

**BỘ THÔNG TIN VÀ TRUYỀN THÔNG**  
**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**



**BÁO CÁO ĐỒ ÁN**  
**THỰC TẬP TỐT NGHIỆP**  
**ĐẠI HỌC**

***Đề tài:***

**“PHÁT TRIỂN ỨNG DỤNG THI TRẮC NGHIỆM TRÊN NỀN  
TẢNG LARAVEL”**

**Người hướng dẫn : ThS. NGUYỄN HOÀNG THÀNH**  
**Sinh viên thực hiện : LƯƠNG THÚY VY**  
**Mã số sinh viên : N20DCCN085**  
**Lớp : D20CQCNHT01-N**  
**Khoá : 2019 - 2024**  
**Ngành : HỆ THỐNG THÔNG TIN**  
**Hệ : ĐẠI HỌC CHÍNH QUY**

**TP.HCM, tháng 8/2024**

LƯƠNG THÚY VY

MSSV: N20DCCN085

Tên Đề tài: Phát triển ứng dụng thi trắc nghiệm trên nền tảng Laravel

Lớp: D20CQCNHT01-N

2019 - 2024

TP.  
HCM  
2024

**BỘ THÔNG TIN VÀ TRUYỀN THÔNG**  
**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

---



**BÁO CÁO ĐỒ ÁN**  
**THỰC TẬP TỐT NGHIỆP**  
**ĐẠI HỌC**

*Đề tài:*

**“PHÁT TRIỂN ỨNG DỤNG THI TRẮC NGHIỆM TRÊN NỀN  
TẢNG LARAVEL”**

**Người hướng dẫn : ThS. NGUYỄN HOÀNG THÀNH**  
**Sinh viên thực hiện : LƯƠNG THÚY VY**  
**Mã số sinh viên : N20DCCN085**  
**Lớp : D20CQCNHT01-N**  
**Khoá : 2019 - 2024**  
**Ngành : HỆ THỐNG THÔNG TIN**  
**Hệ : ĐẠI HỌC CHÍNH QUY**

**TP.HCM, tháng 8/2024**

**PHIẾU GIAO ĐỀ CƯƠNG THỰC TẬP TỐT NGHIỆP ĐƯỢC PHÊ DUYỆT**

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM  
Độc lập- Tự do- Hạnh phúc

TP. Hồ Chí Minh, ngày 5 tháng 8 năm 2024

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN  
THỰC TẬP TỐT NGHIỆP ĐẠI HỌC

- 1. Tên đề tài: Phát triển ứng dụng thi trắc nghiệm trên nền tảng Laravel
- 2. Sinh viên: Lương Thúy Vy Lớp: D20CQCNHT01-N
- 3. Giáo viên hướng dẫn: Ths. Nguyễn Hoàng Thành
- 4. Nơi công tác: khoa Công nghệ thông tin 2

NỘI DUNG NHẬN XÉT

- 1. Đánh giá chung:  
.....  
.....  
.....
- 2. Đánh giá chi tiết:  
.....  
.....  
.....  
.....  
.....  
.....
- 3. Nhận xét về tinh thần, thái độ làm việc:  
.....  
.....
- 4. Kết luận:  
.....  
.....
- 5. Điểm hướng dẫn:

GIẢNG VIÊN HƯỚNG DẪN  
(Ký, ghi rõ họ tên)

## LỜI CẢM ƠN

Lời đầu tiên, em xin gửi lời tri ân sâu sắc đến các thầy cô trường Học Viện Công Nghệ Bưu Chính Viễn thông cơ sở tại TP.HCM đặc biệt là quý thầy cô Khoa Công nghệ thông tin 2 đã tận tình dẫn dắt và truyền đạt cho em rất nhiều kiến thức quý báu trong các học kỳ vừa qua.

Đặc biệt, em xin gửi lời cảm ơn chân thành đến ThS. Nguyễn Hoàng Thành. Thầy đã hướng dẫn tận tình, truyền đạt kiến thức, chỉ bảo cho em trong suốt thời gian thực hiện đề tài. Kính chúc thầy và gia đình nhiều sức khỏe và thành công trong cuộc sống.

Tuy nhiên, vì thời gian và kiến thức chuyên môn còn hạn chế cùng nhiều yếu tố khách quan khác nên trong quá trình thực hiện đề tài không tránh khỏi những sai sót. Kính mong được lời nhận xét và góp ý của thầy cô, để em có thể hoàn thiện bản thân mình hơn. Một lần nữa, em xin chân thành cảm ơn!

TP.HCM, tháng 8 năm 2024

Sinh viên thực hiện

Lương Thúy Vy

# MỤC LỤC

<b>CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI.....</b>	<b>2</b>
1.1. Tổng quan đề tài .....	2
1.1.1. Lí do chọn đề tài .....	2
1.1.2. Mục đích, ý nghĩa chọn đề tài.....	2
1.2. Phương pháp nghiên cứu.....	3
1.2.1. Phạm vi, đối tượng của đề tài .....	3
1.2.2. Về mặt lý thuyết.....	3
1.2.3. Về mặt thực hành .....	3
1.3. Mục tiêu đề tài.....	3
1.4. Bố cục báo cáo .....	3
1.5. Công cụ sử dụng.....	3
<b>CHƯƠNG 2. CƠ SỞ LÝ THUYẾT.....</b>	<b>4</b>
2.1. Tổng quan về website thi trắc nghiệm .....	4
2.2. Giới thiệu về MySQL.....	4
2.3. Giới thiệu về Laravel Framework .....	5
2.4. Giới thiệu về ngôn ngữ lập trình PHP .....	6
2.5. Giới thiệu về JavaScript .....	6
2.6. Giới thiệu về React.....	6
2.7. Giới thiệu về RESTful API .....	8
2.8. Giới thiệu về công cụ lập trình Visual Studio Code.....	9
<b>CHƯƠNG 3. PHÂN TÍCH THIẾT KẾ HỆ THỐNG.....</b>	<b>11</b>
3.1. Khảo sát hệ thống thi trực tuyến.....	11
3.2. Các chức năng của hệ thống.....	11
3.2.1. Yêu cầu chức năng của hệ thống .....	11
3.2.2. Yêu cầu phi chức năng.....	12
3.3. Mô hình chức năng nghiệp vụ (BDF – Business Function Diagram) .....	12
3.3.1. Sơ đồ phân cấp chức năng mức đỉnh .....	13
3.4. Mô hình hóa yêu cầu .....	15
3.4.1. Các tác nhân của hệ thống .....	15
3.4.2. Xác định các usecase .....	16

3.4.3. Sơ đồ usecase .....	17
3.5. Mô hình thực thể (ERD – Entity Relationship Diagram) .....	26
3.5.1. Xác định thực thể .....	26
3.5.2. Mô hình ERD .....	27
3.5.3. Mô hình cơ sở dữ liệu quan hệ .....	27
3.6. Thiết kế dữ liệu .....	28
3.6.1. Mô hình Diagram .....	28
3.6.2. Từ điển dữ liệu .....	28
3.7. Thiết kế kiến trúc hệ thống .....	33
<b>CHƯƠNG 4. DEMO CHƯƠNG TRÌNH.....</b>	<b>35</b>
4.1. Giao diện trang chủ .....	35
4.2. Giao diện đăng ký .....	35
4.3. Giao diện đăng nhập .....	36
4.4. Giao diện dành cho thí sinh .....	36
4.4.1. Giao diện xem các ca thi đang có .....	37
4.4.2. Giao diện xem lịch thi .....	37
4.4.3. Giao diện xem phòng chờ .....	38
4.4.4. Giao diện thi .....	39
4.4.5. Giao diện xem kết quả thi .....	39
4.4.6. Giao diện xem lịch sử làm bài .....	40
4.4.7. Giao diện xem và chỉnh sửa thông tin cá nhân .....	40
4.5. Giao diện dành cho quản lý .....	42
4.5.1. Giao diện quản lý bộ câu hỏi .....	42
4.5.2. Giao diện quản lý đề thi .....	43
4.5.3. Giao diện quản lý ca thi .....	46
4.5.4. Giao diện quản lý tài khoản .....	48
<b>CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.....</b>	<b>50</b>
5.1. Kết luận .....	50
5.2. Kết quả đạt được .....	50
5.3. Hướng phát triển .....	50
<b>TÀI LIỆU THAM KHẢO.....</b>	<b>52</b>

## DANH MỤC CÁC KÝ HIỆU VÀ CHỮ VIẾT TẮT

API: Application Programming Interface	Giao thức lập trình ứng dụng
ERD: Entity Relationship Diagram	Mô hình thực thể
IDE: Integrated Development Environment	Môi trường phát triển tích hợp
SQL: Structured Query Language	Ngôn ngữ truy vấn dữ liệu
UI: User Interface	Giao diện người dùng



## DANH MỤC CÁC BẢNG, SƠ ĐỒ, HÌNH

Bảng 3.1 Bảng xác định các tác nhân của hệ thống .....	15
Bảng 3.2 Xác định các usecase.....	16
Bảng 3.3 Bảng thực thể accounts .....	29
Bảng 3.4 Bảng thực thể examinees .....	29
Bảng 3.5 Bảng thực thể roles.....	29
Bảng 3.6 Bảng thực thể grades.....	29
Bảng 3.7 Bảng thực thể subjects .....	30
Bảng 3.8 Bảng thực thể subject_grades.....	30
Bảng 3.9 Bảng thực thể question_answers .....	31
Bảng 3.10 Bảng thực thể exams .....	31
Bảng 3.11 Bảng thực thể exam_questions.....	31
Bảng 3.12 Bảng thực thể exam_shifts .....	32
Bảng 3.13 Bảng thực thể tests .....	32
Bảng 3.14 Bảng thực thể test_answers .....	33
Sơ đồ 3.1 Sơ đồ phân cấp chức năng mức đỉnh .....	13
Sơ đồ 3.2 Sơ đồ phân cấp chức năng thi trắc nghiệm .....	13
Sơ đồ 3.3 Sơ đồ phân cấp chức năng quản lý.....	14
Sơ đồ 3.4 Sơ đồ phân rã chức năng hoàn chỉnh .....	15
Sơ đồ 3.5 Usecase tổng quát.....	17
Sơ đồ 3.6 Usecase các chức năng của quản lý .....	18
Sơ đồ 3.7 Usecase các chức năng của thí sinh .....	18
Sơ đồ 3.8 Sơ đồ usecase đăng nhập.....	19
Sơ đồ 3.9 Sơ đồ usecase đăng xuất.....	19
Sơ đồ 3.10 Sơ đồ usecase thay đổi thông tin cá nhân.....	20
Sơ đồ 3.11 Sơ đồ usecase đăng ký thi .....	20
Sơ đồ 3.12 Sơ đồ usecase xem lịch thi .....	21
Sơ đồ 3.13 Sơ đồ usecase làm bài thi trắc nghiệm .....	21
Sơ đồ 3.14 Sơ đồ usecase xem lịch sử thi .....	22
Sơ đồ 3.15 Sơ đồ usecase quản lý câu hỏi.....	23
Sơ đồ 3.16 Sơ đồ usecase quản lý đề thi .....	23

Sơ đồ 3.17 Sơ đồ usecase thêm câu hỏi vào đề thi .....	24
Sơ đồ 3.18 Sơ đồ usecase quản lý ca thi.....	25
Sơ đồ 3.19 Sơ đồ usecase quản lý thí sinh.....	25
Hình 2.1 Giới thiệu về MySQL .....	4
Hình 2.2 Giới thiệu về Laravel Framework.....	5
Hình 2.3 Giới thiệu về PHP .....	6
Hình 2.4 Giới thiệu React.....	7
Hình 2.5 Giới thiệu về Restful API .....	8
Hình 3.1 Mô hình ERD .....	27
Hình 3.2 Mô hình Diagram .....	28
Hình 3.3 Mô hình kiến trúc của hệ thống .....	33
Hình 4.1 Giao diện trang chủ.....	35
Hình 4.2 Giao diện đăng ký (Trang 1).....	35
Hình 4.3 Giao diện đăng ký (Trang 2).....	36
Hình 4.4 Giao diện đăng nhập .....	36
Hình 4.5 Giao diện trang chủ của thí sinh .....	37
Hình 4.6 Giao diện xem các ca thi đang có .....	37
Hình 4.7 Giao diện xem lịch thi .....	38
Hình 4.8 Giao diện xem phòng chờ.....	38
Hình 4.9 Giao diện thi .....	39
Hình 4.10 Giao diện xem kết quả thi.....	40
Hình 4.11 Giao diện xem lịch sử làm bài .....	40
Hình 4.12 Giao diện xem thông tin cá nhân .....	41
Hình 4.13 Giao diện cập nhật thông tin cá nhân .....	41
Hình 4.14 Giao diện quản lý bộ câu hỏi .....	42
Hình 4.15 Giao diện thêm câu hỏi.....	42
Hình 4.16 Giao diện chỉnh sửa câu hỏi .....	43
Hình 4.17 Giao diện xóa câu hỏi .....	43
Hình 4.18 Giao diện xem đề thi.....	44
Hình 4.19 Giao diện thêm đề thi .....	44
Hình 4.20 Giao diện chỉnh sửa đề thi .....	45
Hình 4.21 Giao diện xóa đề thi.....	45
Hình 4.22 Giao diện quản lý chi tiết đề thi.....	46

Hình 4.23 Giao diện quản lý các ca thi.....	46
Hình 4.24 Giao diện thêm ca thi.....	47
Hình 4.25 Giao diện xóa ca thi.....	47
Hình 4.26 Giao diện chỉnh sửa ca thi .....	48
Hình 4.27 Giao diện quản lý người dùng .....	48
Hình 4.28 Giao diện xem chi tiết thông tin người dùng.....	49

## LỜI MỞ ĐẦU

Trong thời đại mà công nghệ số đang ngày càng chiếm ưu thế và thâm nhập sâu vào mọi khía cạnh của cuộc sống, giáo dục không phải là ngoại lệ. Công nghệ đã thay đổi cách chúng ta học tập và giảng dạy, mở ra những cơ hội mới và tối ưu hóa quy trình giáo dục. Trong bối cảnh này, việc phát triển một ứng dụng thi trực tuyến không chỉ đơn thuần là một dự án công nghệ, mà là một bước tiến quan trọng trong việc cải thiện chất lượng giáo dục và nâng cao hiệu quả của quản lý giáo dục.

Đề án này đặt mục tiêu nghiên cứu và xây dựng một hệ thống thi trực tuyến linh hoạt và tiện lợi, hỗ trợ các tổ chức giáo dục tổ chức các kỳ thi một cách hiệu quả và chuyên nghiệp hơn bao giờ hết, không chỉ đáp ứng nhu cầu thực tiễn của các tổ chức giáo dục hiện nay mà còn góp phần nâng cao chất lượng giáo dục toàn diện, phù hợp với xu hướng phát triển của xã hội hiện đại.

Ngoài ra, đề án cũng nhấn mạnh vào lợi ích to lớn mà việc áp dụng công nghệ thi trực tuyến mang lại. Từ khả năng cung cấp môi trường thi công bằng, minh bạch và công bằng, đến việc tối ưu hóa quy trình tổ chức và đánh giá kết quả thi, ứng dụng này hứa hẹn đem lại sự tiện lợi và hiệu quả cho cả giảng viên và học sinh. Nó không chỉ giúp tăng cường tính năng động và tương tác trong quá trình học tập mà còn thúc đẩy sự phát triển năng lực toàn diện của người học, chuẩn bị cho họ với một môi trường học tập phù hợp với thế kỷ 21.

Với những lợi ích và tiềm năng vượt trội như vậy, nên em đã quyết định chọn đề tài “**Phát triển ứng dụng thi trắc nghiệm trên nền tảng Laravel**”, sử dụng nền tảng Laravel, với tính bảo mật cao và khả năng mở rộng linh hoạt, sẽ giúp cho ứng dụng có thể phát triển và mở rộng theo nhu cầu thực tế của từng tổ chức giáo dục cụ thể.

## CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI

### 1.1. Tổng quan đề tài

#### 1.1.1. Lí do chọn đề tài

Như chúng ta cũng thấy, việc tổ chức các kỳ thi trong môi trường giáo dục hiện nay đang gặp nhiều thách thức vì quy mô lớn và yêu cầu về tính chính xác cao. Bên cạnh đó, các phương pháp tổ chức thi truyền thống thường mất nhiều thời gian, nhân lực và dễ phát sinh sai sót. So với các phương pháp truyền thống, việc ứng dụng công nghệ thông tin vào quản lý thi cử giúp tiết kiệm chi phí, nâng cao hiệu quả và đảm bảo tính minh bạch.

Với sự phát triển của Internet và các công nghệ số, việc triển khai hệ thống quản lý thi cử trực tuyến không chỉ giúp tiết kiệm chi phí mà còn nâng cao hiệu quả quản lý. Hệ thống này có khả năng quản lý thông tin thí sinh, đề thi, ca thi và kết quả thi một cách tự động, giảm thiểu sai sót và tăng cường tính minh bạch.

Nắm bắt được những nhu cầu đó, em đã quyết định chọn đề tài **“Phát triển ứng dụng thi trắc nghiệm trực tuyến trên nền tảng Laravel”** nhằm giúp các tổ chức giáo dục chuyển đổi từ phương pháp quản lý thi cử truyền thống sang phương pháp quản lý hiện đại, ứng dụng công nghệ thông tin để cải thiện hiệu quả quản lý và nâng cao chất lượng thi cử. Hệ thống này sẽ cung cấp các ca thi trực tuyến và cho phép thí sinh lựa chọn giờ thi phù hợp với mình. Thí sinh có thể thi thử, từ đó làm quen với cấu trúc đề thi và nâng cao kỹ năng làm bài trước khi bước vào kỳ thi chính thức.

#### 1.1.2. Mục đích, ý nghĩa chọn đề tài

Trong bối cảnh giáo dục hiện nay, việc tổ chức thi cử truyền thống đang đối mặt với nhiều thách thức như tốn kém thời gian, nhân lực, chi phí và nguy cơ sai sót cao trong quá trình quản lý.

Hệ thống quản lý thi cử trực tuyến sẽ tối ưu hóa toàn bộ quy trình, từ việc đăng ký thí sinh, quản lý đề thi, ca thi cho đến chấm thi và thông báo kết quả, giảm thiểu tối đa sai sót và tăng cường tính minh bạch. Đặc biệt, hệ thống sẽ cung cấp các ca thi trực tuyến linh hoạt, cho phép thí sinh tự do lựa chọn giờ thi phù hợp với lịch trình cá nhân. Thêm vào đó, thí sinh có thể tham gia các bài thi thử để làm quen với cấu trúc đề thi và nâng cao kỹ năng làm bài, chuẩn bị tốt nhất cho kỳ thi chính thức.

Nhận thấy nhu cầu cấp thiết này, em quyết định chọn đề tài **“Phát triển ứng dụng thi trắc nghiệm trực tuyến trên nền tảng Laravel”** nhằm mang đến một giải pháp hiệu quả và hiện đại hơn. Đề tài không chỉ nhằm nâng cao hiệu quả quản lý thi cử mà còn đóng góp một công cụ hữu ích cho các tổ chức giáo dục, thúc đẩy việc áp dụng công nghệ thông tin vào giáo dục và đáp ứng nhu cầu học tập, thi cử ngày càng cao của xã hội. Hệ thống thi trắc nghiệm trực tuyến này được xây dựng trên nền tảng Laravel, với hy vọng mang đến một trải nghiệm thi cử mới mẻ, hiệu quả và tiết kiệm chi phí cho cả tổ chức giáo dục lẫn thí sinh.

Với những kiến thức còn hạn chế, em chắc chắn không thể tránh khỏi những thiếu sót trong quá trình thực hiện đề tài. Vì vậy, em rất mong nhận được sự góp ý và hướng dẫn của quý thầy cô để hoàn thiện đề tài tốt hơn. Em xin chân thành cảm ơn!

## **1.2. Phương pháp nghiên cứu**

### **1.2.1. Phạm vi, đối tượng của đề tài**

- Phạm vi: thực hiện những chức năng chính của một trang thi trực tuyến.
- Đối tượng:
  - + Quản lý: là người quản trị trang, truy cập vào hệ thống nhằm mục đích quản trị trang thi trực tuyến.
  - + Thí sinh: là những người tham gia thực hiện các bài thi trên hệ thống.

### **1.2.2. Về mặt lý thuyết**

- Hiểu được cách lưu và truy xuất dữ liệu của hệ quản trị cơ sở dữ liệu MySQL.
- Hiểu được RestfulAPI và cách thức hoạt động của nó.
- Tìm hiểu các kỹ thuật lập trình, cách thức hoạt động của Laravel.
- Tìm hiểu các kỹ thuật lập trình, cách thức hoạt động của React.

### **1.2.3. Về mặt thực hành**

Phân tích và thiết kế cơ sở lưu giữ dữ liệu.

Xây dựng website tổ chức thi trực tuyến với các chức năng:

- Đăng ký, đăng nhập, đăng xuất, tạo tài khoản, phân quyền
- Quản trị đề thi: thêm/ xóa/ sửa đề thi
- Quản lý câu hỏi trong đề thi: thêm/ xóa/ sửa các câu hỏi và đáp án trong đề thi
- Quản trị ca thi: thêm/ xóa/ sửa ca thi, gán đề thi cho ca thi
- Quản trị thí sinh dự thi: quản lý thông tin thí sinh, xem danh sách thí sinh dự thi, xem kết quả của các thí sinh sau khi hoàn thành bài thi
- Thí sinh xem thông tin của ca thi, tham gia thi, nộp bài thi và xem kết quả thi

## **1.3. Mục tiêu đề tài**

Đảm bảo việc thi trực tuyến của thí sinh diễn ra nhanh chóng, chính xác:

- Giúp người quản lý quản lý tình hình hoạt động của hệ thống cũng như các thí sinh.
- Đảm bảo thí sinh thực hiện việc thi trực tuyến thành công và quá trình đăng ký cũng thi bắt đầu thi diễn ra đơn giản.

## **1.4. Bố cục báo cáo**

- Chương 1: Giới thiệu đề tài
- Chương 2: Cơ sở lý thuyết
- Chương 3: Phân tích thiết kế hệ thống
- Chương 4: Demo chương trình
- Chương 5: Kết luận, hướng phát triển

## **1.5. Công cụ sử dụng**

- Visual Studio Code
- MySQL Server Workbench

## CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

### 2.1. Tổng quan về website thi trắc nghiệm

Ngày nay hình thức thi trắc nghiệm đang trở thành một xu hướng tất yếu cho rất nhiều kì thi, đặc biệt là trong những năm gần đây Bộ Giáo Dục nước ta đã đưa hình thức thi trắc nghiệm vào trong kỳ thi trung học phổ thông quốc gia với hầu hết các môn. Cùng với đó khi ngành công nghệ thông tin ngày càng phát triển, mạng lưới internet ngày càng lớn mạnh đa số học sinh, sinh viên đều có cơ hội sử dụng, tìm hiểu về các website.

Để góp phần phát triển thêm cho hình thức thi trắc nghiệm tôi đã nghiên cứu xây dựng website thi thử trắc nghiệm nhằm hỗ trợ cho người thi thử có thể làm bài thi, xem kết quả của mình và cũng hỗ trợ cả những người tạo bộ đề thi, quản lý các câu hỏi và bộ đề thi dễ dàng hơn.

### 2.2. Giới thiệu về MySQL

MySQL là một hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) mã nguồn mở và miễn phí, được sử dụng rộng rãi trên toàn thế giới. Nó được phát triển bởi Oracle Corporation và được sử dụng bởi các doanh nghiệp, tổ chức và cá nhân thuộc mọi quy mô.

MySQL là một hệ quản trị cơ sở dữ liệu rất linh hoạt và mạnh mẽ, có thể được sử dụng để lưu trữ và quản lý một lượng lớn dữ liệu. Nó cũng rất dễ sử dụng và có thể được học và sử dụng bởi người dùng ở mọi cấp độ kinh nghiệm.

MySQL là một hệ quản trị cơ sở dữ liệu quan hệ, có nghĩa là dữ liệu được lưu trữ trong các bảng. Mỗi bảng có một số cột và mỗi cột có một kiểu dữ liệu cụ thể. Ví dụ, một bảng có thể có các cột cho tên, địa chỉ và số điện thoại.

MySQL cũng có một số tính năng mạnh mẽ, chẳng hạn như khả năng tạo khóa chính, khóa ngoại và chỉ mục. Khóa chính là một cột hoặc một nhóm cột xác định duy nhất mỗi hàng trong một bảng. Khóa ngoại là một cột hoặc một nhóm cột tham chiếu khóa chính của một bảng khác. Chỉ mục là một cấu trúc dữ liệu giúp cải thiện hiệu suất truy vấn.

MySQL là một hệ quản trị cơ sở dữ liệu rất phổ biến và được sử dụng bởi nhiều ứng dụng khác nhau, chẳng hạn như các trang web, ứng dụng di động và các ứng dụng doanh nghiệp. Nó cũng được sử dụng bởi nhiều nhà cung cấp dịch vụ đám mây, chẳng hạn như Amazon Web Services, Microsoft Azure và Google Cloud Platform [1].



Hình 2.1 Giới thiệu về MySQL

### 2.3. Giới thiệu về Laravel Framework

Laravel là một framework mã nguồn mở phát triển bằng ngôn ngữ PHP, được thiết kế để xây dựng các ứng dụng web mạnh mẽ và linh hoạt. Laravel tuân thủ mô hình MVC (Model-View-Controller) và cung cấp nhiều tính năng và công cụ hữu ích như routing, giao diện dễ sử dụng, ORM (Object-Relational Mapping), tạo và quản lý cơ sở dữ liệu, bảo mật, caching, và nhiều hơn nữa. Laravel cung cấp cú pháp đơn giản, dễ hiểu và đảm bảo viết mã mạnh mẽ, hiệu quả và dễ bảo trì [2].



Hình 2.2 Giới thiệu về Laravel Framework

Laravel ứng dụng một mô hình kiến trúc phát triển mạnh mẽ gọi là Model-View-Controller (MVC). Mô hình này chia ứng dụng thành ba phần quan trọng, mỗi phần có nhiệm vụ riêng biệt, tạo nên một quy trình phát triển ứng dụng mượt mà và có tổ chức:

**Controller:** Đó là trái tim của ứng dụng. Controller nhận yêu cầu từ người dùng, sau đó tương tác với model nếu cần thao tác với cơ sở dữ liệu, và sau đó gọi đến view để hiển thị dữ liệu cho người dùng. Điều này giúp tách biệt logic xử lý và giao diện người dùng, giúp quản lý mã nguồn dễ dàng hơn và tạo điểm khả năng mở rộng.

**Model:** Model là nơi xác định cấu trúc dữ liệu và quan hệ giữa các thành phần trong dự án. Nó cho phép ta thực hiện các truy vấn vào cơ sở dữ liệu và thực hiện các thao tác CRUD (Create, Read, Update, Delete) một cách dễ dàng. Điều này giúp quản lý dữ liệu một cách hiệu quả và bảo trì tính nhất quán.

**View:** Đây là thành phần cuối cùng trong chu trình. View đảm nhiệm nhiệm vụ hiển thị dữ liệu cho người dùng. Giúp ta có thể tạo các template view đẹp và dễ đọc, và từ đó, hiển thị kết quả cuối cùng của quá trình xử lý. Điều này cho phép ta kiểm soát cách dữ liệu được hiển thị một cách linh hoạt, mà không cần can thiệp vào logic xử lý hoặc cơ sở dữ liệu.

MVC là một kiến trúc mạnh mẽ và có tổ chức, giúp tách biệt các phần quan trọng của ứng dụng và quản lý chúng một cách hiệu quả. Laravel đã tích hợp và tối ưu hóa sự sử dụng của kiến trúc này, hỗ trợ phát triển ứng dụng một cách nhanh chóng và dễ dàng.

Những tính năng tuyệt vời có thể kể đến của Laravel là:

- Composer: sử dụng để nâng cấp, cài đặt...
- Eloquent ORM: thao tác với cú pháp đẹp mắt và đơn giản.
- Restful API: hỗ trợ biến Laravel thành một web service API.
- Artisan: cung cấp các lệnh cần thiết để phát triển ứng dụng.



- View: giúp code sạch sẽ hơn rất nhiều.
- Migrations: hỗ trợ tạo các trường trong cơ sở dữ liệu, thêm các cột trong bảng, tạo mối quan hệ giữa các bảng, hỗ trợ quản lý cơ sở dữ liệu.
- Authentication: cung cấp sẵn các tính năng đăng nhập, đăng ký, quên mật khẩu...
- Unit Testing: hỗ trợ test lỗi để sửa chữa.

## 2.4. Giới thiệu về ngôn ngữ lập trình PHP

PHP (Hypertext Preprocessor) là một ngôn ngữ lập trình mã nguồn mở được sử dụng rộng rãi cho phát triển ứng dụng web. PHP là ngôn ngữ kịch bản được thực thi trên máy chủ, giúp tạo nội dung động và tương tác với cơ sở dữ liệu. PHP hỗ trợ nhiều hệ quản trị cơ sở dữ liệu phổ biến như MySQL, PostgreSQL và SQLite. Ngôn ngữ này cũng cung cấp các thư viện và framework mạnh mẽ như Laravel, Symfony và CodeIgniter để phát triển ứng dụng web hiệu quả [3].



Hình 2.3 Giới thiệu về PHP

## 2.5. Giới thiệu về JavaScript

JavaScript là ngôn ngữ lập trình được nhà phát triển sử dụng để tạo trang web tương tác. Từ làm mới bảng tin trên trang mạng xã hội đến hiển thị hình ảnh động và bản đồ tương tác, các chức năng của JavaScript có thể cải thiện trải nghiệm người dùng của trang web. Là ngôn ngữ kịch bản phía máy khách, JavaScript là một trong những công nghệ cốt lõi của World Wide Web.

### Ưu điểm của JavaScript:

- Dễ dàng học và sử dụng.
- Có thể không phụ thuộc vào nền tảng.
- Giảm tải máy chủ.
- Cải thiện giao diện người dùng.
- Hỗ trợ đồng thời việc chạy song song nhiều tập hợp hướng dẫn khác nhau.

## 2.6. Giới thiệu về React

ReactJS là một opensource được phát triển bởi Facebook, ra mắt vào năm 2013, bản thân nó là một thư viện Javascript được dùng để xây dựng các tương tác với các thành phần trên website. Một trong những điểm nổi bật nhất của ReactJS đó là việc render dữ liệu không chỉ thực hiện được trên tầng server mà còn ở dưới client nữa.



*Hình 2.4 Giới thiệu React*

### **Ưu điểm của React:**

- Phù hợp với đa dạng thể loại website: ReactJS khiến cho việc khởi tạo website dễ dàng hơn bởi vì không cần phải code nhiều như khi tạo trang web thuần chỉ dùng JavaScript, HTML và nó đã cung cấp đủ các công cụ có thể dùng cho nhiều trường hợp.
- Tái sử dụng các Component: Nếu xây dựng các Component đủ tốt, đủ flexible để có thể thoả các “yêu cầu” của nhiều dự án khác nhau, chúng ta chỉ tốn thời gian xây dựng ban đầu và sử dụng lại hầu như toàn bộ ở các dự án sau. Không chỉ riêng mỗi ReactJS mà các framework hiện nay cũng đều cho phép chúng ta thực hiện điều đó.
- Thân thiện với SEO: SEO là một phần không thể thiếu để đưa thông tin website của bạn lên top đầu tìm kiếm của Google. Bản chất ReactJS là một thư viện JavaScript, Google Search Engine hiện nay đã crawl và index được code JavaScript.
- Debug dễ dàng: Facebook đã phát hành 1 Chrome extension dùng trong việc debug trong quá trình phát triển ứng dụng. Điều đó giúp tăng tốc quá trình release sản phẩm cũng như quá trình coding [4].

Nguyên lý xây dựng của React dựa trên components (component-based approach), có thể tái sử dụng và phù hợp với ứng dụng 1 trang (Single Page Application – SPA). React giúp lập trình viên xây dựng giao diện người dùng dựa trên JSX (một cú pháp mở rộng của JavaScript), tạo ra các DOM ảo (virtual DOM) để tối ưu việc render 1 trang web.

ReactJs sau khi ra đời đã cho thấy sự phù hợp của nó trong việc phát triển các ứng dụng Web với nhiều chức năng được tích hợp. Nó đã tạo thành 1 xu thế, 1 hình mẫu phát triển website với nhiều chức năng, khả năng tương tác đa dạng với người dùng. Hiện tại sau hơn 10 năm phát triển thì React vẫn đang chiếm vị trí số 1 trong các thư viện Front-end hiện tại.

### **Cách hoạt động của React:**

Component-Based Architecture: React sử dụng kiến trúc dựa trên component, trong đó giao diện người dùng được chia thành các thành phần nhỏ, độc lập và có thể tái sử dụng. Mỗi component trong React là một lớp hoặc một hàm JavaScript, chứa logic và giao diện riêng biệt. Các component có thể lồng vào nhau để tạo ra cấu trúc UI phức tạp.

Single-way data flow (Luồng dữ liệu một chiều): ReactJS không có những module chuyên dụng để xử lý data, vì vậy ReactJS chia nhỏ view thành các component nhỏ có mối quan hệ chặt chẽ với nhau.

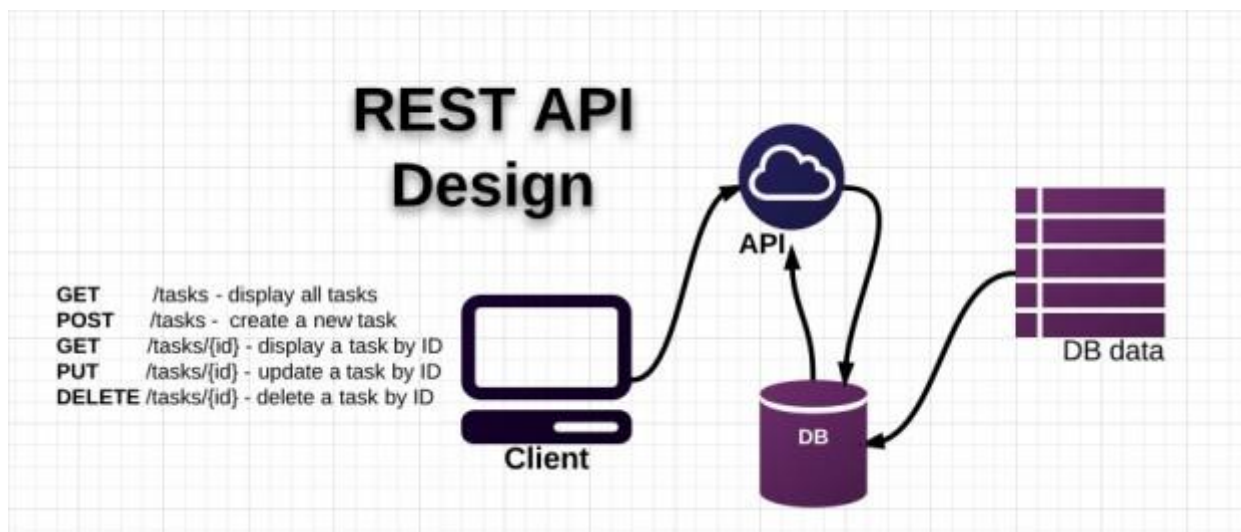
Virtual DOM: Những Framework sử dụng Virtual-DOM như ReactJS khi Virtual-DOM thay đổi, chúng ta không cần thao tác trực tiếp với DOM trên View mà vẫn phản ánh được sự thay đổi đó. Do Virtual-DOM vừa đóng vai trò là Model, vừa đóng vai trò là View nên mọi sự thay đổi trên Model đã kéo theo sự thay đổi trên View và ngược lại. Điều này làm cho tốc độ ứng dụng tăng lên đáng kể – một lợi thế không thể tuyệt vời hơn khi sử dụng Virtual-DOM.

Lifecycle Methods: Các component trong React có các phương thức vòng đời (lifecycle methods) cho phép lập trình viên can thiệp vào các giai đoạn khác nhau của vòng đời component.

Handling Events: React xử lý các sự kiện bằng cách sử dụng cú pháp camelCase và truyền hàm sự kiện trực tiếp trong JSX. Điều này tương tự như cách làm việc với các sự kiện trong DOM, nhưng với React, chúng ta không cần phải sử dụng addEventListener.

## 2.7. Giới thiệu về RESTful API

RESTful API là một tiêu chuẩn dùng trong việc thiết kế API cho các ứng dụng web (thiết kế Web services) để tiện cho việc quản lý các resource. Nó chú trọng vào tài nguyên hệ thống (tệp văn bản, ảnh, âm thanh, video, hoặc dữ liệu động...), bao gồm các trạng thái tài nguyên được định dạng và được truyền tải qua HTTP.



Hình 2.5 Giới thiệu về Restful API

Diễn giải các thành phần:

API (Application Programming Interface) là một tập các quy tắc và cơ chế mà theo đó, một ứng dụng hay một thành phần sẽ tương tác với một ứng dụng hay thành phần khác. API có thể trả về dữ liệu mà bạn cần cho ứng dụng của mình ở những kiểu dữ liệu phổ biến như JSON hay XML.

REST (Representational State Transfer) là một dạng chuyển đổi cấu trúc dữ liệu, một kiểu kiến trúc để viết API. Nó sử dụng phương thức HTTP đơn giản để tạo cho giao tiếp giữa các máy.

Vì vậy, thay vì sử dụng một URL cho việc xử lý một số thông tin người dùng, REST gửi một yêu cầu HTTP như GET, POST, DELETE, vv đến một URL để xử lý dữ liệu.

RESTful API là một tiêu chuẩn dùng trong việc thiết kế các API cho các ứng dụng web để quản lý các resource. RESTful là một trong những kiểu thiết kế API được sử dụng phổ biến ngày nay để cho các ứng dụng (web, mobile...) khác nhau giao tiếp với nhau.

Chức năng quan trọng nhất của REST là quy định cách sử dụng các HTTP method (như GET, POST, PUT, DELETE...) và cách định dạng các URL cho ứng dụng web để quản các resource. RESTful không quy định logic code ứng dụng và không giới hạn bởi ngôn ngữ lập trình ứng dụng, bất kỳ ngôn ngữ hoặc framework nào cũng có thể sử dụng để thiết kế một RESTful API. JavaScript là một ngôn ngữ lập trình của HTML và WEB. Nó là nhẹ và được sử dụng phổ biến nhất như là một phần của các trang web, mà sự thi hành của chúng cho phép Client-Side script tương tác với người sử dụng và tạo các trang web động. Nó là một ngôn ngữ chương trình thông dịch với các khả năng hướng đối tượng.

## 2.8. Giới thiệu về công cụ lập trình Visual Studio Code

Visual Studio Code (thường được gọi là VS Code) là một trình biên tập mã nguồn mở và miễn phí, được phát triển bởi Microsoft. Nó là một công cụ lập trình đa nền tảng mạnh mẽ, hỗ trợ nhiều ngôn ngữ lập trình và tích hợp nhiều tính năng tùy chỉnh.

Dưới đây là một số điểm nổi bật của Visual Studio Code:

- Đa nền tảng: VS Code có sẵn trên Windows, macOS và Linux, cho phép bạn phát triển trên bất kỳ hệ điều hành nào theo sở thích của bạn.
- Hỗ trợ nhiều ngôn ngữ: VS Code hỗ trợ nhiều ngôn ngữ lập trình phổ biến như JavaScript, TypeScript, Python, C++, C#, Java, Go, HTML, CSS và nhiều ngôn ngữ khác. Nó cung cấp các tính năng như kiểm tra cú pháp, gợi ý mã, định dạng mã tự động và nhiều hơn nữa.
- Tích hợp Git: VS Code tích hợp mạnh mẽ với hệ thống quản lý phiên bản Git, cho phép bạn thực hiện các hoạt động như commit, push, pull và xem lịch sử phiên bản trực tiếp từ giao diện của nó.
- Extensions và Marketplace: VS Code cho phép bạn mở rộng tính năng của nó thông qua extensions. Có hàng ngàn extensions có sẵn trong Visual Studio Code Marketplace, cho phép bạn tùy chỉnh và mở rộng IDE theo nhu cầu của mình. Bạn có thể tìm thấy các extensions liên quan đến ngôn ngữ, framework, công cụ hỗ trợ, giao diện và nhiều thứ khác.
- Tự động hoàn thành mã: VS Code cung cấp chức năng tự động hoàn thành mã thông minh, giúp tiết kiệm thời gian và tăng năng suất. Nó hiển thị gợi ý mã và thông tin API dựa trên ngôn ngữ và context của dự án của bạn.
- Debugging: VS Code hỗ trợ tính năng debug mạnh mẽ. Bạn có thể thiết lập các điểm dừng, theo dõi biến và xem ngăn xếp gọi hàm trong quá trình debug.
- Hỗ trợ Terminal tích hợp: VS Code cung cấp một terminal tích hợp cho phép người dùng thực thi các lệnh hệ thống và lệnh liên quan đến dự án từ cùng một giao diện.

- Giao diện người dùng thân thiện: VS Code có một giao diện người dùng đơn giản, tối giản và dễ sử dụng. Nó cho phép bạn tùy chỉnh giao diện và bố trí theo ý thích của người dùng.

## CHƯƠNG 3. PHÂN TÍCH THIẾT KẾ HỆ THỐNG

### 3.1. Khảo sát hệ thống thi trực tuyến

Một hệ thống thi trực tuyến được tổ chức như sau:

- Mỗi tài khoản chỉ thuộc một quyền và một quyền có nhiều tài khoản khác nhau.
- Thí sinh có thông tin bao gồm: họ tên, ngày sinh, giới tính, địa chỉ, lớp.
- Mỗi thí sinh đều có riêng 1 tài khoản đăng nhập là email và password. Email đăng nhập sẽ là email mà thí sinh đó đã đăng ký.
- Mỗi lớp học sẽ có nhiều môn học, mỗi môn học sẽ thuộc về nhiều lớp học
- Mỗi câu hỏi sẽ có thông tin câu hỏi, đáp án A, B, C, D, độ khó, đáp án đúng. Mỗi câu hỏi thuộc một lớp và một môn.
- Mỗi đề thi sẽ có thời gian làm bài, số câu hỏi, mô tả, tên đề thi. Mỗi đề thi thuộc một lớp và một môn.
- Mỗi ca thi sẽ có một đề của một môn lớp, mỗi đề sẽ có nhiều ca thi.
- Mỗi bài thi của thí sinh sẽ có các thông tin về thời gian hoàn thành, điểm số.
- Mỗi đáp án của thí sinh sẽ thuộc một câu hỏi, một bài thi sẽ bao gồm đáp án thí sinh đã lựa chọn, kết quả của lựa chọn đó.

### 3.2. Các chức năng của hệ thống

#### 3.2.1. Yêu cầu chức năng của hệ thống

Đối tượng tham gia hệ thống:

- Thí sinh
- Quản lý trang web

Chức năng của mỗi đối tượng:

#### ❖ Quản lý:

- Quản trị nội dung câu hỏi
- Thêm, sửa, xóa câu hỏi
- Quản trị đề thi
- Thêm, sửa, xóa đề thi
- Thêm và xóa câu hỏi khỏi đề thi
- Quản trị ca thi
- Thêm, sửa, xóa ca thi
- Phân loại đề thi theo ca thi
- Quản lý thí sinh
- Xem danh sách thí sinh
- Khóa thí sinh

#### ❖ Thí sinh:

- Đăng ký: Khi người dùng muốn sử dụng website thì có thể thực hiện chức năng đăng ký để trở thành thí sinh và thực hiện chức năng thi.
- Đăng nhập: Sử dụng tài khoản đã đăng ký để thực hiện đăng nhập vào hệ thống.
- Đăng xuất: Người dùng đăng xuất khỏi hệ thống
- Thí sinh có thể xem kết quả của bài thi thử sau khi hoàn thành.
- Thí sinh có thể xem lịch sử thi của mình, và kết quả sau mỗi lần thi.

### 3.2.2. Yêu cầu phi chức năng

- Hệ thống đáng tin cậy, chính xác, giao diện thân thiện, dễ sử dụng, truy cập dữ liệu nhanh chóng và hiệu quả.
- Đảm bảo tính bảo mật cho người điều hành hệ thống.
- Đảm bảo tính toàn vẹn dữ liệu phải định kỳ lưu giữ thông tin để phòng khi có sự cố còn có thể khắc phục một cách dễ dàng. Có sao lưu và phục hồi cơ sở dữ liệu khi gặp sự cố.
- Quyền hạn mỗi người dùng được bảo vệ chặt chẽ. Chỉ có quản lý mới có quyền xóa và cập nhật dữ liệu.

### 3.3. Mô hình chức năng nghiệp vụ (BDF – Business Function Diagram)

Chức năng là công việc mà tổ chức cần làm và được phân theo nhiều mức từ tổng hợp đến chi tiết.

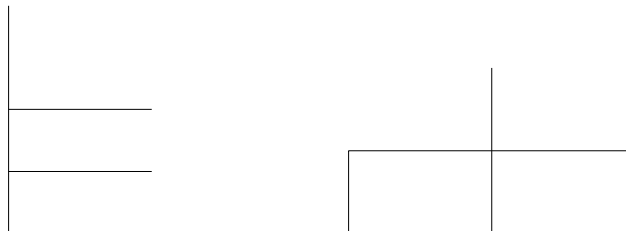
Mỗi chức năng có một tên duy nhất, các chức năng khác nhau phải có tên khác nhau. Để xác định tên cho các chức năng một cách chính xác và cụ thể, có thể bàn luận và nhất trí với người sử dụng.

Hình thức biểu diễn: Hình chữ nhật:

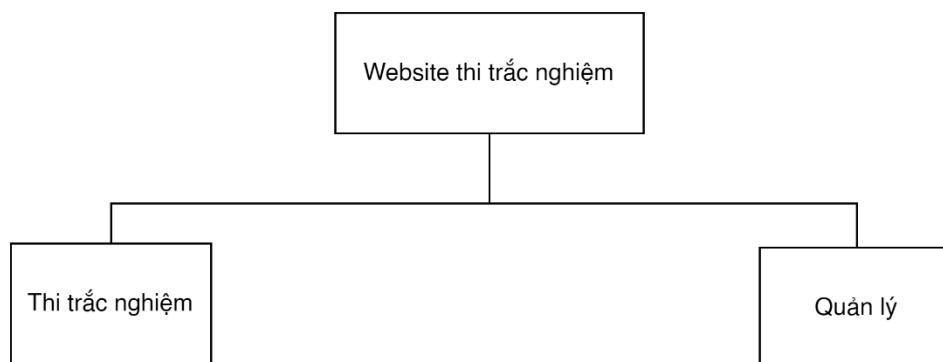
Tên chức năng

*Mỗi chức năng được phân rã thành các chức năng con. Các chức năng con có quan hệ phân cấp với chức năng cha.*

*Biểu diễn mối quan hệ phân cấp chức năng như sau:*



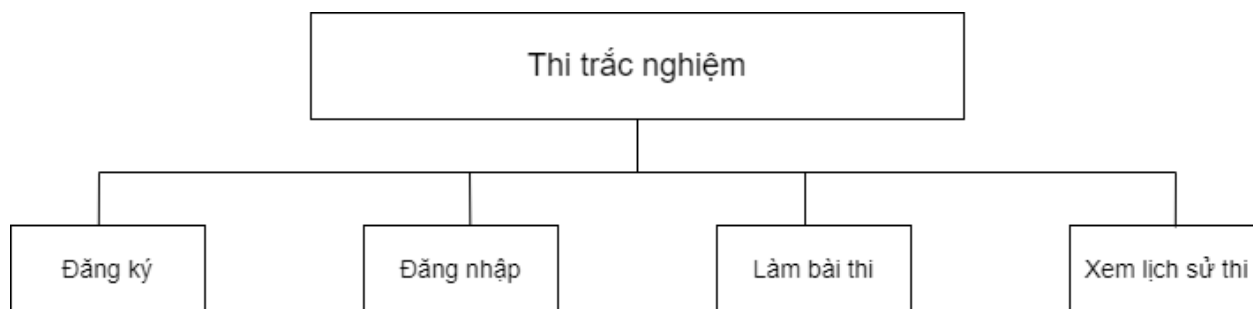
### 3.3.1. Sơ đồ phân cấp chức năng mức đỉnh



*Sơ đồ 3.1 Sơ đồ phân cấp chức năng mức đỉnh*

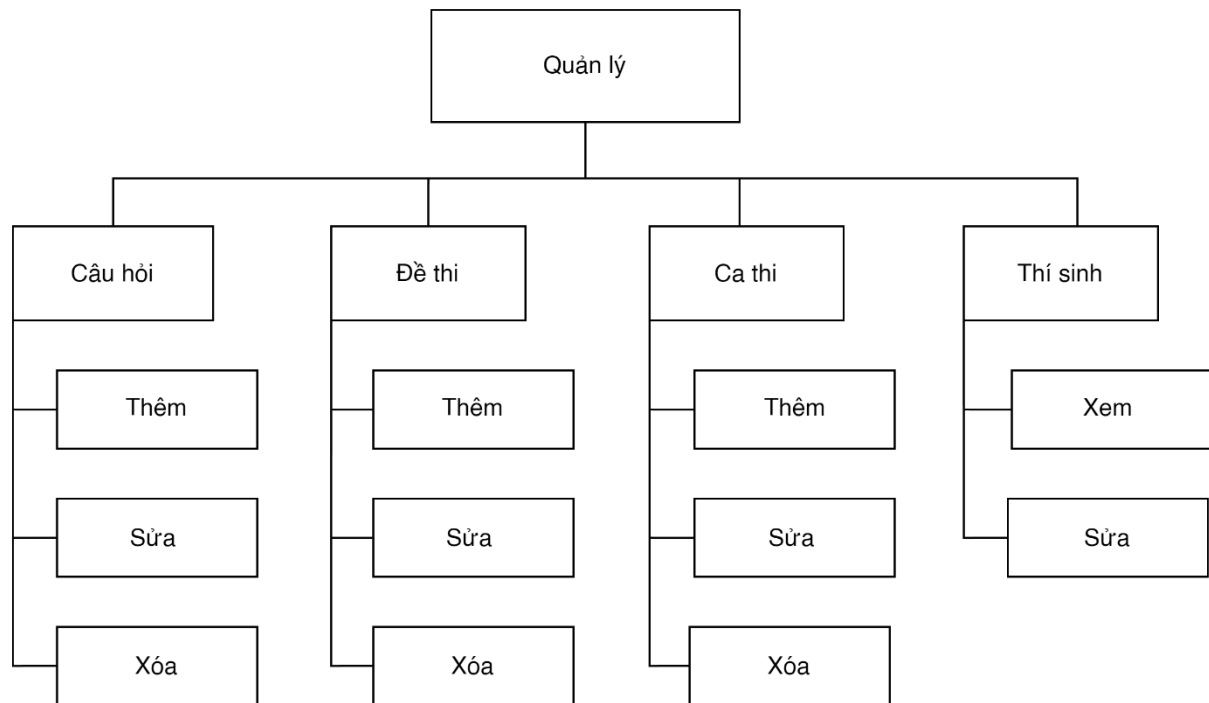
Hệ thống nghiệp vụ được xét gồm các chức năng chính: thi trắc nghiệm và quản lý. Mỗi chức năng này được phân rã thành một số chức năng nhỏ hơn nhằm phục vụ cho mục đích đảm bảo các thao tác trên trang website được vận hành đầy đủ.

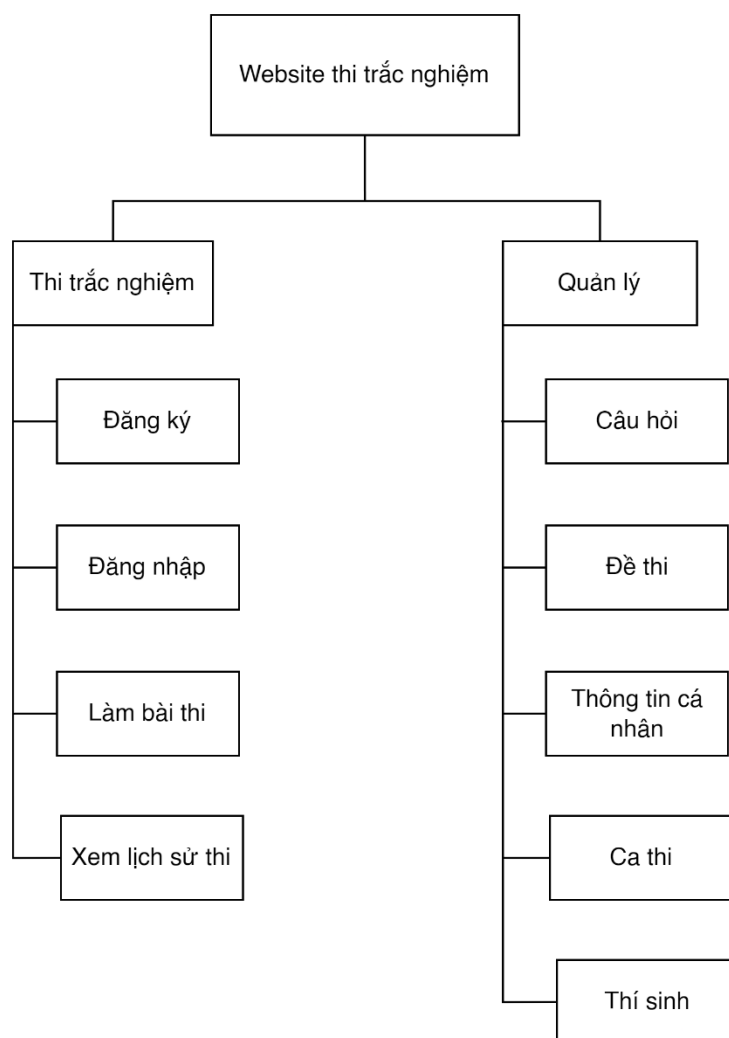
#### 3.3.1.1. Chức năng thi trắc nghiệm



*Sơ đồ 3.2 Sơ đồ phân cấp chức năng thi trắc nghiệm*



**3.3.1.2. Chức năng quản lý***Sơ đồ 3.3 Sơ đồ phân cấp chức năng quản lý*



Sơ đồ 3.4 Sơ đồ phân rã chức năng hoàn chỉnh

### 3.4. Mô hình hóa yêu cầu

#### 3.4.1. Các tác nhân của hệ thống

Tác nhân	Ý nghĩa, nhiệm vụ
Quản lý	Tác nhân có quyền cao nhất trong hệ thống. Quản lý có quyền xem danh sách thí sinh, điều hành, quản lý, theo dõi các đối tượng chính như câu hỏi, đề thi, ca thi, thông tin của thí sinh.
Thí sinh	Thí sinh truy cập vào website, xem danh sách ca thi, xem lịch thi của bản thân, thực hiện thi trắc nghiệm và xem lịch sử các bài đã thi

Bảng 3.1 Bảng xác định các tác nhân của hệ thống

**3.4.2. Xác định các usecase**

STT	Chức năng	Mô tả
1	Đăng ký	Chức năng cho phép người dùng tạo tài khoản đăng nhập vào hệ thống
2	Đăng nhập	Chức năng cho phép quản lý hay thí sinh nhập vào hệ thống để thực hiện các chức năng của họ.
3	Chỉnh sửa thông tin cá nhân	Chức năng cho phép người dùng chỉnh sửa thông tin cá nhân của mình như sdt, email,....
4	Đăng ký thi	Chức năng cho phép thí sinh có thể đăng ký vào một ca thi đang có trên hệ thống
5	Xem lịch thi	Chức năng cho phép thí sinh có thể xem lịch thi của bản thân
6	Làm bài thi trắc nghiệm	Chức năng cho phép thí sinh có thể làm bài thi trắc nghiệm
7	Xem lịch sử thi	Chức năng cho phép thí sinh xem lại các bài thi đã làm
8	Quản lý câu hỏi	Chức năng cho phép quản lý tạo câu hỏi, chỉnh sửa câu hỏi, quản lý bộ câu hỏi
9	Quản lý đề thi	Chức năng cho phép tạo đề thi, thêm câu hỏi vào đề thi, sửa đề thi và xóa đề thi.
10	Quản lý ca thi	Chức năng cho phép quản lý tạo các loại ca thi, sửa ca thi, và xóa ca thi.
11	Quản lý thí sinh	Quản lý có thể xem danh sách thí sinh đã đăng ký trên hệ thống, khóa thí sinh.

*Bảng 3.2 Xác định các usecase*

Từ các chức năng đã có nhóm chúng em đã suy ra được những use case sau:

- Use case xem các ca thi
- Use case đăng ký
- Use case đăng nhập
- Use case đăng xuất
- Use case chỉnh sửa thông tin cá nhân

Use case các chức năng của thí sinh:

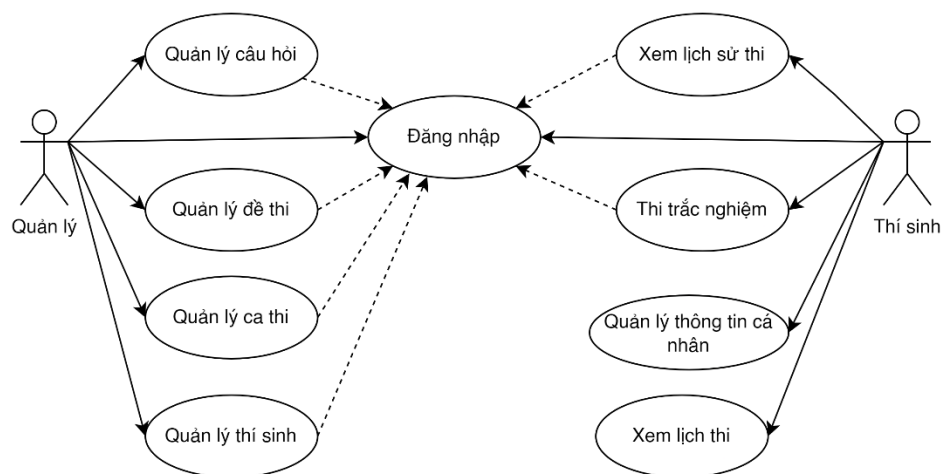
- Use case đăng ký thi
- Use case xem lịch thi
- Use case thi trắc nghiệm
- Use case xem lịch sử thi

Use case các chức năng của quản lý:

- Use case quản lý câu hỏi
- Use case quản lý đề thi
- Use case quản lý ca thi
- Use case quản lý thí sinh

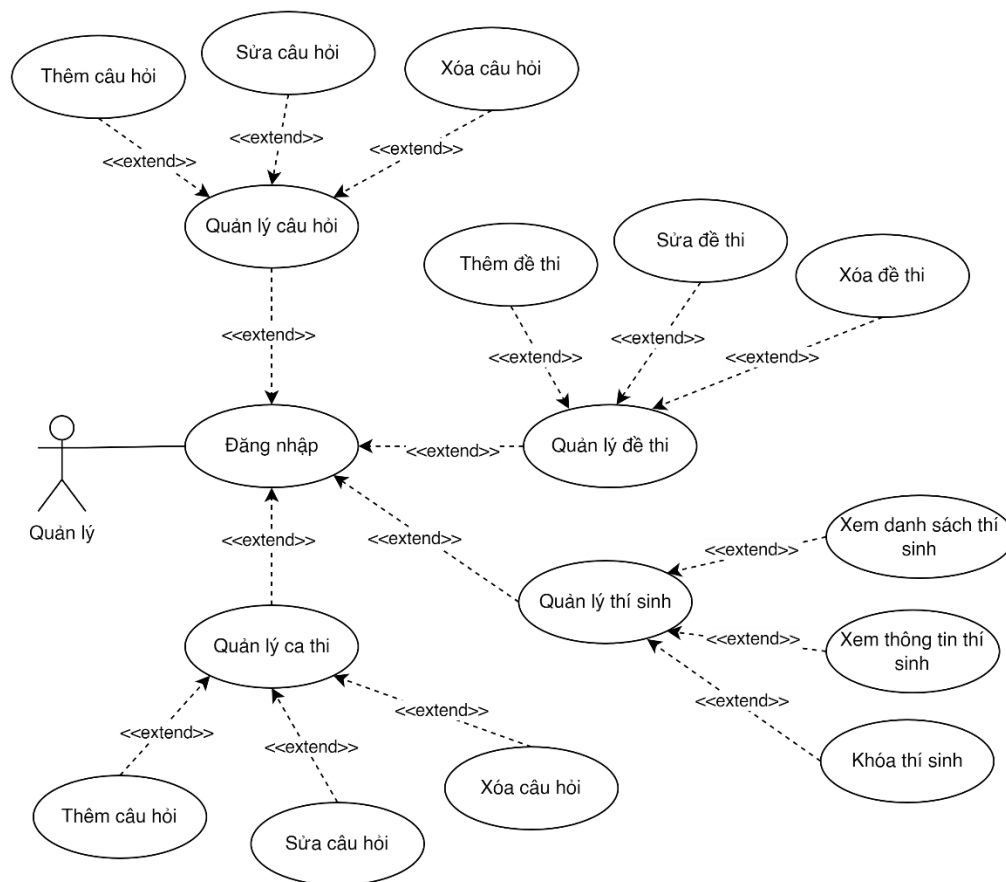
### 3.4.3. Sơ đồ usecase

#### 3.4.3.1. Sơ đồ usecase tổng quát



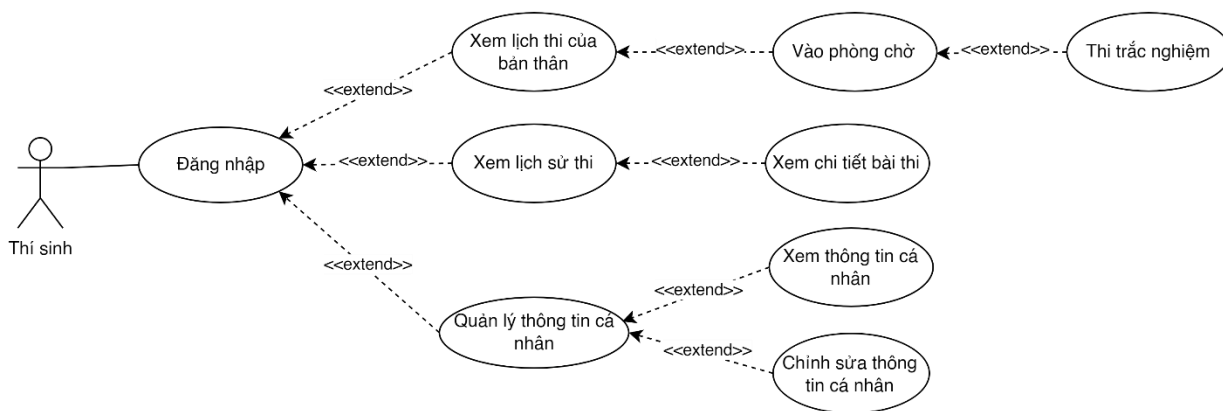
Sơ đồ 3.5 Usecase tổng quát

#### 3.4.3.2. Sơ đồ usecase các chức năng của quản lý

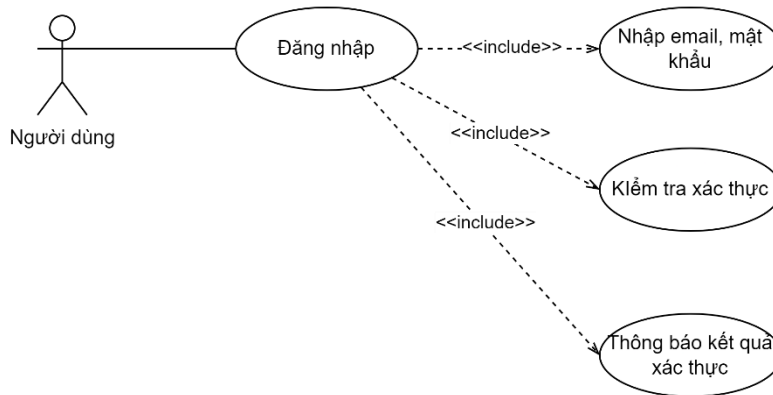


*Sơ đồ 3.6 Usecase các chức năng của quản lý*

#### 3.4.3.3. Sơ đồ usecase các chức năng của thí sinh



*Sơ đồ 3.7 Usecase các chức năng của thí sinh*

**3.4.3.4. Sơ đồ usecase đăng nhập***Sơ đồ 3.8 Sơ đồ usecase đăng nhập*

Mô tả: Người dùng dùng tài khoản của mình để đăng nhập hệ thống theo phân quyền. Mỗi quyền sẽ được sử dụng chức năng khác nhau trong hệ thống.

Tên Use Case	Đăng nhập
Tác nhân chính	Người dùng
Điều kiện trước	Tài khoản phải tồn tại trong cơ sở dữ liệu
Đảm bảo tối thiểu	Cho phép đăng nhập nếu tài khoản không bị khóa
Điều kiện sau	Đăng nhập thành công
Chuỗi sự kiện chính: <ol style="list-style-type: none"> <li>1. Người dùng nhập các trường email, password để đăng nhập.</li> <li>2. Hệ thống xác thực email, password.</li> <li>3. Báo lỗi nếu sai.</li> <li>4. Đăng nhập thành công khi không có lỗi.</li> </ol>	

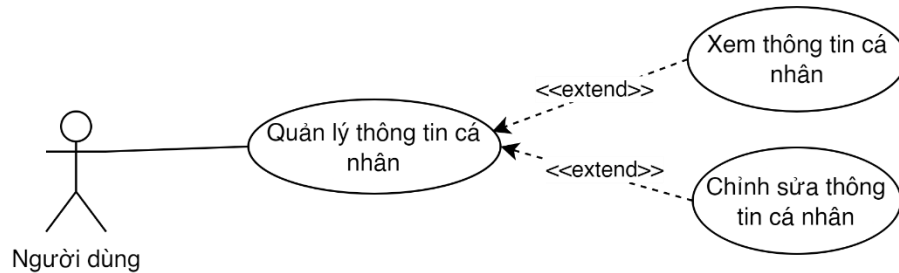
**3.4.3.5. Sơ đồ usecase đăng xuất***Sơ đồ 3.9 Sơ đồ usecase đăng xuất*

Mô tả: Người dùng có thể đăng xuất khỏi tài khoản.

Tên Use Case	Đăng xuất
Tác nhân chính	Người dùng
Điều kiện trước	Tài khoản đang đăng nhập

Đảm bảo tối thiểu	Cho phép đăng xuất
Điều kiện sau	Tài khoản không tồn tại/ không hiện hữu tên người dùng trên website
Chuỗi sự kiện chính: <ol style="list-style-type: none"> <li>Hệ thống đăng xuất tài khoản người dùng.</li> <li>Hệ thống ghi nhận người dùng đang không đăng nhập.</li> </ol>	

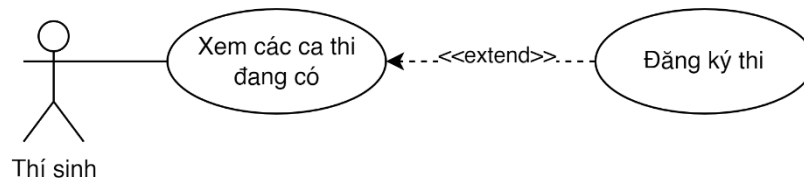
### 3.4.3.6. Sơ đồ usecase thay đổi thông tin cá nhân



Sơ đồ 3.10 Sơ đồ usecase thay đổi thông tin cá nhân

Tên Use Case	Chỉnh sửa thông tin cá nhân
Tác nhân chính	Thí sinh, quản lý
Điều kiện trước	Người dùng đã đăng nhập thành công vào hệ thống
Đảm bảo tối thiểu	Rollback quy trình khi có lỗi
Điều kiện sau	Thông tin được thay đổi thành công
Chuỗi sự kiện chính <ol style="list-style-type: none"> <li>Người dùng chọn chức năng xem thông tin cá nhân</li> <li>Hệ thống hiển thị giao diện chứa các thông tin của người dùng</li> <li>Người dùng thay đổi các thông tin cần thiết và xác nhận cập nhật</li> <li>Hệ thống kiểm tra ràng buộc, nếu vi phạm thì thông báo lỗi</li> <li>Hệ thống cập nhật dữ liệu và hiển thị thay đổi thông tin thành công</li> </ol>	

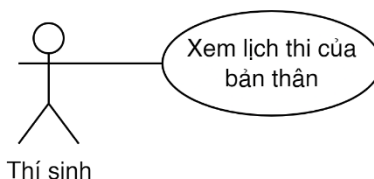
### 3.4.3.7. Sơ đồ usecase đăng ký thi



Sơ đồ 3.11 Sơ đồ usecase đăng ký thi

Tên Use Case	Đăng ký thi
Tác nhân chính	Thí sinh
Điều kiện trước	Thí sinh đã đăng nhập thành công vào hệ thống
Đảm bảo tối thiểu	Rollback quy trình khi có lỗi
Điều kiện sau	Thí sinh đăng ký vào ca thi thành công
Chuỗi sự kiện chính <ol style="list-style-type: none"> <li>1. Thí sinh chọn chức năng xem các ca thi đang có</li> <li>2. Hệ thống hiển thị giao diện chứa các ca thi mà người dùng chưa đăng ký</li> <li>3. Người dùng chọn ca thi phù hợp với bản thân</li> <li>4. Nhân đăng ký</li> </ol>	

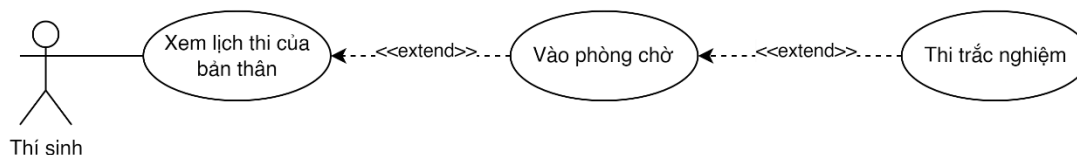
#### 3.4.3.8. Sơ đồ usecase xem lịch thi



Sơ đồ 3.12 Sơ đồ usecase xem lịch thi

Tên Use Case	Xem lịch thi
Tác nhân chính	Thí sinh
Điều kiện trước	Thí sinh đã đăng nhập thành công vào hệ thống
Đảm bảo tối thiểu	Thí sinh đã đăng ký ít nhất 1 ca thi
Điều kiện sau	Thí sinh xem được lịch thi
Chuỗi sự kiện chính <ol style="list-style-type: none"> <li>1. Thí sinh chọn chức năng xem lịch thi</li> <li>2. Hệ thống hiển thị giao diện lịch thi của thí sinh</li> </ol>	

#### 3.4.3.9. Sơ đồ usecase làm bài thi trắc nghiệm

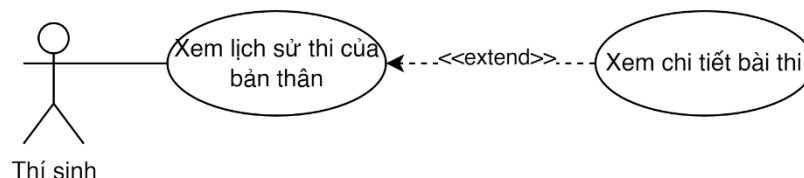


Sơ đồ 3.13 Sơ đồ usecase làm bài thi trắc nghiệm



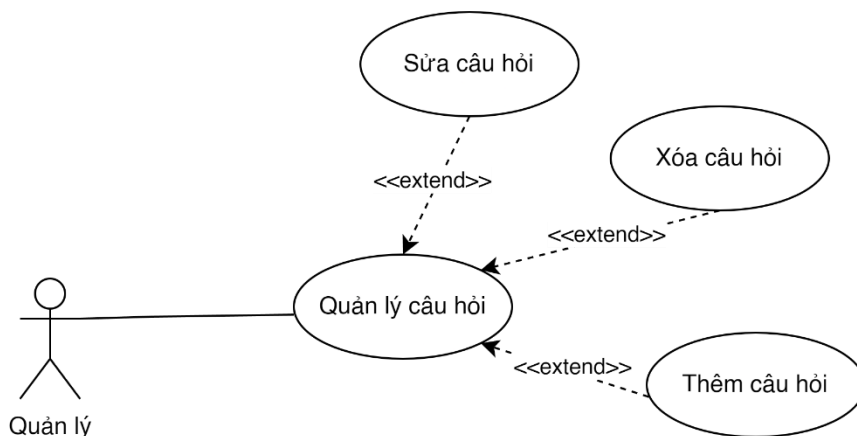
Tên Use Case	Thi trắc nghiệm
Tác nhân chính	Thí sinh
Điều kiện trước	Thí sinh đã đăng nhập thành công vào hệ thống
Đảm bảo tối thiểu	Rollback quy trình khi có lỗi
Điều kiện sau	Thí sinh thi thành công
Chuỗi sự kiện chính <ol style="list-style-type: none"> <li>1. Thí sinh chọn chức năng xem lịch thi</li> <li>2. Hệ thống hiển thị giao diện lịch thi của thí sinh</li> <li>3. Thí sinh chọn ca thi và ấn “Start test”</li> <li>4. Hệ thống sẽ đưa thí sinh vào phòng chờ</li> <li>5. Thí sinh xem thông tin về bài thi và ấn “Start test” để xác nhận vào thi</li> <li>6. Thí sinh tiến hành làm bài thi</li> <li>7. Thí sinh bấm “Submit” để hoàn thành bài thi</li> </ol>	

### 3.4.3.10. Sơ đồ usecase xem lịch sử thi

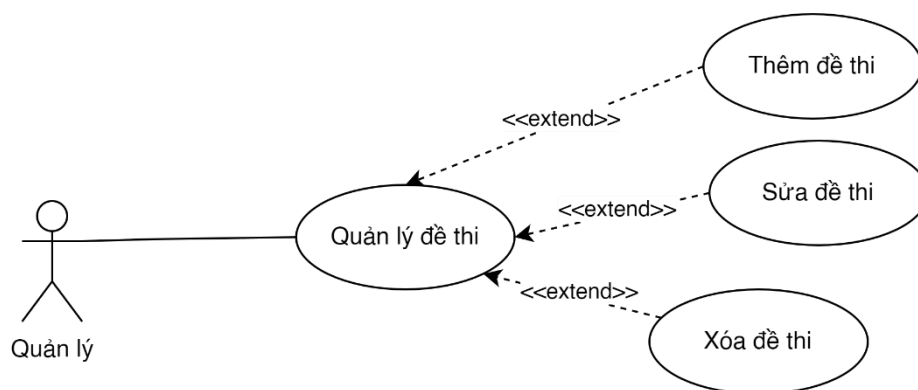


Sơ đồ 3.14 Sơ đồ usecase xem lịch sử thi

Tên Use Case	Xem lịch sử thi
Tác nhân chính	Thí sinh
Điều kiện trước	Thí sinh đã đăng nhập thành công vào hệ thống
Đảm bảo tối thiểu	Thí sinh đã đăng ký ít nhất 1 ca thi
Điều kiện sau	Thí sinh xem được lịch sử thi
Chuỗi sự kiện chính <ol style="list-style-type: none"> <li>1. Thí sinh chọn chức năng xem lịch sử thi</li> <li>2. Hệ thống hiển thị giao diện lịch sử thi của thí sinh</li> <li>3. Thí sinh có thể xem chi tiết từng bài thi mà mình đã làm</li> </ol>	

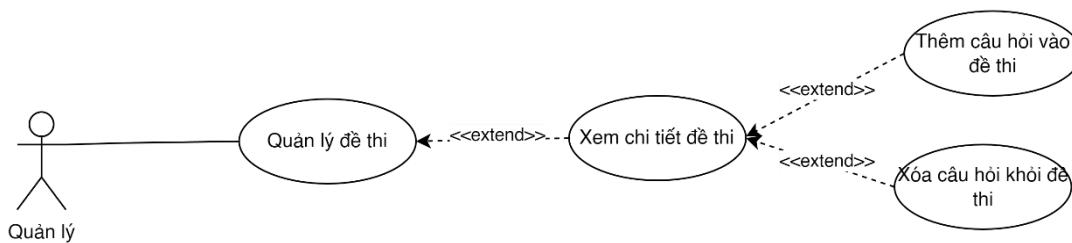
**3.4.3.11. Sơ đồ usecase quản lý câu hỏi***Sơ đồ 3.15 Sơ đồ usecase quản lý câu hỏi*

Tên Use Case	Quản lý câu hỏi
Tác nhân chính	Quản lý
Điều kiện trước	Quản lý đã đăng nhập thành công vào hệ thống
Đảm bảo tối thiểu	Rollback quy trình khi có lỗi
Điều kiện sau	Quy trình diễn ra thành công
Chuỗi sự kiện chính <ol style="list-style-type: none"> <li>Vào chức năng quản lý ca thi</li> <li>Hệ thống hiển thị danh sách các câu hỏi</li> <li>Quản lý thực hiện thêm/ xóa/ sửa câu hỏi</li> <li>Hệ thống thông báo lỗi hoặc thành công nếu không sai trường dữ liệu.</li> </ol>	

**3.4.3.12. Sơ đồ usecase quản lý đề thi***Sơ đồ 3.16 Sơ đồ usecase quản lý đề thi*

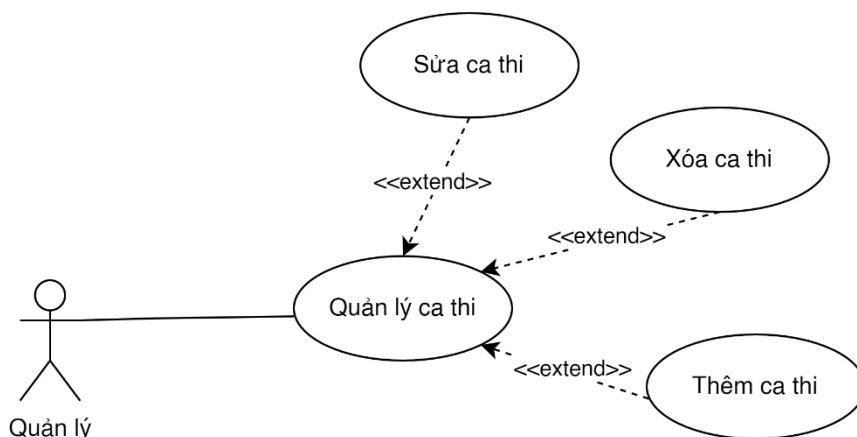
Tên Use Case	Quản lý đề thi
Tác nhân chính	Quản lý
Điều kiện trước	Quản lý đã đăng nhập thành công vào hệ thống
Đảm bảo tối thiểu	Rollback quy trình khi có lỗi
Điều kiện sau	Quy trình diễn ra thành công
Chuỗi sự kiện chính <ol style="list-style-type: none"> <li>1. Vào chức năng quản lý đề thi</li> <li>2. Hệ thống hiển thị danh sách các đề thi</li> <li>3. Quản lý thực hiện thêm/ xóa/ sửa đề thi</li> <li>4. Hệ thống thông báo lỗi hoặc thành công nếu không sai trường dữ liệu.</li> </ol>	

### 3.4.3.13. Sơ đồ usecase thêm câu hỏi vào đề thi

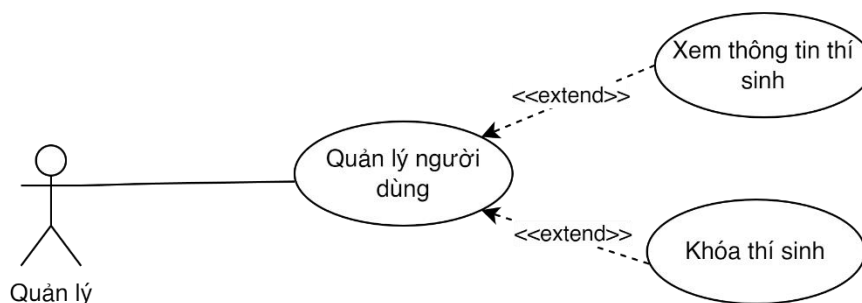


Sơ đồ 3.17 Sơ đồ usecase thêm câu hỏi vào đề thi

Tên Use Case	Thêm câu hỏi vào đề thi
Tác nhân chính	Quản lý
Điều kiện trước	Quản lý đã đăng nhập thành công vào hệ thống
Đảm bảo tối thiểu	Rollback quy trình khi có lỗi
Điều kiện sau	Quy trình diễn ra thành công
Chuỗi sự kiện chính <ol style="list-style-type: none"> <li>1. Vào chức năng quản lý đề thi</li> <li>2. Hệ thống hiển thị danh sách các đề thi</li> <li>3. Quản lý chọn đề thi mình muốn thêm và bấm vào chi tiết</li> <li>4. Hệ thống hiển thị danh sách câu hỏi có trong đề thi và bộ câu hỏi</li> <li>5. Quản lý thực hiện thêm câu hỏi vào đề thi hoặc xóa câu hỏi ra khỏi đề thi</li> <li>6. Hệ thống thông báo lỗi hoặc thành công nếu không sai trường dữ liệu.</li> </ol>	

**3.4.3.14. Sơ đồ usecase quản lý ca thi***Sơ đồ 3.18 Sơ đồ usecase quản lý ca thi*

Tên Use Case	Quản lý ca thi
Tác nhân chính	Quản lý
Điều kiện trước	Quản lý đã đăng nhập thành công vào hệ thống
Đảm bảo tối thiểu	Rollback quy trình khi có lỗi
Điều kiện sau	Quy trình diễn ra thành công
Chuỗi sự kiện chính <ol style="list-style-type: none"> <li>1. Vào chức năng quản lý ca thi</li> <li>2. Hệ thống hiển thị danh sách các ca thi</li> <li>3. Quản lý thực hiện thêm/ xóa/ sửa ca thi</li> <li>4. Hệ thống thông báo lỗi hoặc thành công nếu không sai trường dữ liệu.</li> </ol>	

**3.4.3.15. Sơ đồ usecase quản lý thí sinh***Sơ đồ 3.19 Sơ đồ usecase quản lý thí sinh*

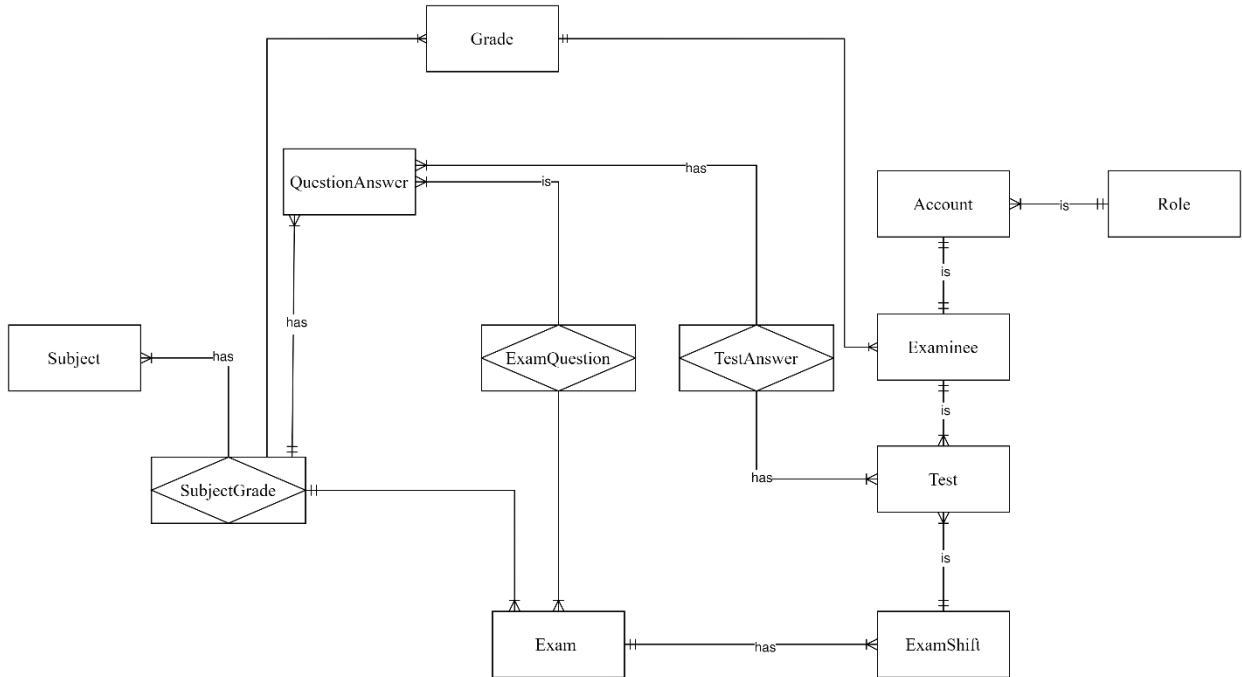
Tên Use Case	Khóa tài khoản thí sinh
Tác nhân chính	Quản lý
Điều kiện trước	Quản lý đã đăng nhập thành công vào hệ thống
Đảm bảo tối thiểu	Rollback quy trình khi có lỗi
Điều kiện sau	Khóa tài khoản thành công
Chuỗi sự kiện chính <ol style="list-style-type: none"> <li>1. Vào chức năng quản lý tài khoản</li> <li>2. Hệ thống hiển thị danh sách các tài khoản</li> <li>3. Quản lý thực hiện khóa tài khoản thí sinh</li> </ol>	

### 3.5. Mô hình thực thể (ERD – Entity Relationship Diagram)

#### 3.5.1. Xác định thực thể

1. accounts(Id, Email, Phone, Password, LockEnable, CreateDate)
2. examinees(Id, FullName, Birth, Gender, AddressDetail)
3. roles(Id, Name)
4. subjects(Id, Name, Value)
5. question\_answers(Id, Content, Difficulty, OptionA, OptionB, OptionC, OptionD, CorrectOption)
6. exams (Id, Name, Description, Duration, TotalQuestions)
7. exam\_questions(Id, Sequence)
8. exam\_shifts(Id, Name, StartTime, EndTime)
9. tests (Id, Score, CompletionTime)
10. test\_answers(Id, SelectedOption, IsCorrect)

### 3.5.2. Mô hình ERD



Hình 3.1 Mô hình ERD

### 3.5.3. Mô hình cơ sở dữ liệu quan hệ

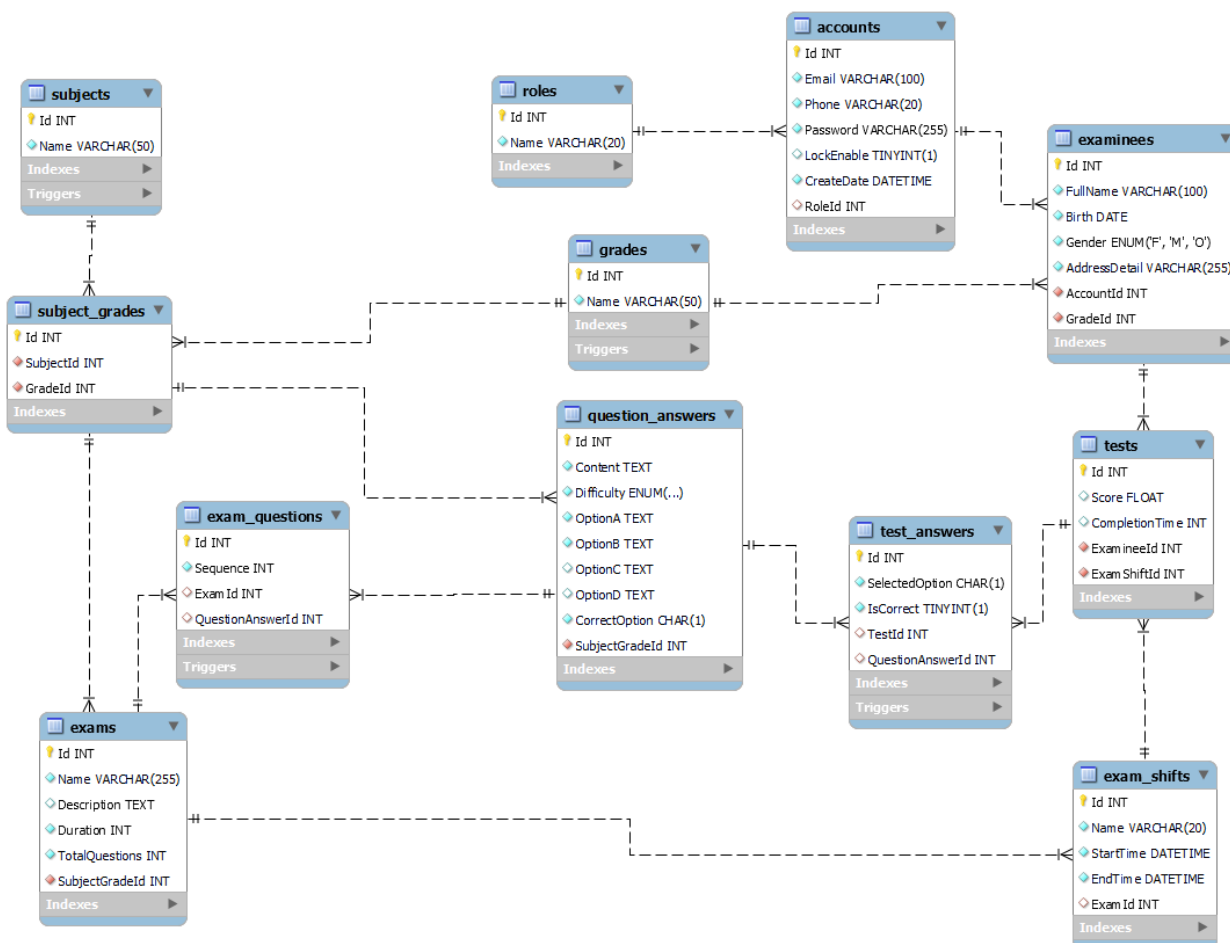
**Quy ước: Primary key:** (in đậm) **MABP**, **Foreign key:** (in nghiêng, gạch chân) **MABP**

**PK,FK:** (vừa thuộc khóa chính, vừa là khóa ngoại) **MAPD, MAHP**

1. accounts(**Id**, Email, Phone, Password, LockEnable, CreateDate, RoleId)
2. examinees(**Id**, FullName, Birth, Gender, AddressDetail, AccountId, GradeId)
3. roles(**Id**, Name)
4. subjects(**Id**, Name)
5. grades(**Id**, Name)
6. subject\_grades(**Id**, SubjectId, GradeId)
7. question\_answers(**Id**, Content, Difficulty, OptionA, OptionB, OptionC, OptionD, CorrectOption, SubjectGradeId)
8. exams (**Id**, Name, Description, Duration, TotalQuestions, SubjectGradeId)
9. exam\_questions(**Id**, Sequence, ExamId, QuestionAnswerId)
10. exam\_shifts(**Id**, Name, StartTime, EndTime, ExamId)
11. tests (**Id**, Score, CompletionTime, ExamineeId, ExamShiftId)
12. test\_answers(**Id**, SelectedOption, IsCorrect, TestId, QuestionAnswerId)

### 3.6. Thiết kế dữ liệu

### 3.6.1. Mô hình Diagram



*Hình 3.2 Mô hình Diagram*

### 3.6.2. Từ điển dữ liệu

1. accounts(***Id***, Email, Phone, Password, LockEnable, CreateDate, *RoleId*)

STT	Tên	Kiểu dữ liệu	Độ dài	Ý nghĩa	Ghi chú
1	Id	int	-	Primary Key	
2	Email	varchar(100)	100	Địa chỉ email	Unique, Not Null
3	Phone	varchar(20)	20	Số điện thoại	Unique, Not Null
4	Password	varchar(255)	255	Mật khẩu	Not Null
5	LockEnable	Tinyint(1)	1	Tình trạng khóa tài khoản	Default False
6	CreateDate	datetime	-	Ngày tạo tài khoản	Default Now()

7	RoleId	int	-	Khóa ngoại tới vai trò	Foreign Key: Role(id)
---	--------	-----	---	------------------------	-----------------------

Bảng 3.3 Bảng thực thể accounts

2. examinees(**Id**, FullName, Birth, Gender, AddressDetail, AccountId)

STT	Tên	Kiểu dữ liệu	Độ dài	Ý nghĩa	Ghi chú
1	Id	int	-	Primary Key	
2	FullName	varchar(100)	100	Họ và tên đầy đủ	Not Null
3	Birth	date	-	Ngày sinh	Not Null
4	Gender	enum	-	Giới tính	Nằm trong các loại (M, F, O), Not Null
5	AddressDetail	varchar(255)	255	Địa chỉ	Not Null
6	GradeId	int	-	Khóa ngoại đến lớp	Foreign Key: Grade(id), Unique, Not null
7	AccountId	int	-	Khóa ngoại tới tài khoản	Foreign Key: Account(id), Unique, Not null

Bảng 3.4 Bảng thực thể examinees

3. roles(**Id**, Name)

STT	Tên	Kiểu dữ liệu	Độ dài	Ý nghĩa	Ghi chú
1	Id	int	-	Primary Key	
2	Name	varchar(20)	20	Tên vai trò	Unique, Not Null

Bảng 3.5 Bảng thực thể roles

4. grades(**Id**, Name)

1	Id	int	-	Primary Key	
2	Name	varchar(50)	50	Tên môn học	Unique, Not Null

Bảng 3.6 Bảng thực thể grades

5. subjects(**Id**, Name)

STT	Tên	Kiểu dữ liệu	Độ dài	Ý nghĩa	Ghi chú
-----	-----	--------------	--------	---------	---------



1	Id	int	-	Primary Key	
2	Name	varchar(50)	50	Tên môn học	Unique, Not Null

*Bảng 3.7 Bảng thực thể subjects*

6. subject\_grades(**Id**, SubjectId, GradeId)

STT	Tên	Kiểu dữ liệu	Độ dài	Ý nghĩa	Ghi chú
1	Id	int	-	Primary Key	
2	SubjectId	int	-	Khóa ngoại tới môn học	Foreign Key: Subject(Id), Not Null
3	GradeId	int	-	Khóa ngoại tới lớp	Foreign Key: Grade(Id), Not Null

*Bảng 3.8 Bảng thực thể subject\_grades*

7. question\_answers(**Id**, Content, Difficulty, OptionA, OptionB, OptionC, OptionD, CorrectOption, SubjectGradeId)

STT	Tên	Kiểu dữ liệu	Độ dài	Ý nghĩa	Ghi chú
1	Id	int	-	Primary Key	
2	Content	Text	-	Tên câu hỏi	Not Null
3	Difficulty	enum	-	Độ khó	Nằm trong các loại (Easy, Normal, Hard), Not Null
4	OptionA	Text	-	Đáp án A	Not Null
5	OptionB	Text	-	Đáp án B	Not Null
6	OptionC	Text	-	Đáp án C	
7	OptionD	Text	-	Đáp án D	
8	CorrectOption	Char(1)	1	Đáp án đúng	Nằm trong các loại (A, B, C, D), Not Null

9	SubjectGradeId	Int	-	Khóa ngoại tới môn học lớp	Foreign Key: SubjectGrade(Id)
---	----------------	-----	---	----------------------------	-------------------------------

Bảng 3.9 Bảng thực thể *question\_answers*

8. exams(***Id***, Name, Description, Duration, TotalQuestions, SubjectGradeId)

ST	Tên	Kiểu dữ liệu	Độ dài	Ý nghĩa	Ghi chú
1	Id	int	-	Primary Key	
2	Name	varchar(255)	255	Tên kỳ thi	Not Null
3	Description	Text	-	Mô tả	
4	TotalQuestions	int	-	Tổng số câu hỏi	Not Null
5	Duration	Int	-	Thời gian làm bài	Not Null
6	SubjectGradeId	Int	-	Khóa ngoại tới lớp môn học	Foreign Key: SubjectGrade(Id)

Bảng 3.10 Bảng thực thể *exams*

9. exam\_questions (***Id***, Sequence, ExamId, QuestionAnswerId)

STT	Tên	Kiểu dữ liệu	Độ dài	Ý nghĩa	Ghi chú
1	Id	int	-	Primary Key	
2	Sequence	int	-	Thứ tự	Not Null
3	ExamId	int	-	Khóa ngoại tới đề thi	Foreign Key: Exam(Id), Not Null
4	QuestionAnswerId	int	-	Khóa ngoại tới câu hỏi	Foreign Key: QuestionAnswer(Id), Not Null

Bảng 3.11 Bảng thực thể *exam\_questions*

10. exam\_shifts (***Id***, Name, StartTime, EndTime, ExamId)

STT	Tên	Kiểu dữ liệu	Độ dài	Ý nghĩa	Ghi chú
1	Id	int	-	Primary Key	

2	Name	varchar(20)	20	Ca thi số	Not Null
3	StartTime	datetime	-	Thời gian bắt đầu thi	Not Null
4	EndTime	datetime	-	Thời gian kết thúc thi	Not Null
5	ExamId	int	-	Khóa ngoại tới đề thi	Foreign Key: Exam(Id), Not Null

Bảng 3.12 Bảng thực thể exam\_shifts

11. tests (**Id**, Score, CompletionTime, ExamineeId, ExamShiftId)

STT	Tên	Kiểu dữ liệu	Độ	Ý nghĩa	Ghi chú
1	Id	int	-	Primary Key	
2	Score	float	-	Điểm	Not Null
3	Completion Time	int	-	Thời gian hoàn thành	Not Null
4	ExamineeId	int	-	Khóa ngoại tới thí sinh	Foreign Key: Examinee(Id) Not Null
5	ExamShiftId	int	-	Khóa ngoại tới ca thi	Foreign Key: ExamShift(Id) Not Null

Bảng 3.13 Bảng thực thể tests

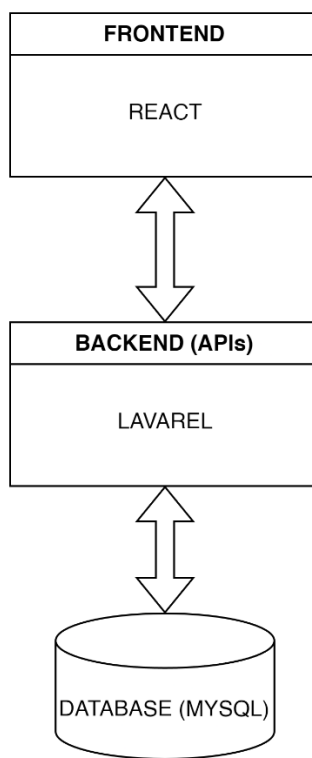
12. test\_answers (**Id**, SelectedOption, IsCorrect, TestId, QuestionAnswerId)

ST	Tên	Kiểu dữ liệu	Độ dài	Ý nghĩa	Ghi chú
1	Id	int	-	Primary Key	
2	SelectedOption	Char(1)	1	Đáp án đã chọn	Not Null

3	IsCorrect	Tinyint(1)	-	Làm đúng	Not Null, Nằm trong các giá trị sau: 1. Đúng, 0. Sai
4	TestId	int	-	Khóa ngoại tới bài thi	Foreign Key: Test(Id), Not Null
5	QuestionAnswerId	int	-	Khóa ngoại tới câu hỏi	Foreign Key: QuestionAnswer(Id), Not Null

Bảng 3.14 Bảng thực thể test\_answers

### 3.7. Thiết kế kiến trúc hệ thống



Hình 3.3 Mô hình kiến trúc của hệ thống

Mô hình kiến trúc gồm có các phần:

- ❖ Tương tác với người dùng (UI): Lớp này có nhiệm vụ chính là giao tiếp với người dùng như tiếp nhận thông tin, thông báo kết quả thực hiện, hiển thị thông tin theo yêu cầu người dùng, ... thông qua website là chính.

Website là giao diện tương tác chính với người dùng trong đó nhân viên quản lý hệ thống trên website sau khi đăng nhập tài khoản và thí sinh thực hiện thi trắc nghiệm trên website hệ thống.

- ❖ Hệ thống kết nối: Lớp này thực hiện kết nối giữa các lớp như giao diện, hệ thống xử lý và database.

Hệ thống kết nối từ giao diện tương tác với người dùng đến hệ thống xử lý thông qua API

Hệ thống kết nối từ hệ thống xử lý đến database (MySQL) thông qua thư viện JDBC

- ❖ Hệ thống xử lý: Thực hiện các hành động nghiệp vụ của phần mềm như tính toán, đánh giá tính hợp lệ của thông tin, ... Tầng này còn di chuyển, xử lý thông tin giữa 2 tầng trên dưới.

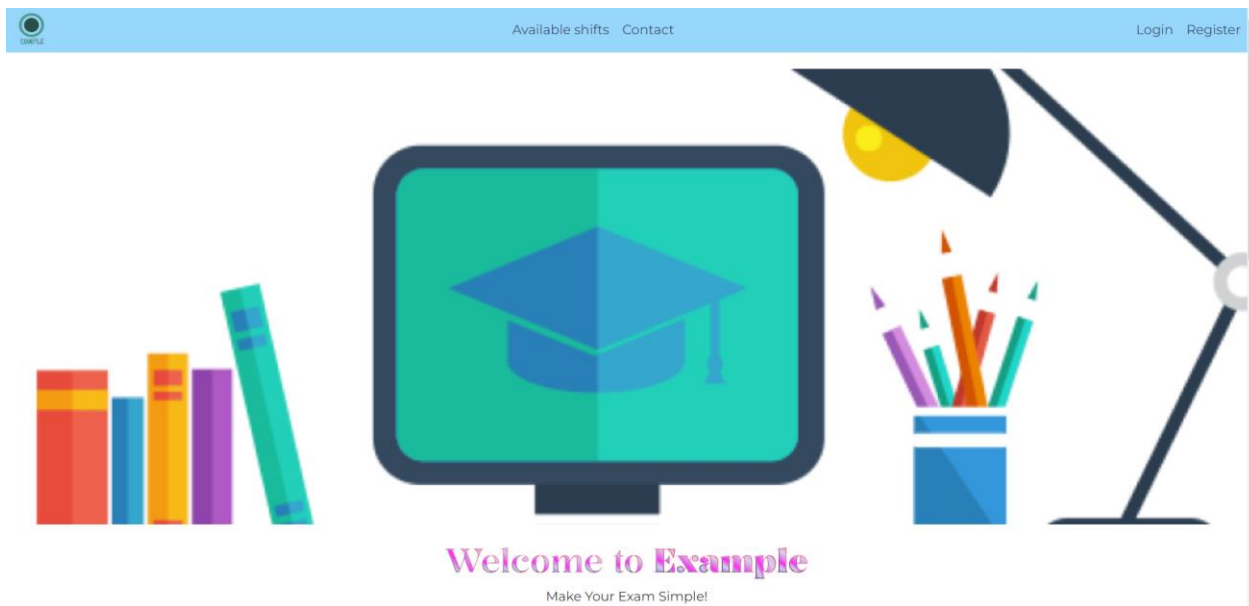
Lớp có 2 nhiệm vụ chính:

- Là nơi đáp ứng các yêu cầu thao tác dữ liệu của người dùng, xử lý dữ liệu từ website trả về trước khi truyền xuống database lưu vào cơ sở dữ liệu
  - Là nơi kiểm tra các ràng buộc, tính toàn vẹn và hợp lệ dữ liệu, thực hiện tính toán và xử lý các yêu cầu nghiệp vụ, trước khi trả kết quả thông báo đến người dùng.
- ❖ Database: lưu trữ thông tin dữ liệu hệ thống

## CHƯƠNG 4. DEMO CHƯƠNG TRÌNH

### 4.1. Giao diện trang chủ

- Giao diện được hiển thị đầu tiên khi người dùng chưa đăng nhập.
- Người dùng chưa đăng nhập có thể xem các ca thi đang có trên hệ thống.



Hình 4.1 Giao diện trang chủ

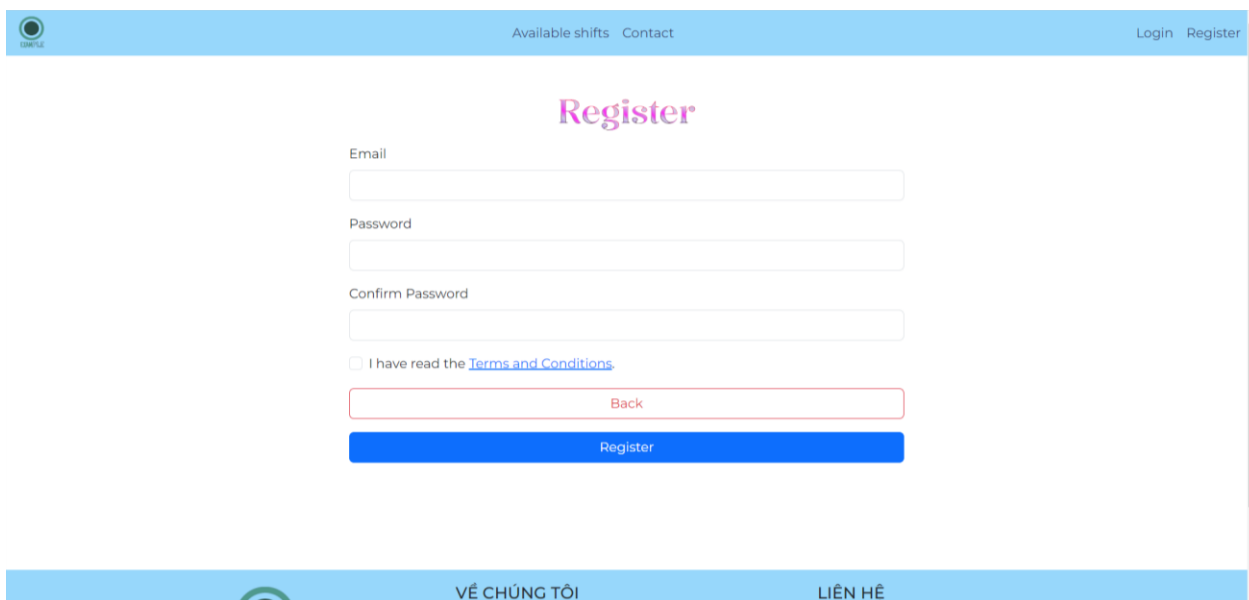
### 4.2. Giao diện đăng ký

- Nếu người dùng chưa có tài khoản có thể đăng ký tài khoản tại form Đăng ký. Khi đăng ký tài khoản, hệ thống sẽ kiểm tra các thông tin có hợp không và trả lại kết quả.
- Nếu các thông tin đăng ký hợp lệ sẽ thông báo đăng ký thành công và chuyển đến trang đăng nhập.

The screenshot shows a registration form titled 'Register' in a pink font. The form is set against a light blue background with a navigation bar at the top containing a logo, 'Available shifts', 'Contact', 'Login', and 'Register' links. The form fields include: 'Full Name' (text input), 'Birth Date' (date picker showing '02/02/2002'), 'Gender' (dropdown menu showing 'Male'), 'Phone' (text input), 'Grade' (dropdown menu showing 'Lớp 10'), and 'Address Detail' (text input showing '97 Man Thien'). At the bottom of the form is a 'Next' button.

Hình 4.2 Giao diện đăng ký (Trang 1)

- Khi đăng ký, người dùng phải điền các thông tin cá nhân, nhập email, mật khẩu, xác nhận lại mật khẩu, và phải đồng ý các điều khoản từ hệ thống.

The screenshot shows a web browser window with a light blue header. The header contains a logo on the left, the text 'Available shifts' and 'Contact' in the center, and 'Login' and 'Register' on the right. The main content area has a white background with the word 'Register' in a large, pink, serif font. Below this, there are three input fields labeled 'Email', 'Password', and 'Confirm Password'. Below the 'Confirm Password' field is a checkbox with the text 'I have read the Terms and Conditions.' Below the checkbox are two buttons: a red 'Back' button and a blue 'Register' button. At the bottom of the page, there is a light blue footer with a logo on the left, the text 'VỀ CHÚNG TÔI' and 'LIÊN HỆ' in the center, and a small circular icon on the right.

Hình 4.3 Giao diện đăng ký (Trang 2)

#### 4.3. Giao diện đăng nhập

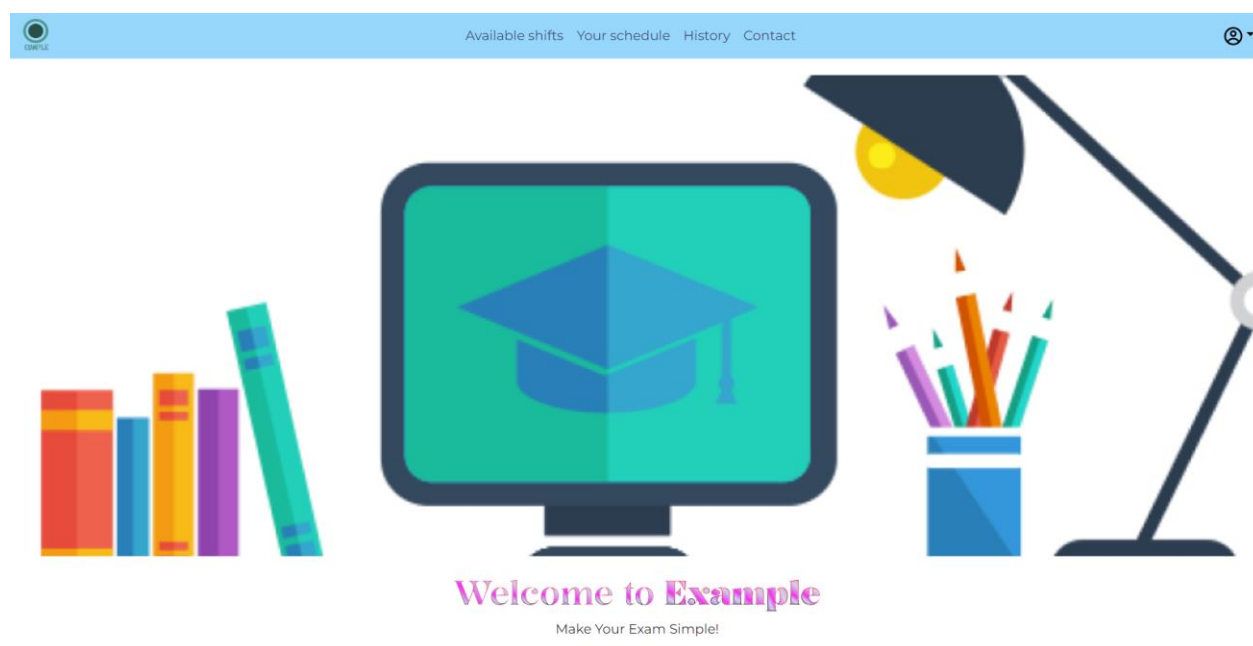
Để đăng nhập vào hệ thống thì người dùng phải nhập đầy đủ các trường Email và Password đã đăng ký trước đó.

The screenshot shows a web browser window with a light blue header. The header contains a logo on the left, the text 'Available shifts' and 'Contact' in the center, and 'Login' and 'Register' on the right. The main content area has a white background with the word 'Login' in a large, pink, serif font. Below this, there are two input fields labeled 'Email' and 'Password'. Below the 'Password' field is a blue 'Login' button. At the bottom of the page, there is a light blue footer with a logo on the left, the text 'VỀ CHÚNG TÔI' and 'LIÊN HỆ' in the center, and a small circular icon on the right.

Hình 4.4 Giao diện đăng nhập

#### 4.4. Giao diện dành cho thí sinh

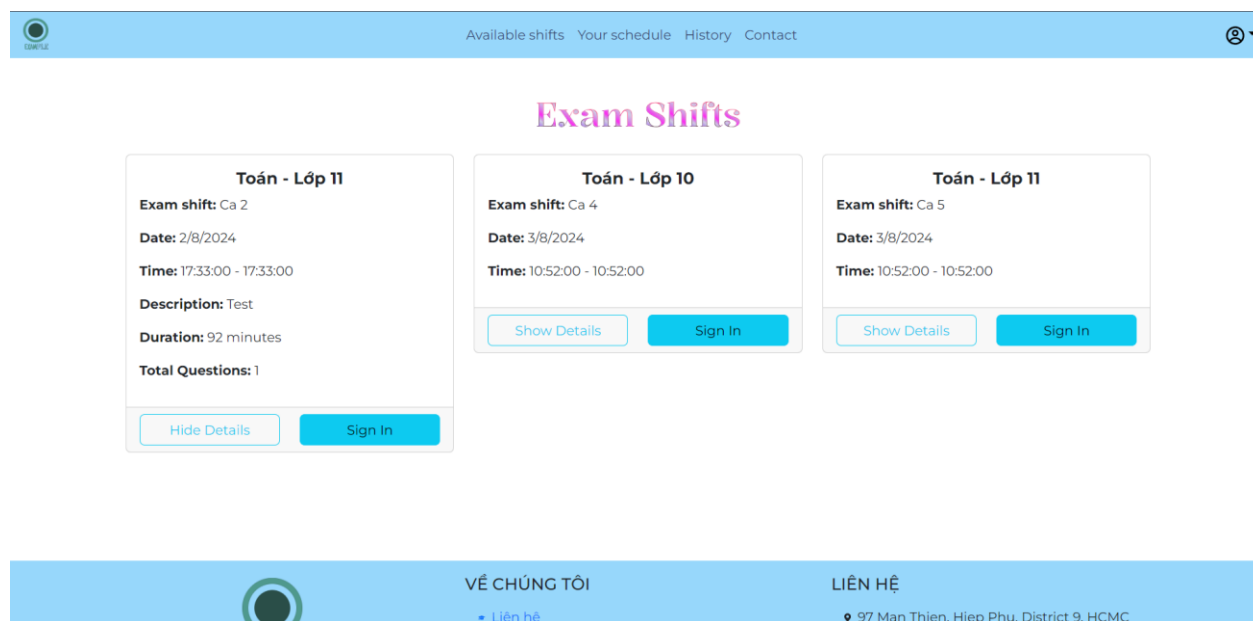
- Thí sinh có thể xem các ca thi sẵn có, đăng ký vào một ca thi, xem các ca thi mà mình đã đăng ký, thi, và xem lịch sử thi.



Hình 4.5 Giao diện trang chủ của thí sinh

#### 4.4.1. Giao diện xem các ca thi đang có

- Thí sinh có thể xem các ca thi đang có trên hệ thống mà thí sinh đó chưa đăng ký thi.
- Thí sinh có thể chọn ca thi bất kì để đăng ký, sau khi đăng ký vào một ca thi thì sẽ hiển thị bên lịch thi.



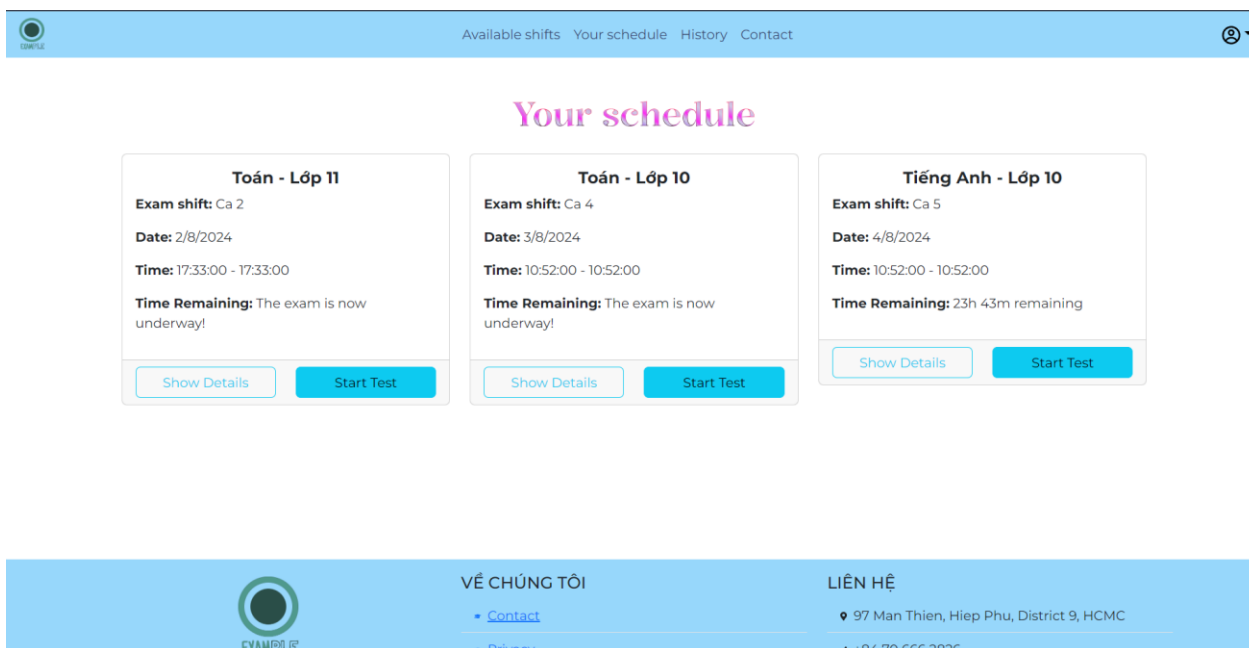
Hình 4.6 Giao diện xem các ca thi đang có

#### 4.4.2. Giao diện xem lịch thi

- Thí sinh có thể xem được lịch thi của bản thân.
- Hệ thống sẽ cho biết thời gian đến khi ca thi của thí sinh đó bắt đầu.



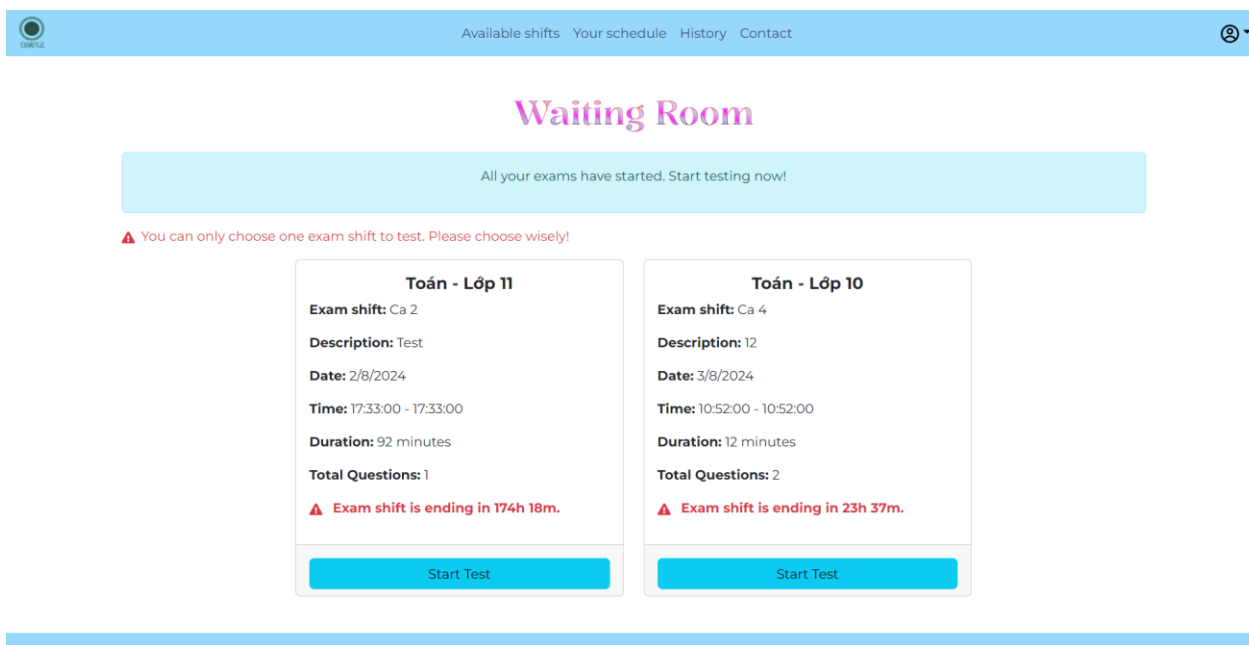
- Thí sinh muốn bắt đầu thi thì có thể ấn “Start test”, nếu ca thi đó đang bắt đầu thì sẽ chuyển sang trang phòng chờ vào thi.



Hình 4.7 Giao diện xem lịch thi

#### 4.4.3. Giao diện xem phòng chờ

- Khi ở phòng chờ, thí sinh có thể chọn “Start Test” để chính thức vào thi.
- Ở phòng chờ sẽ hiện thị các ca thi đã bắt đầu, thí sinh có thể chọn 1 ca thi để bắt đầu thi.
- Giao diện màn hình chờ sẽ có các thông tin cụ thể về bài thi gồm tên ca thi, mô tả ca thi, thời gian làm bài, tổng câu hỏi, thí sinh cần đọc kỹ các thông tin trước khi vào làm bài.



Hình 4.8 Giao diện xem phòng chờ

#### 4.4.4. Giao diện thi

Màn hình làm bài thi bao gồm các phần:

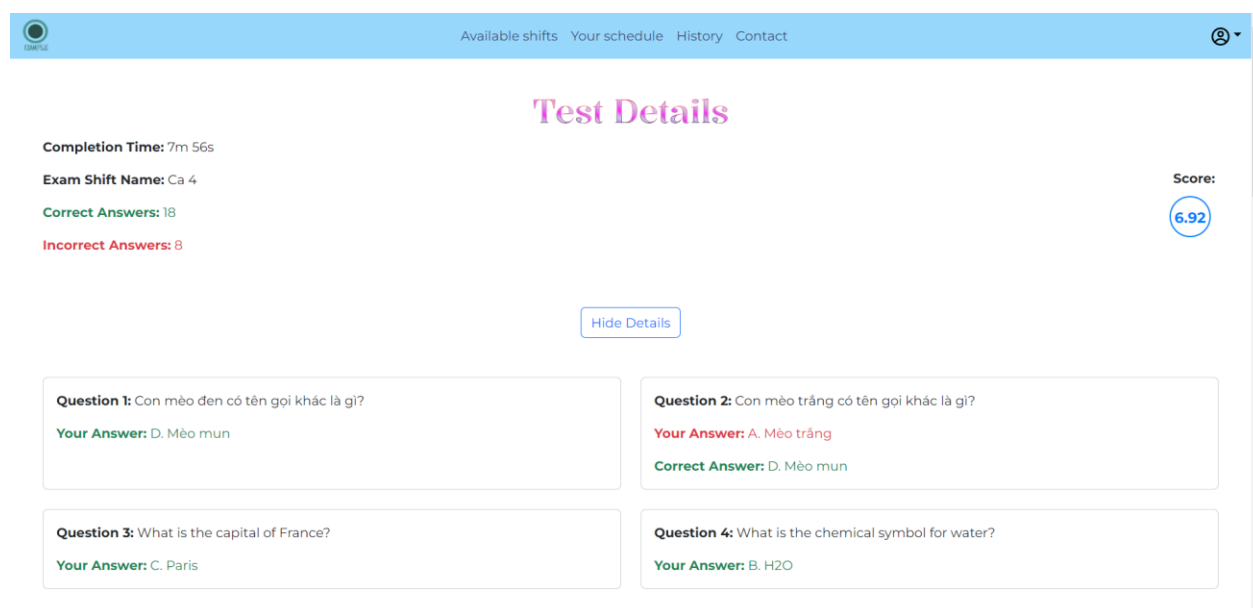
- Câu hỏi thi và các đáp án, thí sinh chỉ có thể chọn một đáp án cho một câu hỏi.
- Đồng hồ thời gian đếm ngược. Đồng hồ đếm ngược từ thời gian làm bài được tạo khi tạo đề thi khi đồng hồ đếm về 00:00:00 thì hệ thống tự động nộp bài và tính điểm
- Các ô vuông thể hiện cho từng câu hỏi. Các ô vuông đỏ thể hiện cho các câu hỏi chưa làm và ô xanh thể hiện câu hỏi đã có đáp án được chọn. Khi người dùng thực hiện chọn đáp án của một câu ô vuông đỏ sẽ đổi màu. Dải ô giúp người thi biết được những câu mình đã làm. Trong trường hợp thí sinh muốn trở về làm lại một câu nào đó, có thể chọn vào ô số tương ứng để trở về câu mình muốn làm.
- Nút “Previous” giúp thí sinh trở về câu hỏi trước đó, nút “Next” để thí sinh đi đến câu hỏi tiếp theo, nếu ở câu cuối cùng, nút “Next” sẽ trở thành nút “Submit”.
- Sau khi đã chắc chắn mình làm bài xong, thí sinh bấm nút “Submit” để nộp bài.

The screenshot shows a web interface for a test. At the top, there's a navigation bar with links: Available shifts, Your schedule, History, and Contact. The main heading is "Test Page" in pink. On the right, it says "Remaining Time: 7m 35s". The central part displays "Question 26" with the text "What is the tallest mountain in the world?". Below this are four radio button options: K2, Kangchenjunga, Lhotse, and Mount Everest (which is selected). At the bottom of the question area are "Previous" and "Submit" buttons. To the right of the question is a grid of green squares numbered 1 to 26. The bottom of the page has a blue footer with a logo, the text "VỀ CHÚNG TÔI" with a "Contact" link, and "LIÊN HỆ" with an address: "97 Man Thien, Hiep Phu, District 9, HCMC".

Hình 4.9 Giao diện thi

#### 4.4.5. Giao diện xem kết quả thi

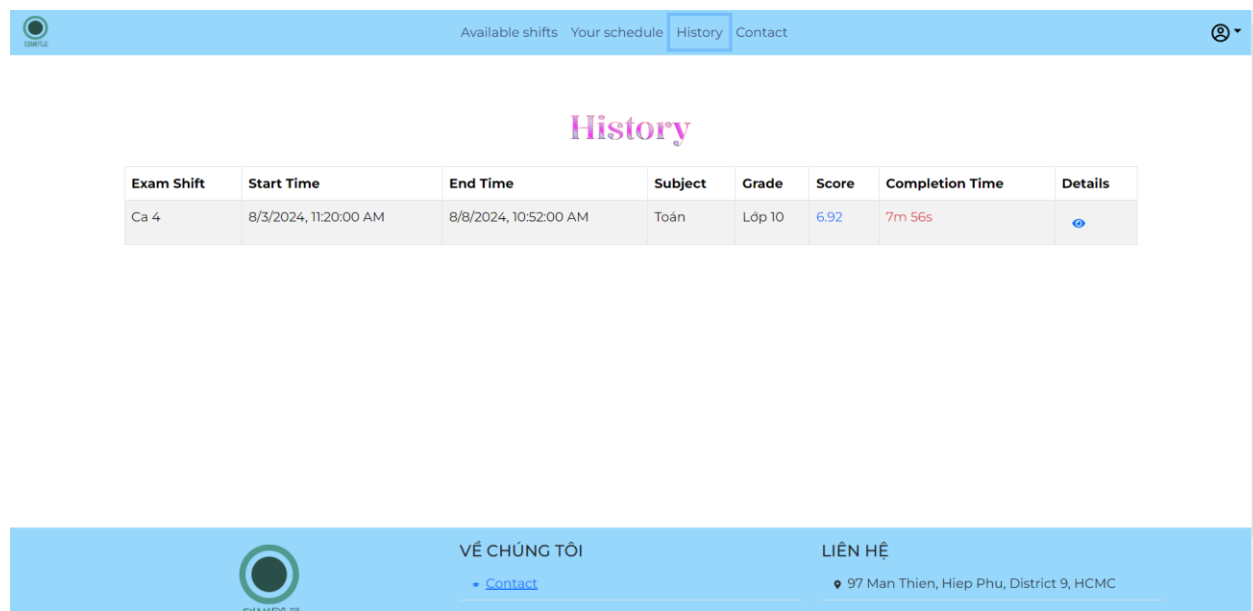
- Sau khi thí sinh xác nhận nộp bài thì hệ thống sẽ tự động chuyển đến trang xem kết quả.
- Trang này sẽ cho biết tên ca thi thí sinh tham gia, thời gian thí sinh hoàn thành bài thi, số câu hỏi làm đúng, số câu hỏi làm sai, và điểm số mà thí sinh đạt được. Ngoài ra, thí sinh có thể xem chi tiết các câu hỏi mình làm.



Hình 4.10 Giao diện xem kết quả thi

#### 4.4.6. Giao diện xem lịch sử làm bài

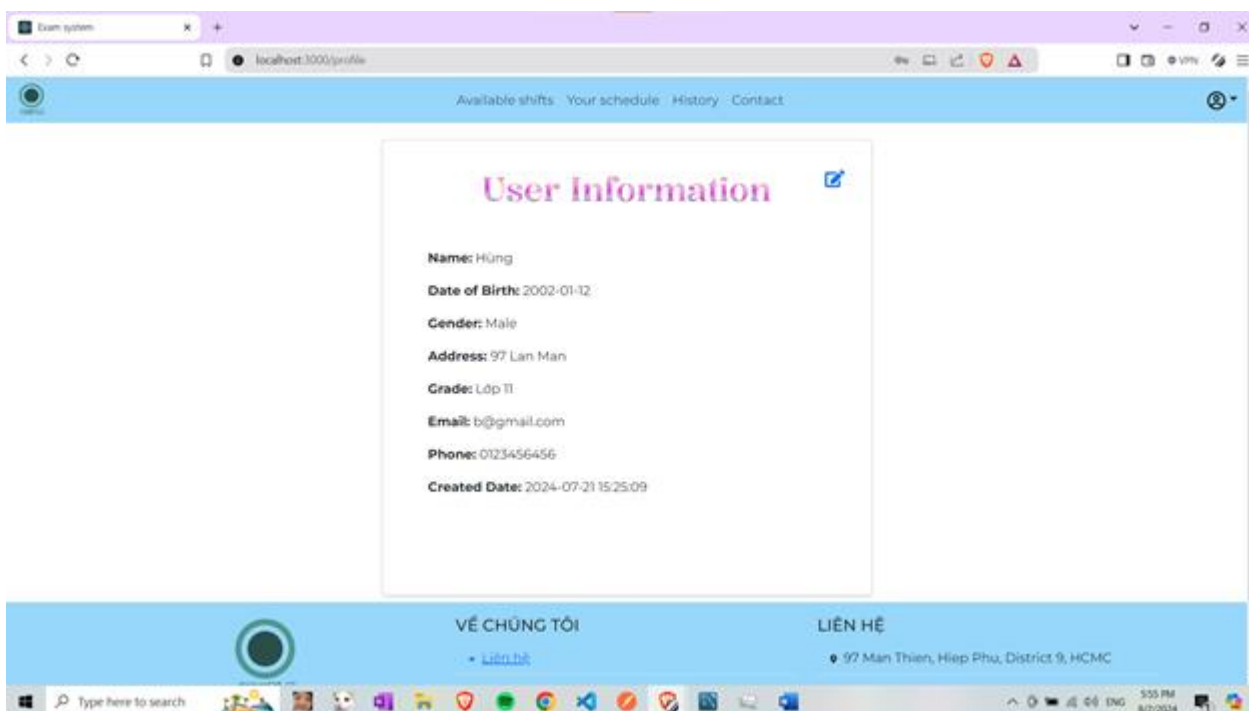
- Ở giao diện xem lịch sử làm bài, thí sinh có thể xem được những bài thi mà mình đã làm, và những bài thi mà mình đã đăng ký.
- Thí sinh có thể xem được thời gian hoàn thành, điểm, ngoài ra thí sinh có thể xem chi tiết bài thi bằng cách nhấn vào biểu tượng đôi mắt ở mục “Details”.



Hình 4.11 Giao diện xem lịch sử làm bài

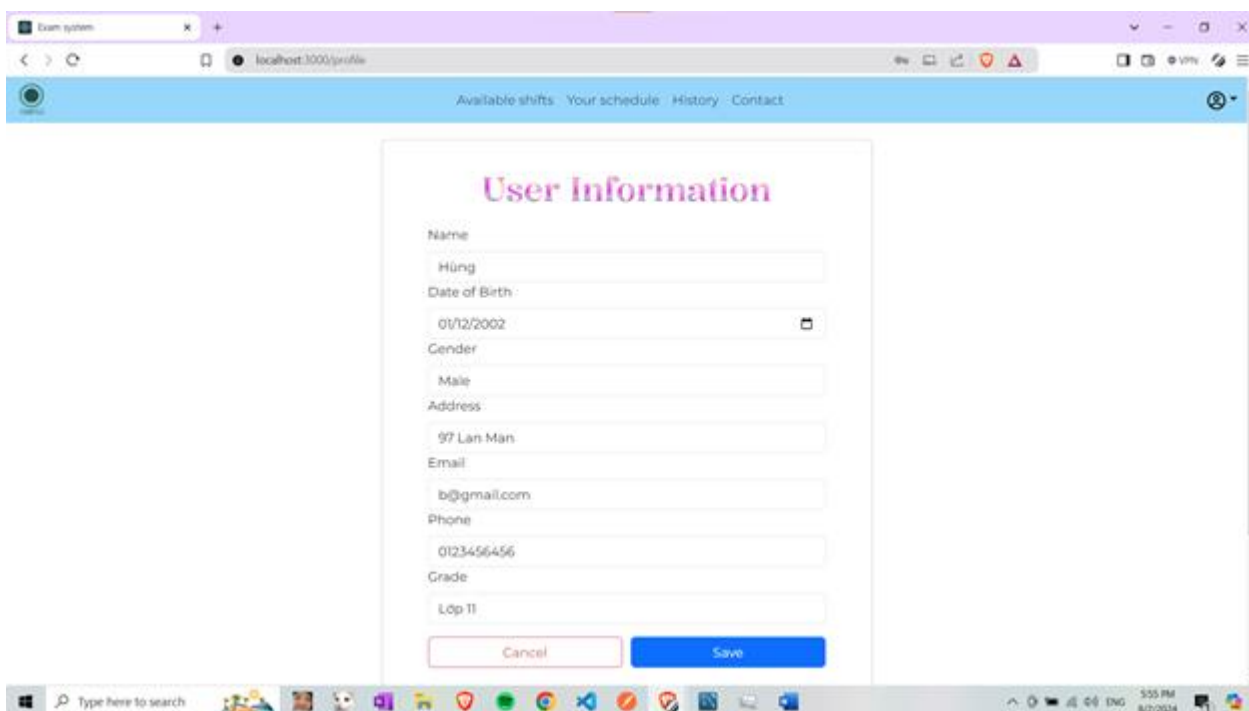
#### 4.4.7. Giao diện xem và chỉnh sửa thông tin cá nhân

- Người dùng có thể xem thông tin cá nhân của mình và chỉnh sửa các thông tin. Để chỉnh sửa thông tin, người dùng cần ấn vào biểu tượng câu bút và tiến hành chỉnh sửa.



Hình 4.12 Giao diện xem thông tin cá nhân

- Giao diện chỉnh sửa thông tin cá nhân, sau khi người dùng chỉnh sửa thông tin thành công, ấn “Save” và thông tin của người dùng sẽ được thay đổi.

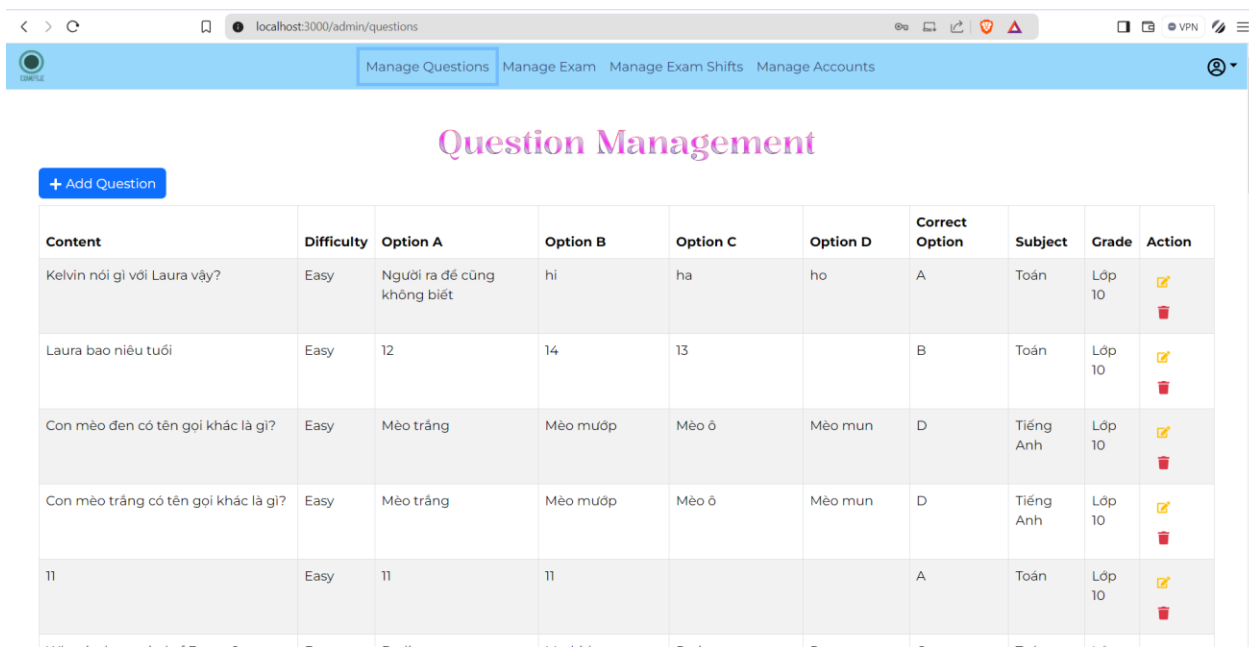


Hình 4.13 Giao diện cập nhật thông tin cá nhân

## 4.5. Giao diện dành cho quản lý

### 4.5.1. Giao diện quản lý bộ câu hỏi

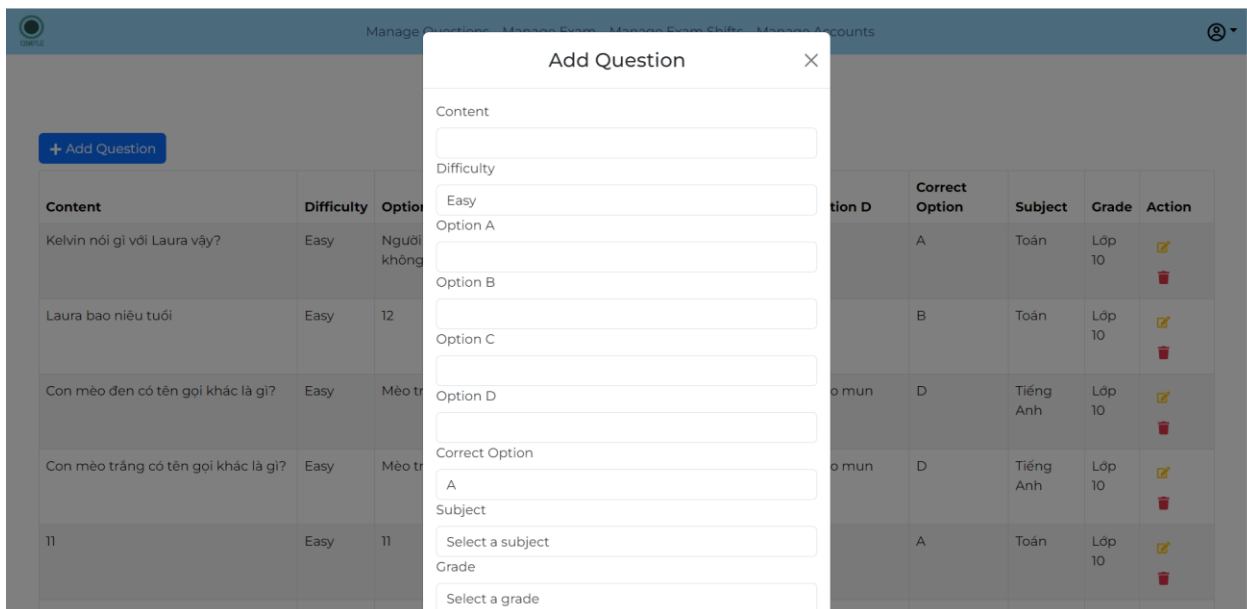
- Giao diện gồm có 3 chức năng chính là thêm câu hỏi, chỉnh sửa câu hỏi và xóa câu hỏi.
- Sau khi thêm thì màn hình sẽ hiển thị chi tiết các thông tin liên quan đến câu hỏi như đề, độ khó, các đáp án, đáp án đúng, tên môn học và tên lớp.



Content	Difficulty	Option A	Option B	Option C	Option D	Correct Option	Subject	Grade	Action
Kelvin nói gì với Laura vậy?	Easy	Người ra đề cũng không biết	hi	ha	ho	A	Toán	Lớp 10	
Laura bao nhiêu tuổi	Easy	12	14	13		B	Toán	Lớp 10	
Con mèo đen có tên gọi khác là gì?	Easy	Mèo trắng	Mèo mập	Mèo ó	Mèo mun	D	Tiếng Anh	Lớp 10	
Con mèo trắng có tên gọi khác là gì?	Easy	Mèo trắng	Mèo mập	Mèo ó	Mèo mun	D	Tiếng Anh	Lớp 10	
11	Easy	11	11			A	Toán	Lớp 10	
What is the capital of France?	Easy	Paris	Madrid	Berlin	Rome	C	Toán	Lớp 10	

Hình 4.14 Giao diện quản lý bộ câu hỏi

- Để thêm câu hỏi mới, người dùng ấn vào nút “Add Question” góc trái màn hình và điền các thông tin chi tiết liên quan đến câu hỏi mà hệ thống yêu cầu. Sau khi điền các thông tin thành công sẽ ấn vào nút “Add Question”.



**Add Question**

Content

Difficulty

Easy

Option A

Option B

Option C

Option D

Correct Option

A

Subject

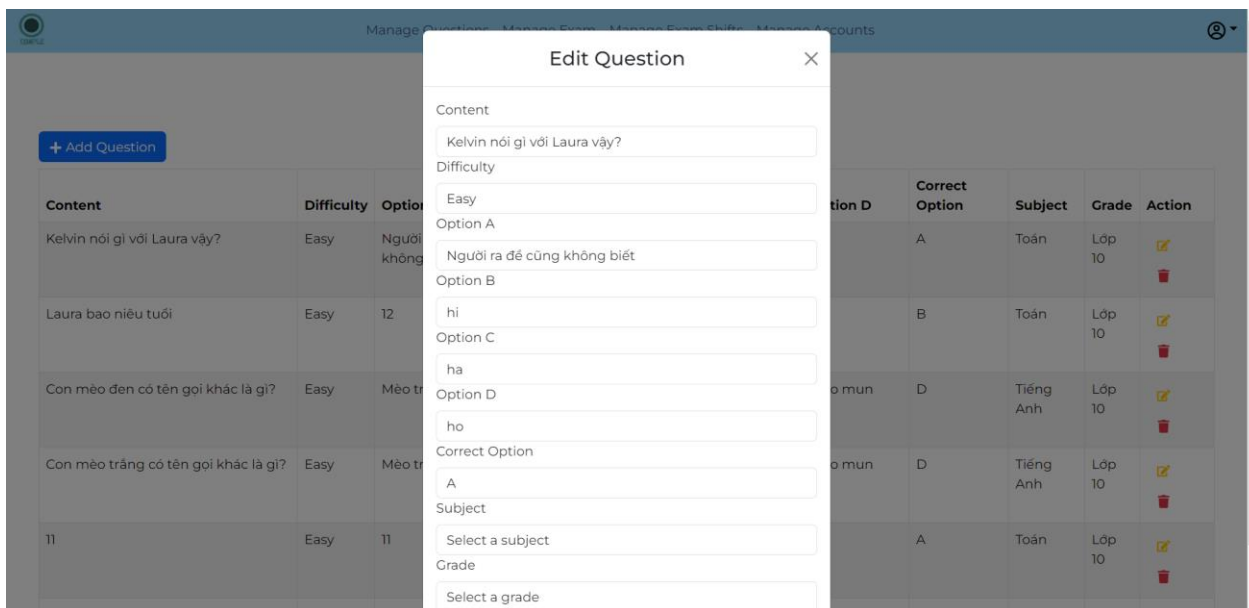
Select a subject

Grade

Select a grade

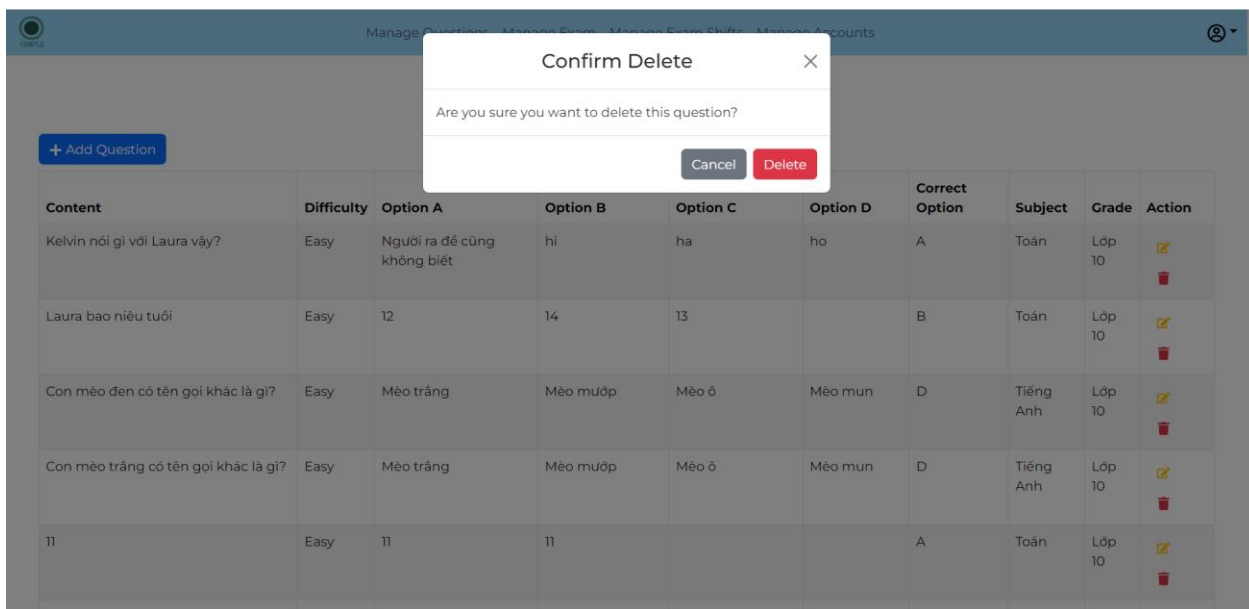
Hình 4.15 Giao diện thêm câu hỏi

- Để chỉnh sửa câu hỏi, người dùng cần ấn vào cây bút ở mục “Action” và tiến hành chỉnh sửa các thông tin mình muốn. Sau khi điền các thông tin thành công sẽ ấn vào nút “Update Question”.



Hình 4.16 Giao diện chỉnh sửa câu hỏi

- Để chỉnh xóa câu hỏi, người dùng cần ấn vào biểu tượng thùng rác ở mục “Action”, lúc này hệ thống sẽ hiện ra một cảnh báo, người dùng cần ấn “Delete” để xác nhận xóa.

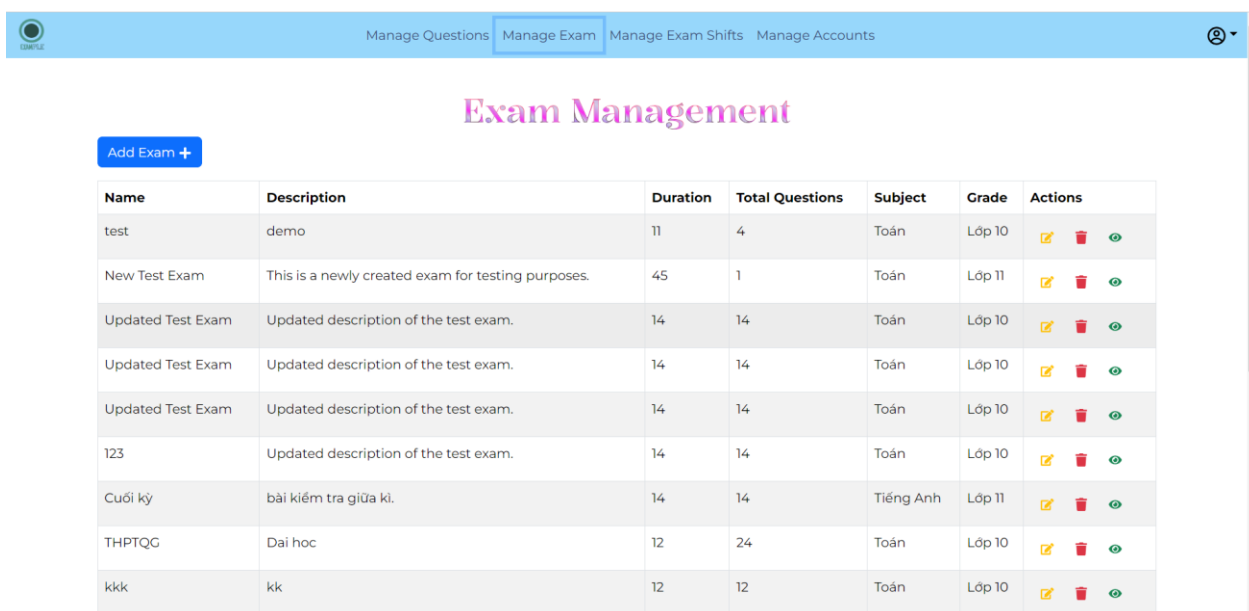


Hình 4.17 Giao diện xóa câu hỏi

#### 4.5.2. Giao diện quản lý đề thi

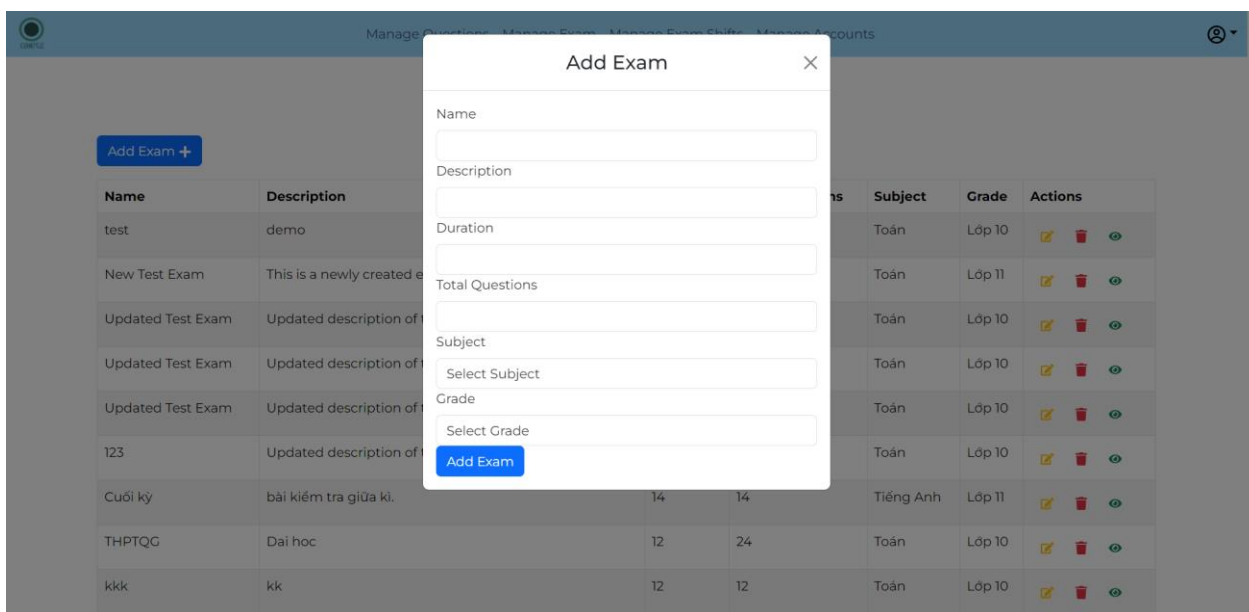
- Giao diện quản lý đề thi giúp người dùng có thể xem các thông tin liên quan đến đề thi bao gồm tên đề thi, mô tả, thời gian làm bài, tổng số câu hỏi có trong đề thi, môn học, lớp.

- Ngoài ra người dùng còn có thể tạo các đề thi, sửa các đề thi, xóa đề thi và xem chi tiết các câu hỏi có trong đề thi.



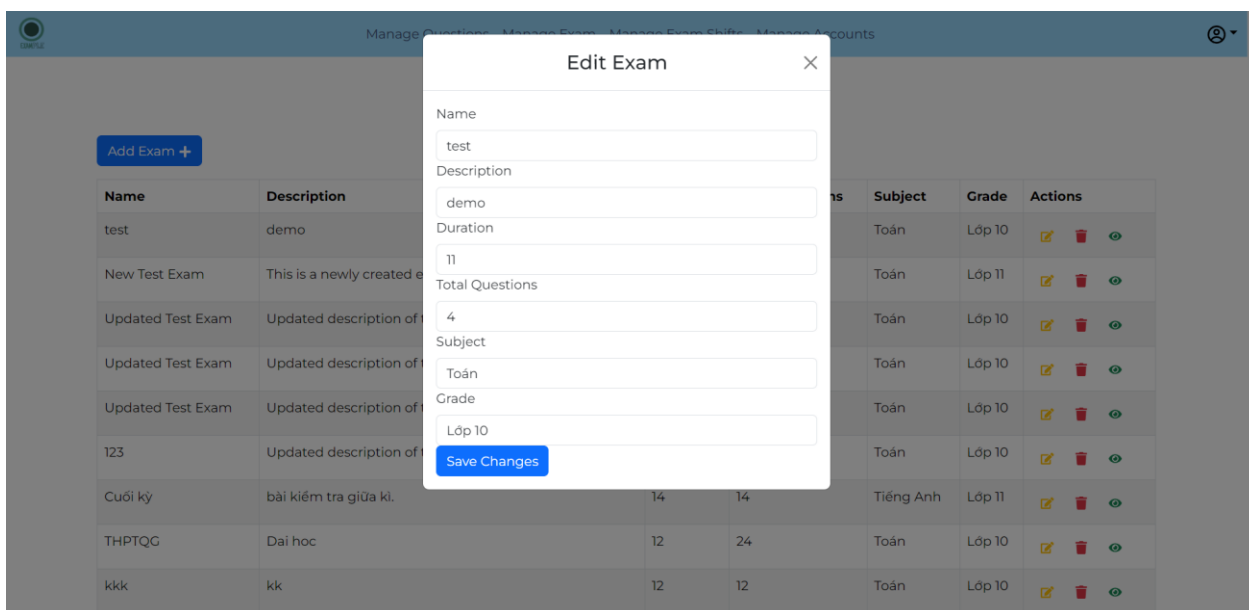
Hình 4.18 Giao diện xem đề thi

- Giao diện thêm đề thi.



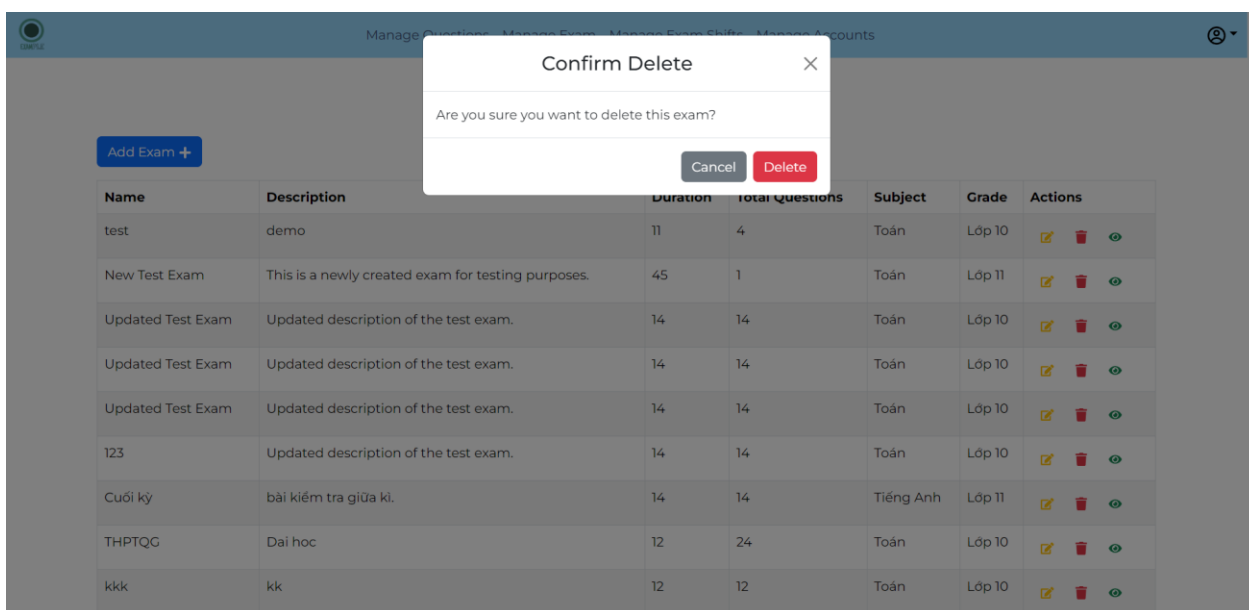
Hình 4.19 Giao diện thêm đề thi

- Giao diện chỉnh sửa đề thi



Hình 4.20 Giao diện chỉnh sửa đề thi

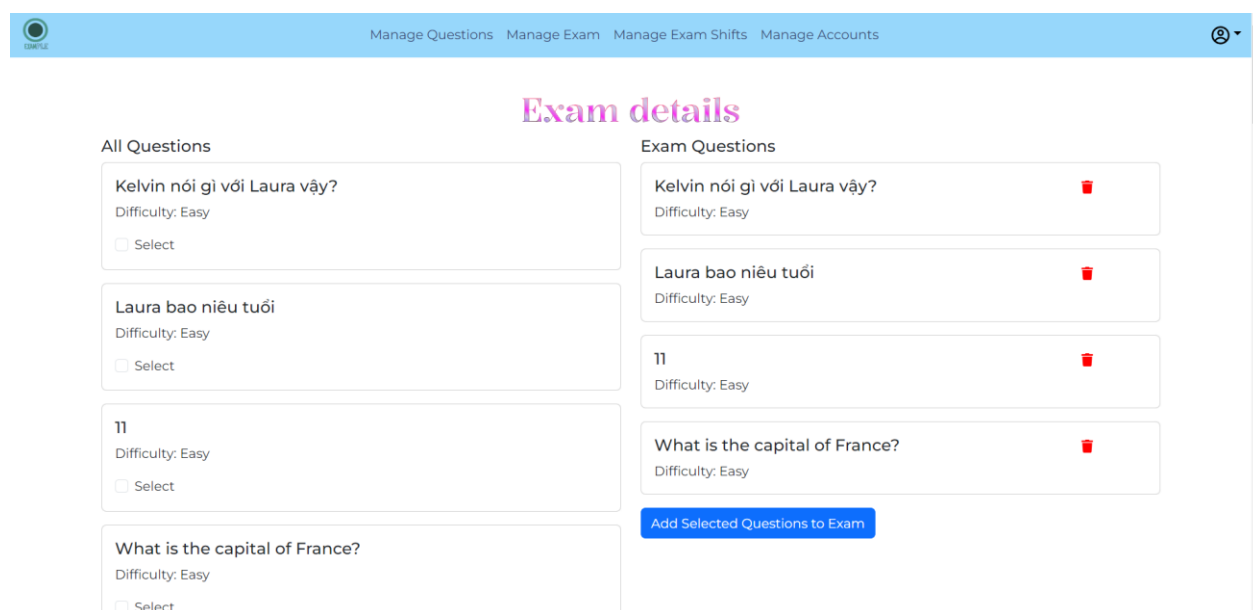
- Giao diện xóa đề thi



Hình 4.21 Giao diện xóa đề thi

- Ngoài ra người dùng có thể xem chi tiết các câu hỏi có trong đề thi bằng cách ấn vào biểu tượng mắt ở mục “Actions”, giao diện như phía dưới sẽ hiện ra.
- Tại trang xem chi tiết đề thi, người dùng có thể thêm các câu hỏi từ bộ câu hỏi vào đề thi bằng cách tích chọn những câu mình muốn thêm và ấn nút “Add Selected Questions to Exam”, và xóa câu hỏi ra khỏi đề thi bằng cách ấn vào biểu tượng thùng rác.

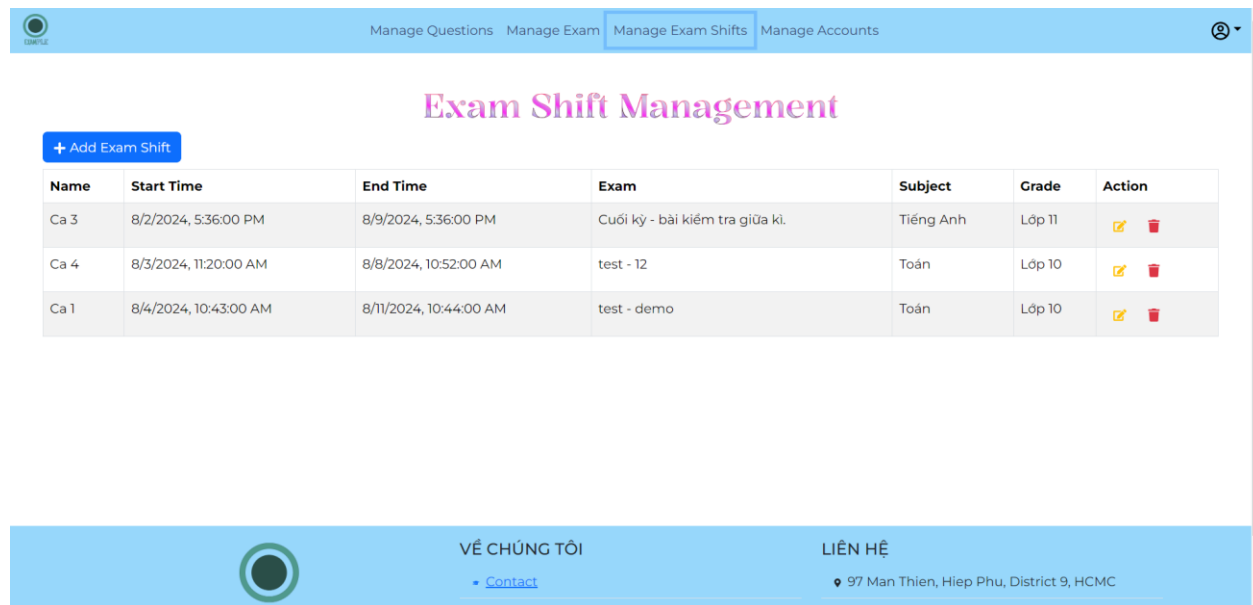




Hình 4.22 Giao diện quản lý chi tiết đề thi

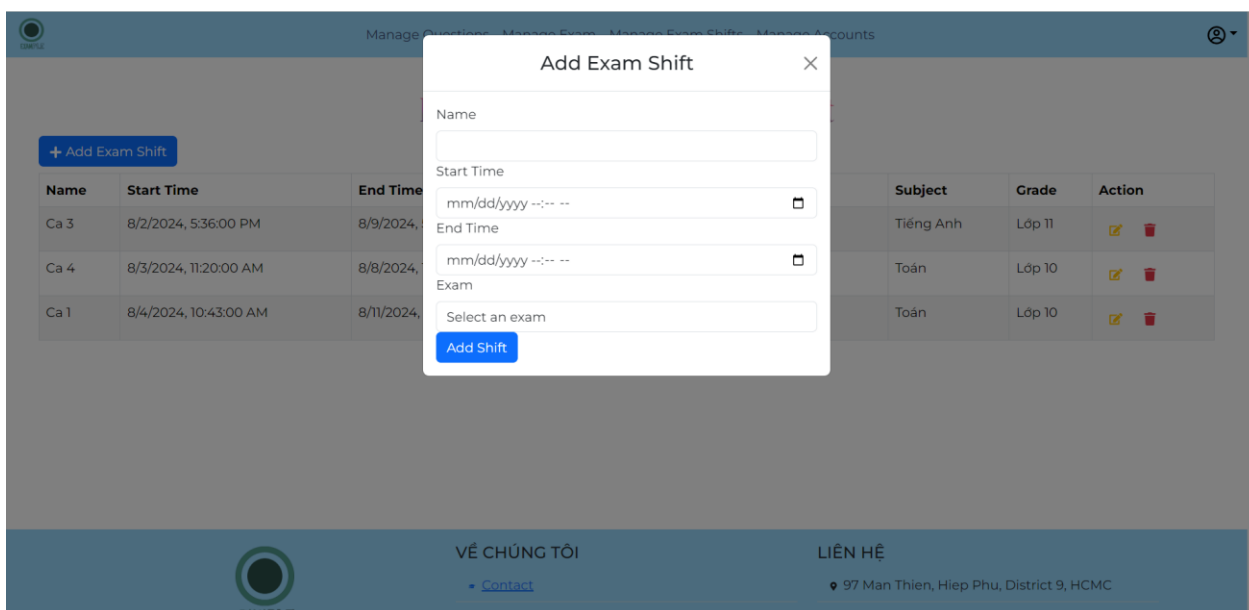
### 4.5.3. Giao diện quản lý ca thi

- Ở trang quản lý ca thi, người dùng có thể xem chi tiết các thông tin liên quan đến ca thi như: tên ca thi, thời gian bắt đầu ca thi, thời gian kết thúc ca thi, tên đề thi của ca thi, môn học và lớp.
- Người dùng còn có thể thêm ca thi, sửa ca thi và xóa ca thi.



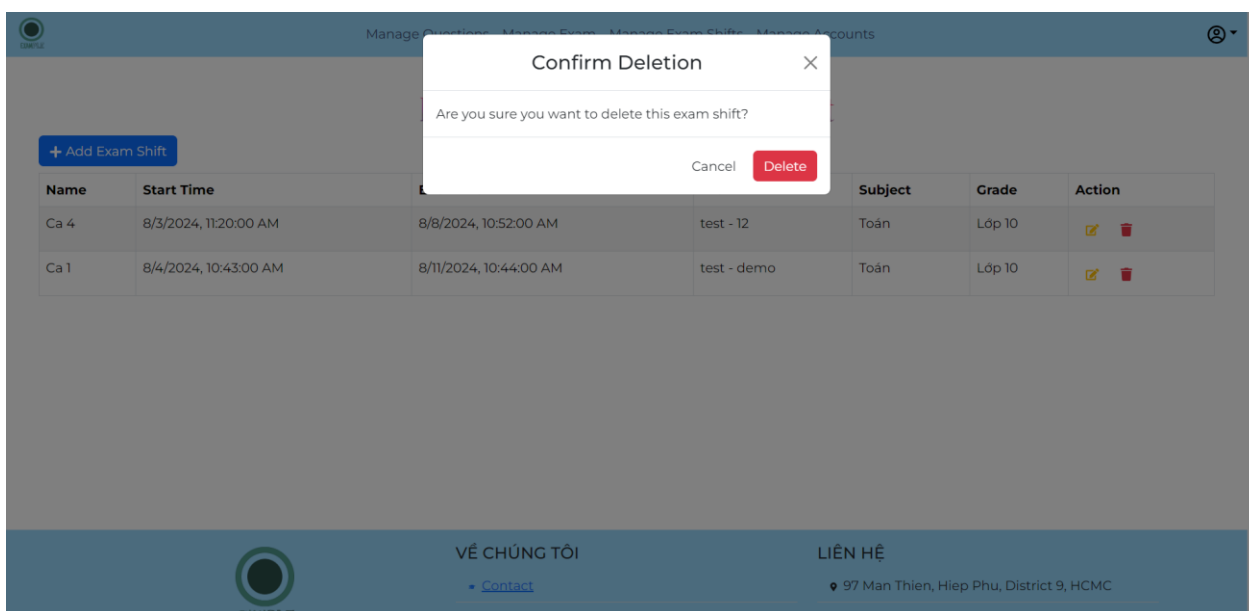
Hình 4.23 Giao diện quản lý các ca thi

- Để thêm ca thi mới, người dùng cần ấn nút “Add Exam Shift”, thêm các thông tin cần thiết và chọn “Add Shift”.



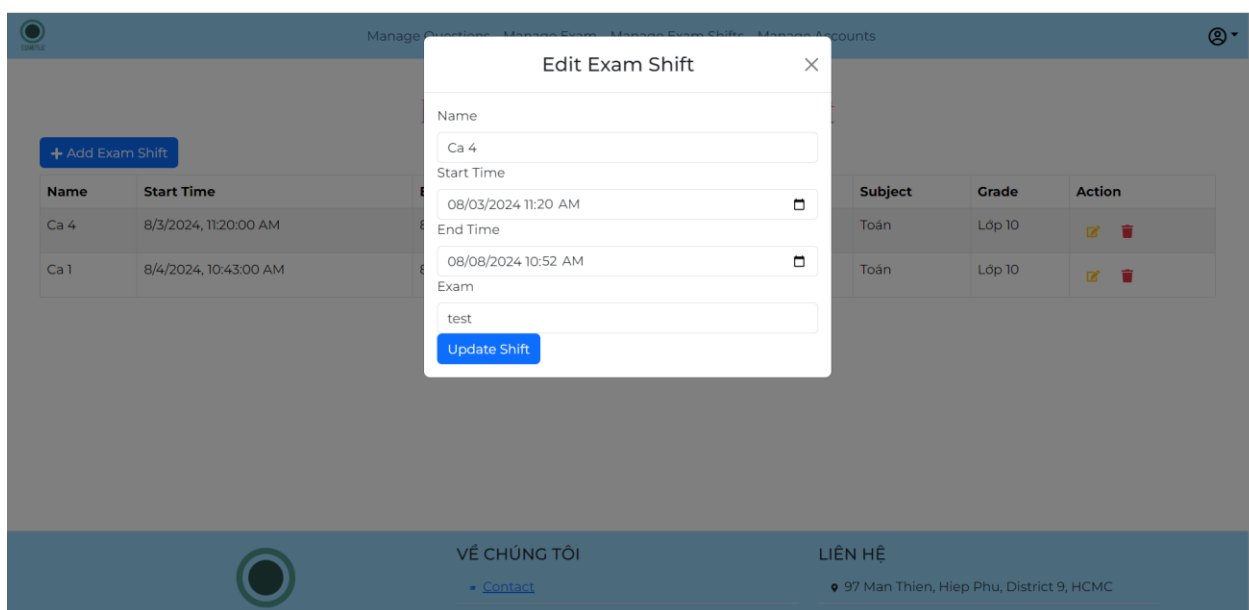
Hình 4.24 Giao diện thêm ca thi

- Để xóa ca thi, người dùng cần ấn biểu tượng thùng rác ngay mục “Actions”, và ấn xác nhận xóa.



Hình 4.25 Giao diện xóa ca thi

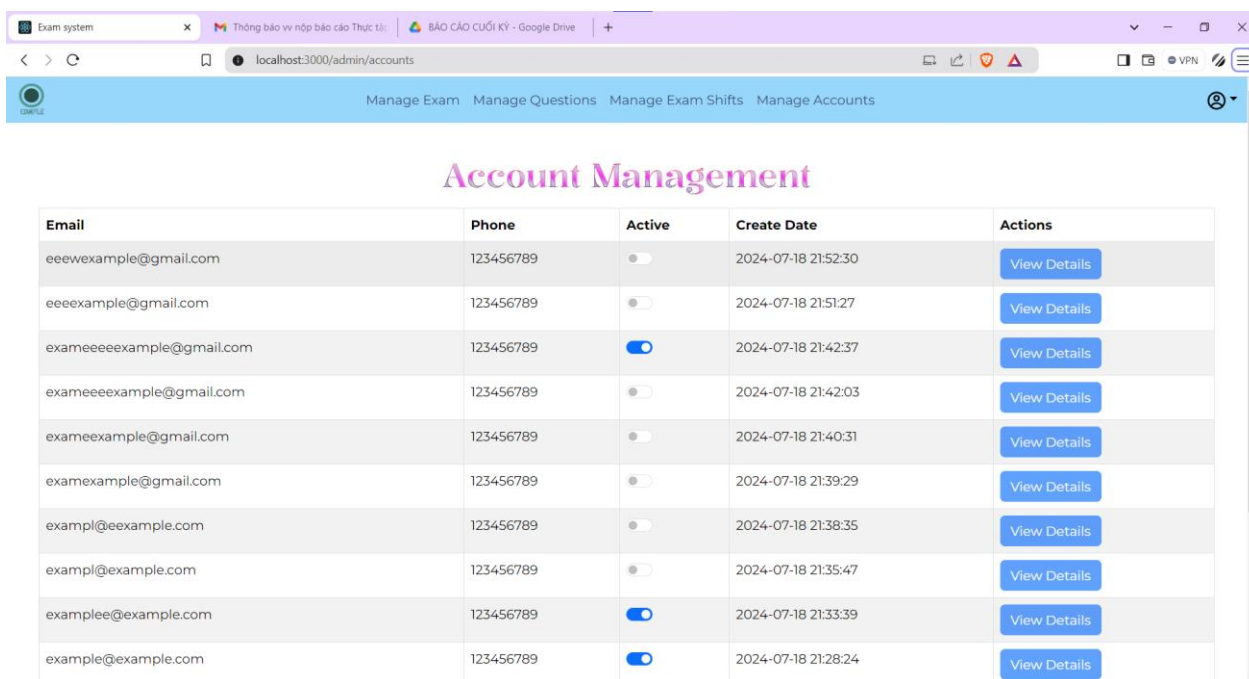
- Để sửa ca thi, người dùng cần ấn biểu tượng cây bút ngay mục “Actions”, thêm các thông tin cần thiết và chọn “Update Shift”.



Hình 4.26 Giao diện chỉnh sửa ca thi

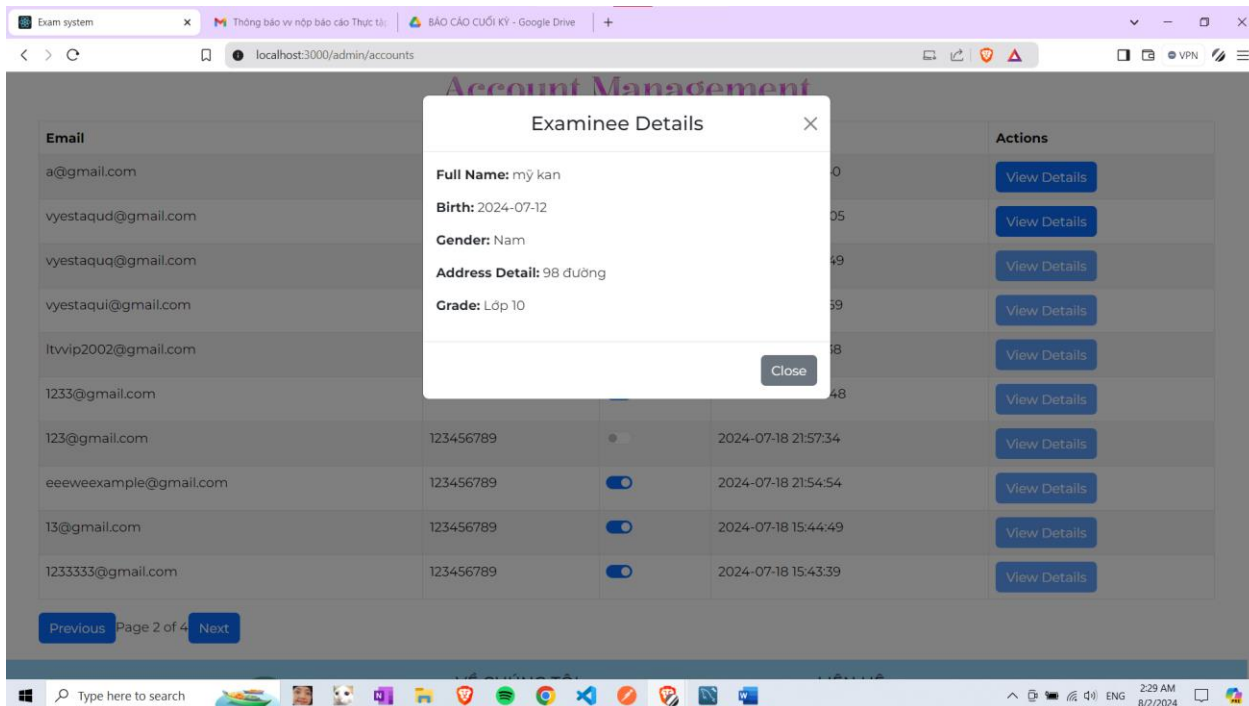
#### 4.5.4. Giao diện quản lý tài khoản

- Ở trang quản lý tài khoản, người dùng có thể xem được các thông tin chi tiết của một tài khoản gồm địa chỉ Email, số điện thoại, ngày tạo tài khoản.
- Có thể khóa tài khoản bằng các ấn vào biểu tượng công tắc ngay mục “Active”.



Hình 4.27 Giao diện quản lý người dùng

- Ngoài ra có thể xem thông tin chi tiết của một thí sinh bao gồm tên, ngày sinh, giới tính, địa chỉ và lớp.



Hình 4.28 Giao diện xem chi tiết thông tin người dùng

## CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

### 5.1. Kết luận

Xây dựng website thi thử trắc nghiệm không phải là một vấn đề mới nhưng nó mang tính thực tế cao, nhất là trong thời buổi 4.0 như hiện nay và cả khi hình thức thi trắc nghiệm đang ngày càng phổ biến. Nhu cầu dữ dụng hệ thống thi thử trắc nghiệm đối với người dùng ngày càng phổ biến em đã thực hiện đề tài “**Phát triển ứng dụng thi trắc nghiệm trên nền tảng Laravel**”. Do sự hạn hẹp về thời gian và trình độ nên website của em trong phạm vi đồ án này chỉ mới đáp ứng được những yêu cầu cơ bản, nếu đưa và thực tế sử dụng thì chắc chắn phải cần nâng cấp theo các định nhất định để đáp ứng nhiều hơn nữa thị hiếu của người dùng ngày càng cao cũng như gia tăng tính bảo mật của trang web để tránh sự tấn công của các hacker.

Trong phạm vi đề tài này, em đã cố gắng để hoàn thành các yêu cầu đặt ra trong phiếu giao nhiệm vụ. Tuy nhiên vẫn còn thiếu sót vì lí do thời gian và trình độ chuyên môn cũng như nghiệp vụ còn hạn chế

### 5.2. Kết quả đạt được

Sau một thời gian thực hiện, đồ án đã đạt được một số kết quả chính như sau:

- Hiểu được quy trình nghiệp vụ trong hệ thống thi trắc nghiệm
- Phân tích và xác định chức năng của hệ thống.
- Phân tích và thiết kế cơ sở dữ liệu lưu trữ thông tin của hệ thống.

Các chức năng đã được xây dựng hoàn chỉnh

❖ Chức năng của quản lý:

- Quản lý câu hỏi bao gồm thêm câu hỏi, xóa câu hỏi, chỉnh sửa câu hỏi.
- Quản lý đề thi bao gồm tạo đề thi, thêm câu hỏi vào đề thi, xóa đề thi, chỉnh sửa đề thi.
- Quản lý ca thi bao gồm tạo ca thi, thêm đề thi vào ca thi, xóa ca thi
- Quản lý thí sinh bao gồm xem thông tin thí sinh và khóa thí sinh.

❖ Chức năng của thí sinh:

- Đăng ký tài khoản, đăng nhập vào hệ thống
- Xem danh sách các ca thi đang có
- Xem lịch thi
- Làm bài thi
- Xem lịch sử danh sách các đề thi đã thi
- Xem và chỉnh sửa thông tin cá nhân

### 5.3. Hướng phát triển

Mặc dù hệ thống website đã hoàn thành chức năng cơ bản cho quản lý và thí sinh, nhưng vẫn còn rất nhiều mặt hạn chế và vấn đề thiếu sót. Như là, mô tả các chức năng chưa đầy đủ, thiếu chính xác, hệ thống chưa thực sự hoàn thiện, ...và ngoài ra để phát triển hệ thống tốt hơn em nghĩ nên có hướng phát triển cho hệ thống như:

❖ Chức năng cho quản lý:

- Hệ thống cần phát triển hoàn thiện hơn về giao diện các chức năng cho quản lý.
- Xử lý về tốc độ xử lý yêu cầu của sản phẩm.

❖ Chức năng cho khách hàng

- Hệ thống cần phát triển hoàn thiện hơn về giao diện cho thí sinh để có thể thu hút thêm nhiều thí sinh mới.

❖ Hệ thống:

- Xây dựng ứng dụng đa nền tảng trên điện thoại di động.
- Cải thiện tiếp tốc độ thực thi của ứng dụng.

---

## TÀI LIỆU THAM KHẢO

- [1] F. Cloud, "MySQL là gì? Cơ chế hoạt động và cách thức cài đặt MySQL," 3 12 2022. [Online]. Available: <https://fptcloud.com/mysql-la-gi/>.
- [2] M. Bảo, "Laravel là gì? Lợi ích của Laravel, hướng dẫn cài đặt Laravel," 3 12 2021. [Online]. Available: <https://wiki.matbao.net/>.
- [3] T. N. T. Thủy, "PHP là gì? Lập trình viên PHP làm công việc gì?," 16 6 2022. [Online]. Available: <https://hocvienagile.com/>.
- [4] T. Q. Hưng, "ReactJS là gì? Tổng quan kiến thức về ReactJS," 19 8 2023. [Online]. Available: <https://200lab.io/>.