNG PHÁT TRIỂN CỦA ĐỀ TÀI**

**5.1. NÂNG CẤP TÍNH NĂNG QUẢN TRỊ**

**5.1.1.Hệ thống báo cáo thông minh:**

**- Tích hợp các thư viện biểu đồ chuyên sâu như Chart.js hoặc Highcharts vào trang Dashboard**

**- Thay thế dữ liệu mẫu hiện tại bằng dữ liệu thực tế từ Database**

**- Hiển thị biểu đồ doanh thu theo tháng, quý, năm**

**- Thống kê số lượng học viên đăng ký mới theo thời gian**

**- Phân tích tỷ lệ hoàn thành khóa học và điểm trung bình của học viên**

**5.1.2.Quản lý nội dung học tập:**

**- Xây dựng hệ thống quản lý bài học chi tiết (Lesson Management):**

**\* Upload và quản lý video bài giảng**

**\* Quản lý tài liệu đính kèm (PDF, Word, PowerPoint)**

**\* Tạo và quản lý bài tập cho từng bài học**

**\* Thiết lập thứ tự học tập và điều kiện mở khóa bài học tiếp theo**

**- Công cụ theo dõi tiến trình học tập (Learning Progress Tracking):**

**\* Hiển thị phần trăm hoàn thành khóa học của từng học viên**

**\* Theo dõi thời gian học tập và số lần xem lại bài giảng**

**\* Thống kê điểm số bài tập và bài kiểm tra**

**- Hệ thống chứng chỉ tự động:**

**\* Tạo chứng chỉ điện tử khi học viên hoàn thành khóa học**

**\* Tích hợp mã QR để xác thực chứng chỉ**

**\* Cho phép tải xuống và chia sẻ chứng chỉ trên mạng xã hội**

**5.1.3.Tự động hóa tương tác và chăm sóc học viên:**

**- Hệ thống thông báo tự động:**

**\* Sử dụng thư viện Nodemailer trong Node.js để tự động gửi Email xác nhận ngay khi học viên đăng ký khóa học**

**\* Gửi Email nhắc nhở khi có bài học mới hoặc deadline bài tập sắp đến**

**\* Gửi Email chúc mừng khi hoàn thành khóa học kèm chứng chỉ**

**- Tích hợp thông báo thời gian thực:**

**\* Sử dụng Socket.io để đẩy thông báo tức thời (Real-time notification) đến màn hình quản trị viên**

**\* Thông báo khi có đơn đăng ký mới, câu hỏi từ học viên hoặc đánh giá mới**

**- Chatbot hỗ trợ tự động:**

**\* Tích hợp chatbot AI để trả lời các câu hỏi thường gặp**

**\* Hỗ trợ 24/7 về thông tin khóa học, giá cả, chính sách**

**5.2. MỞ RỘNG TRẢI NGHIỆM NGƯỜI DÙNG**

**5.2.1.Hệ thống Tìm kiếm và Lọc thông minh (Smart Search & Filter):**

**- Cải tiến thanh tìm kiếm sử dụng công nghệ AJAX:**

**\* Gợi ý kết quả khóa học ngay khi người dùng gõ từ khóa**

**\* Tìm kiếm mờ (fuzzy search) để xử lý lỗi chính tả**

**\* Hiển thị lịch sử tìm kiếm gần đây**

**- Tích hợp bộ lọc nâng cao:**

**\* Lọc theo khoảng giá với thanh trượt (price range slider)**

**\* Lọc theo đánh giá sao (rating filter)**

**\* Lọc theo thời lượng khóa học**

**\* Lọc theo giảng viên**

**\* Lưu bộ lọc yêu thích để sử dụng lại**

**5.2.2.Cá nhân hóa trải nghiệm học viên:**

**- Hệ thống gợi ý khóa học (Recommendation System):**

**\* Dựa trên lịch sử xem và khóa học đã đăng ký**

**\* Sử dụng thuật toán collaborative filtering**

**\* Hiển thị "Khóa học tương tự bạn đã xem"**

**\* Gợi ý lộ trình học tập phù hợp với mục tiêu**

**- Tính năng "Khóa học yêu thích" (Wishlist):**

**\* Cho phép học viên lưu lại những khóa học ưng ý**

**\* Thông báo khi khóa học yêu thích có giảm giá**

**\* Chia sẻ danh sách yêu thích với bạn bè**

**- Tự động điền thông tin:**

**\* Điền sẵn thông tin cá nhân từ hồ sơ vào form đăng ký**

**\* Lưu phương thức thanh toán ưa thích**

**\* Ghi nhớ bộ lọc và sắp xếp thường dùng**

**5.2.3.Tương tác cộng đồng:**

**- Hệ thống Bình luận và Đánh giá:**

**\* Cho phép học viên đánh giá sao và viết nhận xét chi tiết**

**\* Chia sẻ hình ảnh và video về trải nghiệm học tập**

**\* Hệ thống "helpful" để đánh giá độ hữu ích của review**

**- Diễn đàn thảo luận (Discussion Forum):**

**\* Tạo không gian trao đổi giữa học viên và giảng viên**

**\* Hệ thống tag và phân loại chủ đề**

**\* Tích hợp markdown editor để định dạng nội dung**

**- Live Chat và Chatbot:**

**\* Hỗ trợ trực tuyến 24/7 qua chatbot AI**

**\* Chuyển sang nhân viên hỗ trợ khi cần thiết**

**\* Lưu lịch sử chat để tra cứu**

**5.2.4.Học tập tương tác:**

**- Tính năng học nhóm (Group Study):**

**\* Tạo phòng học nhóm ảo**

**\* Chia sẻ tài liệu và ghi chú**

**\* Lên lịch học chung**

**- Flashcard và công cụ ghi nhớ:**

**\* Tạo flashcard từ vựng tự động từ bài học**

**\* Hệ thống lặp lại ngắt quãng (Spaced Repetition)**

**\* Trò chơi học tập (gamification)**

**5.3. TÍCH HỢP CÔNG NGHỆ TIÊN TIẾN**

**5.3.1.Thanh toán Online:**

**- Tích hợp cổng thanh toán điện tử:**

**\* VNPay: Thanh toán qua QR code, thẻ ATM, thẻ tín dụng**

**\* MoMo: Thanh toán qua ví điện tử**

**\* ZaloPay: Thanh toán qua ứng dụng Zalo**

**\* PayPal: Thanh toán quốc tế**

**- Hệ thống hoàn tiền tự động:**

**\* Chính sách hoàn tiền trong 7 ngày nếu không hài lòng**

**\* Xử lý yêu cầu hoàn tiền tự động**

**- Quản lý hóa đơn điện tử:**

**\* Tự động xuất hóa đơn VAT**

**\* Gửi hóa đơn qua email**

**\* Lưu trữ lịch sử giao dịch**

**5.3.2.Phát triển Mobile App:**

**- Xây dựng ứng dụng di động native:**

**\* iOS App (Swift/SwiftUI)**

**\* Android App (Kotlin/Jetpack Compose)**

**- Hoặc sử dụng framework đa nền tảng:**

**\* React Native**

**\* Flutter**

**- Tính năng đặc biệt cho Mobile:**

**\* Học offline: Tải video và tài liệu về máy**

**\* Nhận thông báo push**

**\* Quét QR code để truy cập nhanh**

**\* Tích hợp camera để chụp và dịch văn bản**

**5.3.3.Trí tuệ nhân tạo (AI) và Học máy (Machine Learning):**

**- Chatbot AI thông minh:**

**\* Sử dụng GPT hoặc các mô hình NLP để trả lời câu hỏi**

**\* Học từ lịch sử hội thoại để cải thiện**

**- Hệ thống chấm bài tự động:**

**\* Chấm bài viết tiếng Anh tự động**

**\* Đánh giá phát âm qua công nghệ Speech Recognition**

**\* Phân tích lỗi sai và đưa ra gợi ý cải thiện**

**- Cá nhân hóa lộ trình học tập:**

**\* Phân tích điểm mạnh, điểm yếu của học viên**

**\* Đề xuất bài học phù hợp với trình độ**

**\* Dự đoán thời gian hoàn thành khóa học**

**5.3.4.Triển khai trên Cloud:**

**- Đưa hệ thống lên các nền tảng điện toán đám mây:**

**\* AWS (Amazon Web Services): EC2, S3, RDS, CloudFront**

**\* Google Cloud Platform: Compute Engine, Cloud Storage**

**\* Microsoft Azure: Virtual Machines, Blob Storage**

**\* Heroku: Triển khai nhanh cho ứng dụng Node.js**

**- Sử dụng Docker và Kubernetes:**

**\* Containerize ứng dụng với Docker**

**\* Quản lý và scale tự động với Kubernetes**

**\* CI/CD pipeline với GitHub Actions hoặc GitLab CI**

**- Tối ưu hóa hiệu năng:**

**\* Sử dụng CDN để phân phối nội dung nhanh hơn**

**\* Caching với Redis hoặc Memcached**

**\* Load Balancing để phân tải**

**\* Auto-scaling khi lượng truy cập tăng cao**

**5.3.5.Bảo mật nâng cao:**

**- Xác thực đa yếu tố (2FA):**

**\* SMS OTP**

**\* Google Authenticator**

**\* Email verification code**

**- Mã hóa dữ liệu:**

**\* HTTPS/SSL cho toàn bộ website**

**\* Mã hóa dữ liệu nhạy cảm trong database**

**\* Mã hóa end-to-end cho chat**

**- Phòng chống tấn công:**

**\* Rate limiting để chống DDoS**

**\* CSRF protection**

**\* XSS protection**

**\* SQL Injection prevention**

**\* Regular security audits**

**5.3.6.Phân tích dữ liệu (Analytics):**

**- Tích hợp Google Analytics:**

**\* Theo dõi lượng truy cập**

**\* Phân tích hành vi người dùng**

**\* Conversion tracking**

**- Hệ thống báo cáo tùy chỉnh:**

**\* Dashboard quản trị với biểu đồ real-time**

**\* Báo cáo doanh thu chi tiết**

**\* Phân tích hiệu quả marketing**

**\* A/B testing để tối ưu conversion**

**5.4. MỞ RỘNG MÔ HÌNH KINH DOANH**

**5.4.1.Đa dạng hóa sản phẩm:**

**- Khóa học trực tiếp (Live Classes):**

**\* Tích hợp Zoom/Google Meet**

**\* Lịch học theo thời gian thực**

**\* Tương tác trực tiếp với giảng viên**

**- Khóa học 1-1 (Private Tutoring):**

**\* Đặt lịch học riêng với giảng viên**

**\* Nội dung học tập cá nhân hóa**

**\* Giá cao hơn nhưng hiệu quả tốt hơn**

**- Gói đăng ký theo tháng (Subscription):**

**\* Truy cập không giới hạn tất cả khóa học**

**\* Giá ưu đãi cho gói dài hạn**

**\* Tự động gia hạn**

**5.4.2.Chương trình đối tác:**

**- Affiliate Marketing:**

**\* Hoa hồng cho người giới thiệu học viên mới**

**\* Cung cấp link tracking và dashboard**

**\* Thanh toán hoa hồng tự động**

**- Hợp tác với doanh nghiệp:**

**\* Gói đào tạo tiếng Anh cho doanh nghiệp**

**\* Quản lý nhiều tài khoản học viên**

**\* Báo cáo tiến độ cho HR**

**- Hợp tác với trường học:**

**\* Cung cấp nền tảng học tập cho sinh viên**

**\* Tích hợp với hệ thống quản lý học tập (LMS)**

**5.4.3.Mở rộng thị trường:**

**- Đa ngôn ngữ (Internationalization):**

**\* Hỗ trợ tiếng Việt, tiếng Anh, tiếng Trung...**

**\* Tự động phát hiện ngôn ngữ trình duyệt**

**\* Cho phép người dùng chọn ngôn ngữ**

**- Đa tiền tệ:**

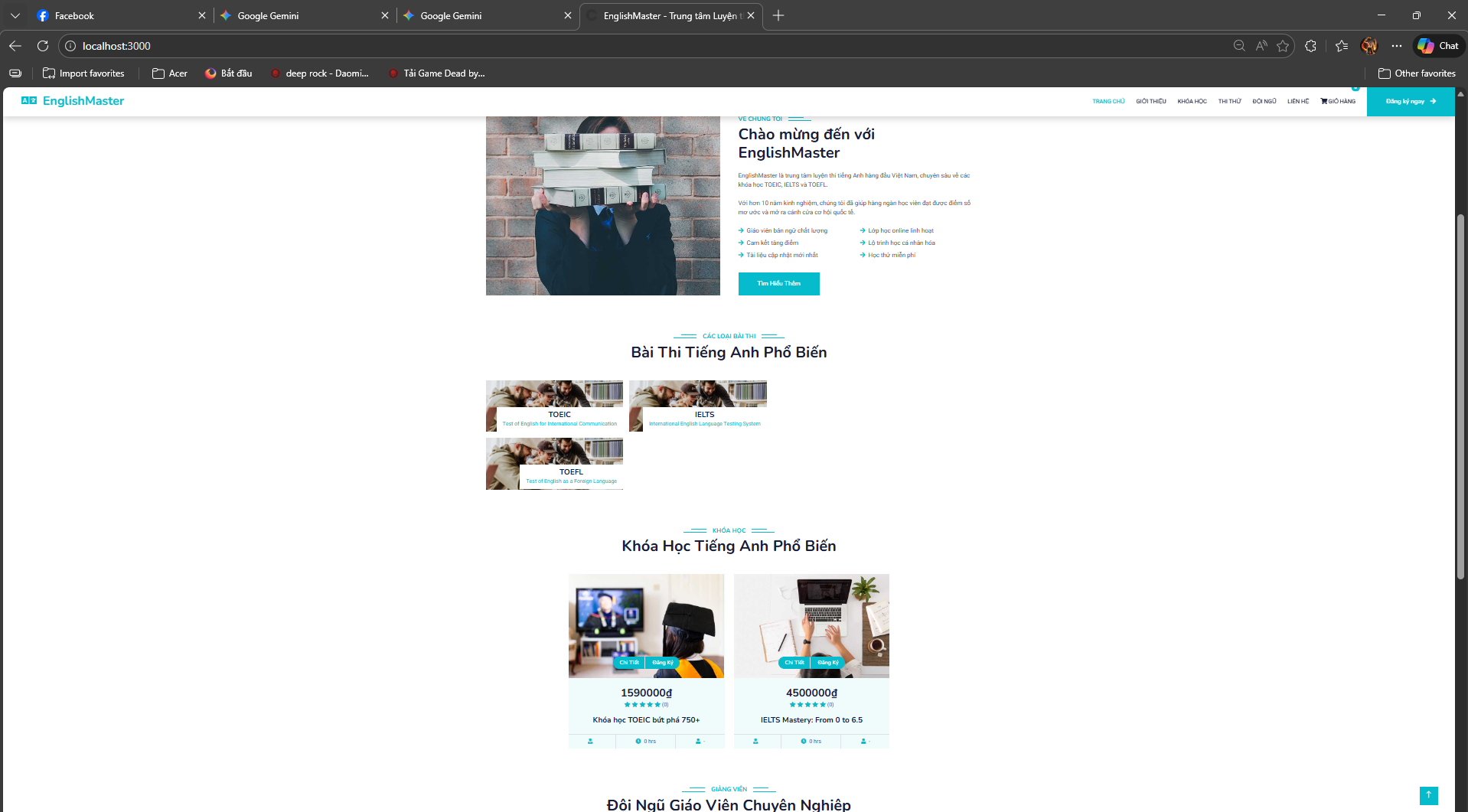
**\* Hỗ trợ VND, USD, EUR...**

**\* Tự động chuyển đổi tỷ giá**

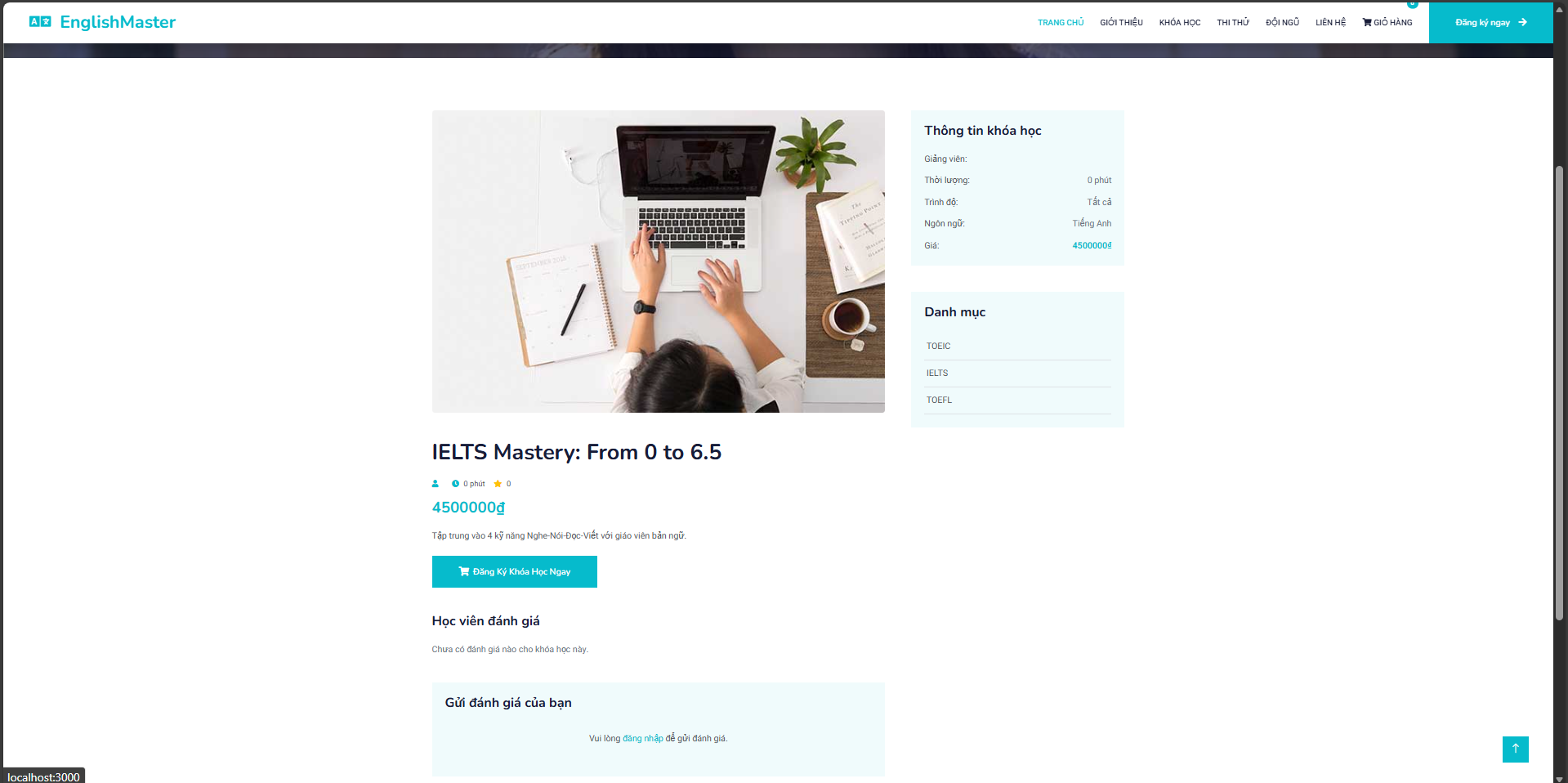
**\* Thanh toán bằng nhiều loại tiền tệ**

**6.GIAO DIỆN BỔ SUNG:**

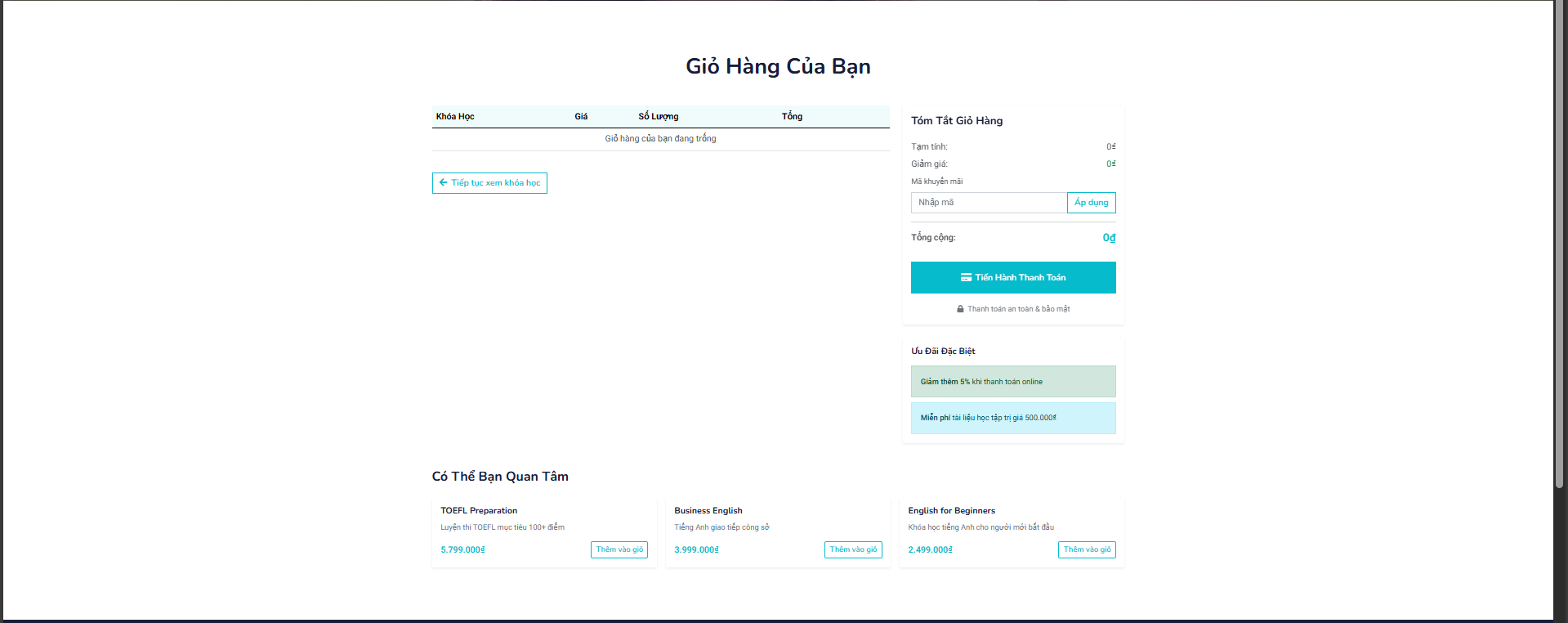
**6.1: Giao diện trang chủ:**

****

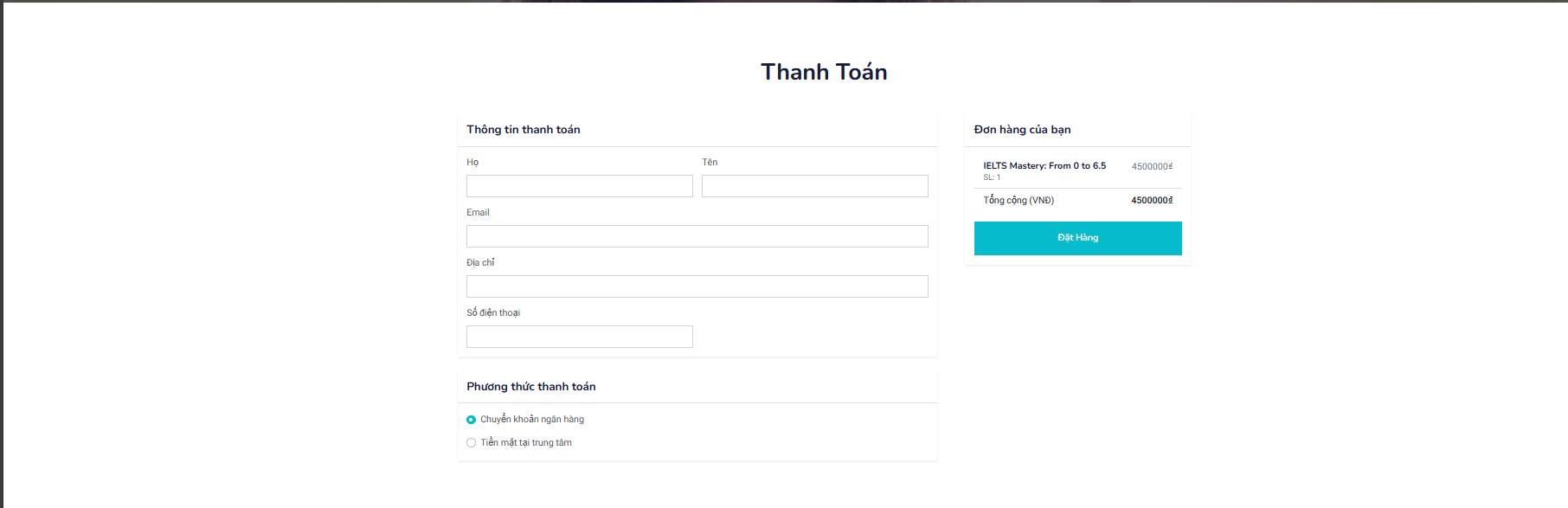
**6.2: Giao diện chi tiết khóa học:**

****

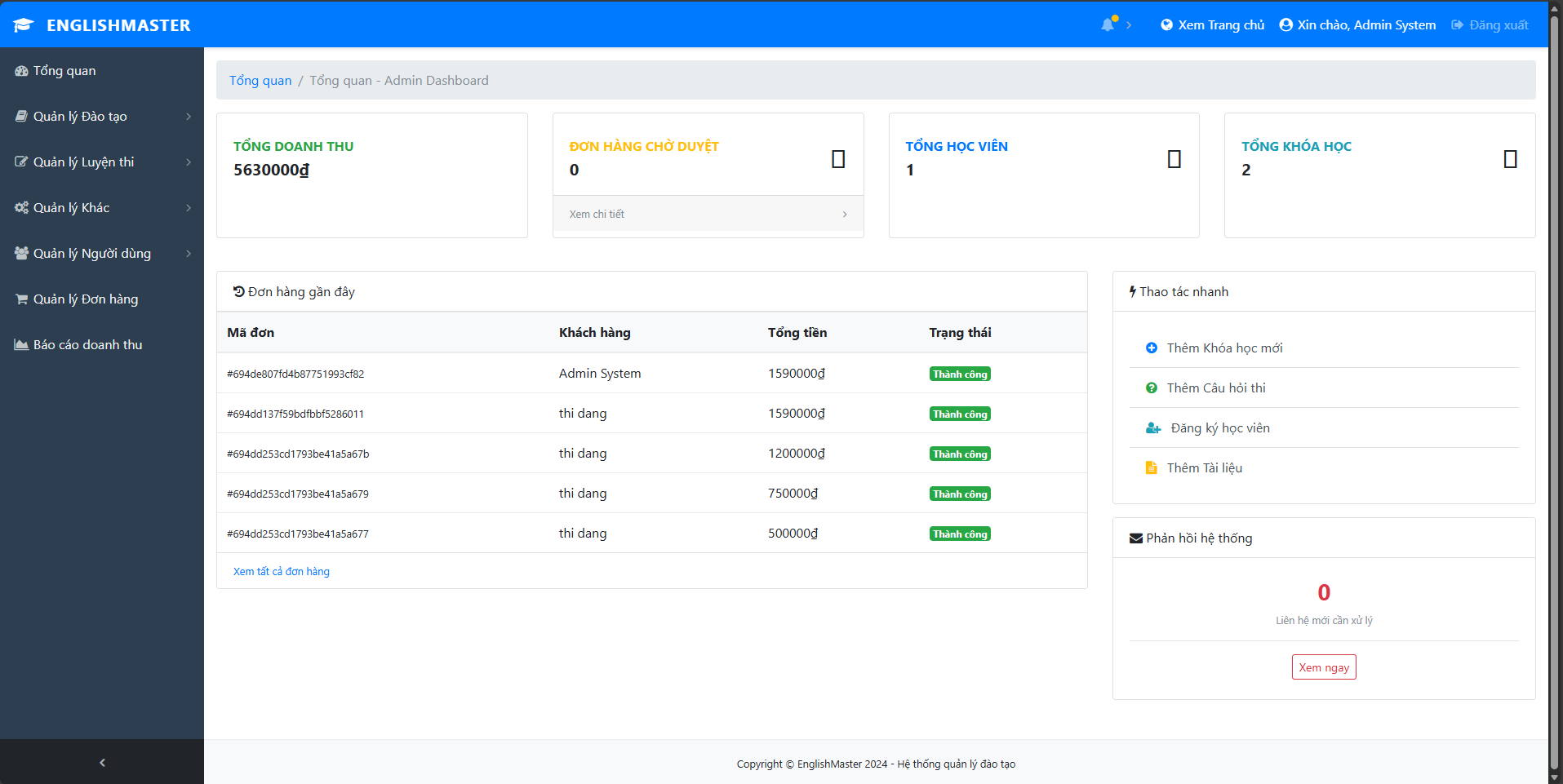
**6.3: Giao diện giỏ hàng:**

****

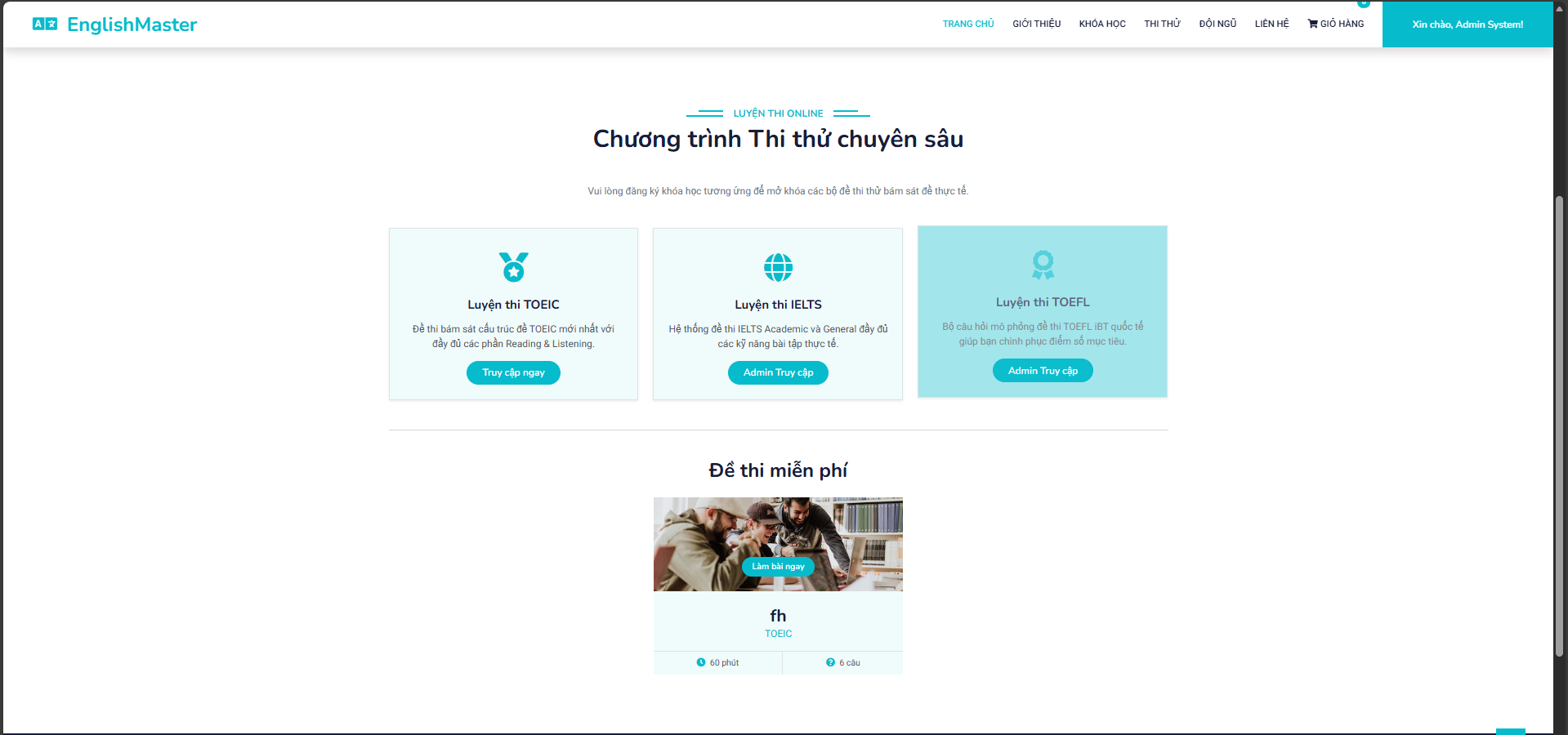
**6.4: Giao diện thanh toán:**

****

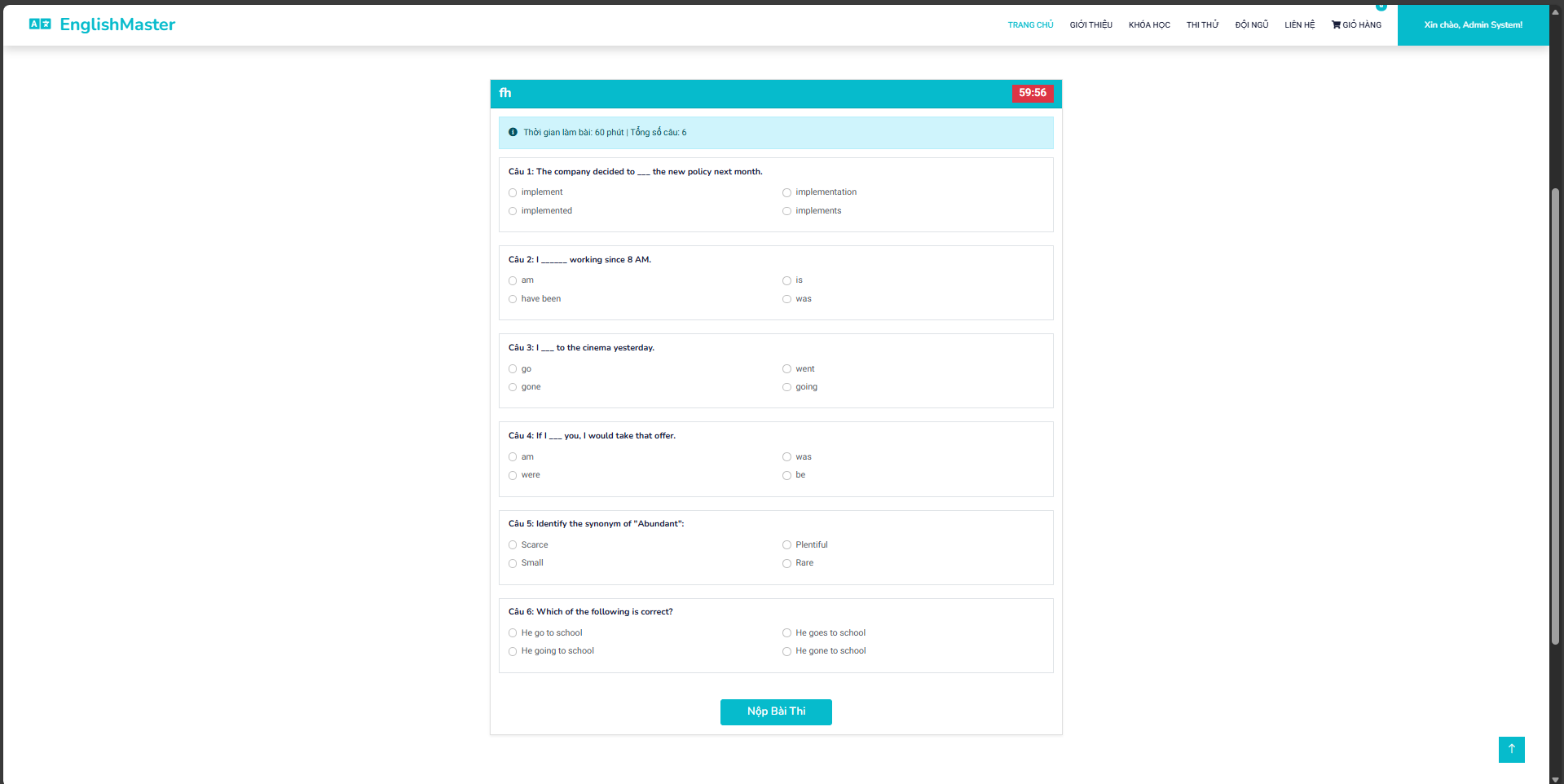
**6.5: Giao diện Dashboard Admin:**

****

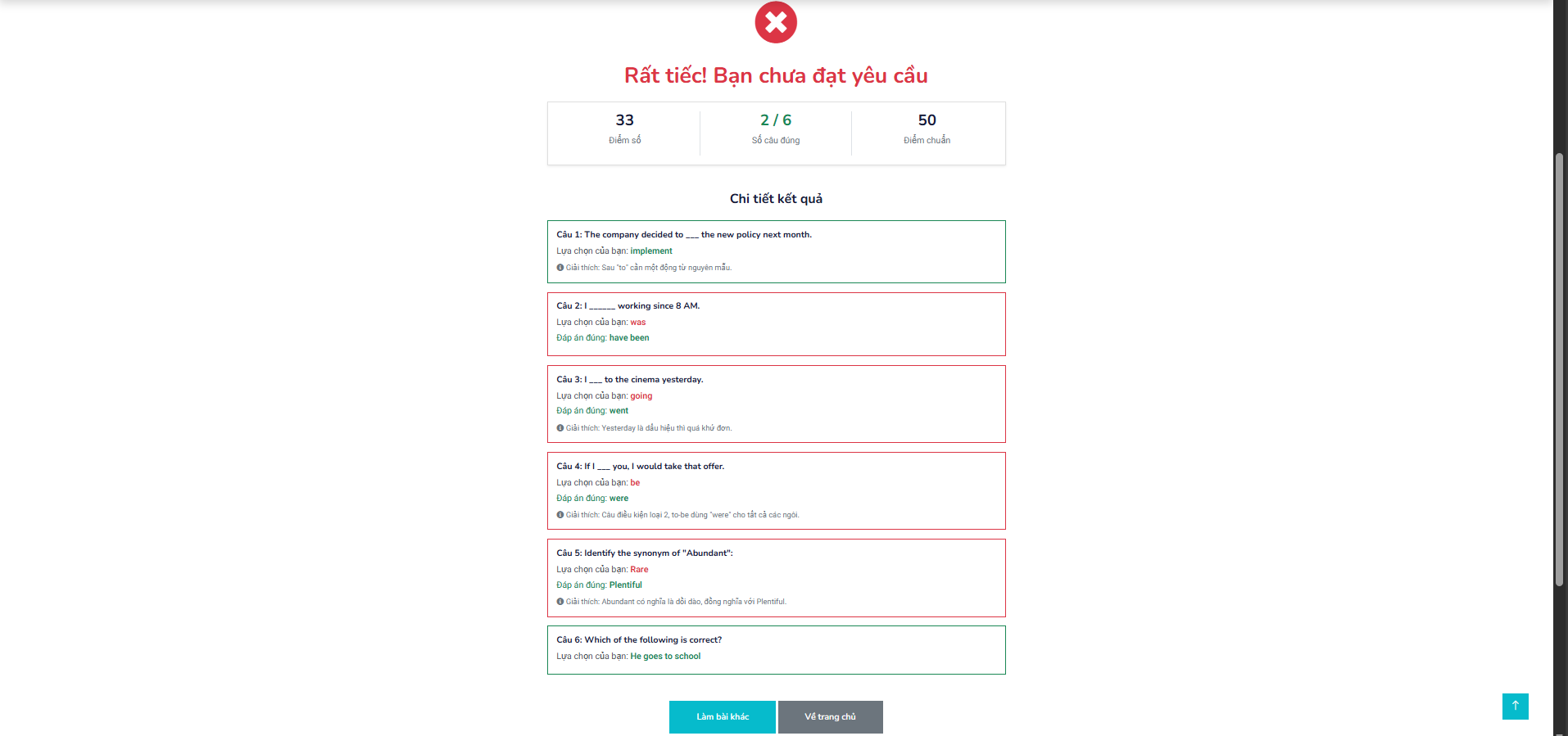
**6.6: Giao diện danh sách đề thi:**

****

**6.7: Giao diện làm bài thi:**

****

**6.8: Giao diện kết quả thi:**

****

**TÀI LIỆU THAM KHẢO**

**[1] Node.js Foundation. (n.d.). Node.js Documentation. Retrieved from https://nodejs.org/en/docs/**

**[2] Express. (n.d.). Express - Node.js web application framework. Retrieved from https://expressjs.com/**

**[3] MongoDB, Inc. (n.d.). MongoDB Documentation. Retrieved from https://docs.mongodb.com/**

**[4] Automattic. (n.d.). Mongoose Documentation. Retrieved from https://mongoosejs.com/docs/**

**[5] Passport. (n.d.). Passport.js documentation. Retrieved from http://www.passportjs.org/docs/**

**[6] NPM, Inc. (n.d.). express-handlebars - npm. Retrieved from https://www.npmjs.com/package/express-handlebars**

**[7] MDN Web Docs. (n.d.). Web technology for developers. Retrieved from https://developer.mozilla.org/en-US/**

**[8] W3Schools. (n.d.). HTML Tutorial. Retrieved from https://www.w3schools.com/html/**

**[9] W3Schools. (n.d.). CSS Tutorial. Retrieved from**

**https://www.w3schools.com/css/**

**[10] Stack Overflow. (n.d.). Questions tagged [node.js]. Retrieved from** [**https://stackoverflow.com/questions/tagged/node.js**](https://stackoverflow.com/questions/tagged/node.js)

**HẾT**