

## NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

Vĩnh Long, ngày ..... tháng ..... năm .....  
**Giảng viên hướng dẫn**  
*(Ký tên và ghi rõ họ tên)*

[illegible]

**Thành viên hội đồng**  
(Ký tên và ghi rõ họ tên)

## LỜI CẢM ƠN

Trước hết, em xin gửi lời cảm ơn chân thành đến quý thầy cô và bạn bè đã luôn quan tâm, giúp đỡ em trong suốt quá trình học tập và thực hiện đề tài "Xây dựng Website Quản lý các Kỳ thi của Trung tâm Ngoại ngữ – Tin học".

Đề tài này là một thử thách lớn đối với em, đòi hỏi kiến thức chuyên môn vững vàng về lập trình web fullstack với các công nghệ hiện đại như Next.js, Node.js, MongoDB. Tuy nhiên, nhờ sự hướng dẫn tận tình của thầy/cô, sự động viên của bạn bè, em đã có thể hoàn thành đề tài đúng tiến độ.

Em xin chân thành cảm ơn thầy/cô hướng dẫn đã luôn dành thời gian quý báu để định hướng, giải đáp thắc mắc và góp ý cho em trong suốt quá trình thực hiện đề tài. Thầy/cô đã giúp em hiểu rõ hơn về quy trình phát triển, từ phân tích yêu cầu, thiết kế hệ thống đến triển khai sản phẩm thực tế.

Em cũng xin gửi lời cảm ơn đến các bạn bè trong lớp DA22TTD đã luôn hỗ trợ, chia sẻ kinh nghiệm và động viên em trong suốt quá trình học tập. Sự giúp đỡ của các bạn đã giúp em có thêm động lực để hoàn thành đề tài một cách tốt nhất.

Mặc dù đã cố gắng hết sức, nhưng do kiến thức và kinh nghiệm còn hạn chế nên đề tài không tránh khỏi những thiếu sót. Em rất mong nhận được sự góp ý của quý thầy cô để đề tài được hoàn thiện hơn.

Em xin chân thành cảm ơn!

Trân trọng,

Trần Tấn Hưng Sinh viên lớp DA22TTD - Trường Đại học Trà Vinh

## MỤC LỤC

CHƯƠNG 1: TỔNG QUAN .....	3
1.1. Giới thiệu bài toán .....	3
1.2. Thực trạng và vấn đề cần giải quyết .....	3
1.3. Giải pháp đề xuất .....	3
1.4. Yêu cầu hệ thống .....	3
CHƯƠNG 2: NGHIÊN CỨU LÝ THUYẾT .....	5
2.1 Tổng quan về công nghệ phát triển web .....	5
2.1.1 Khái niệm về website và hệ thống web động .....	5
2.1.2 Kiến trúc Client – Server .....	5
2.1.3 Công nghệ nền tảng .....	6
2.1.4 Công nghệ áp dụng trong đề tài .....	7
2.2 Ngôn ngữ lập trình và Framework sử dụng .....	7
2.2.1 Next.js .....	7
2.2.1.1 Giới thiệu Next.js .....	7
2.2.1.2 Cấu trúc thư mục dự án .....	8
2.2.1.3 Cơ chế Rendering trong Next.js .....	10
2.2.1.4 Routing và Điều hướng .....	11
2.2.1.5 Quản lý tài nguyên tĩnh .....	12
2.3 API .....	13
2.3.1 Khái niệm về API .....	13
2.3.2 Cấu trúc request và response trong API .....	14
CHƯƠNG 3: HIỆN THỰC HÓA NGHIÊN CỨU .....	15
3.1. Đặc tả yêu cầu hệ thống .....	15
3.1.1. Yêu cầu chức năng .....	15
3.1.2. Yêu cầu phi chức năng .....	15
3.2. Phân tích hệ thống .....	16
3.2.1 Sơ đồ Use Case tổng quát .....	16
3.2.2 Đặc tả chi tiết các Use Case chính .....	16
3.3. Thiết kế hệ thống .....	19
3.3.1 Sơ đồ mức 0 (Context Diagram) .....	19
3.3.2 Sơ đồ mức 1 (Decomposition Diagram) .....	20
3.3.3 Sơ đồ hoạt động (Activity Diagram) .....	20
3.4. Thiết kế cơ sở dữ liệu .....	24
3.4.1 Sơ đồ ERD .....	24
3.4.2 Bảng thuộc tính .....	25
3.5. Thiết kế giao diện .....	30

---

CHƯƠNG 4: KẾT QUẢ NGHIÊN CỨU.....	33
4.1. Giao diện chung .....	33
4.2. Giao diện Admin.....	34
4.3. Giao diện Giáo viên .....	36
4.4. Giao diện Học viên .....	37
4.5. Giao diện Responsive .....	38
CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN .....	39
5.1. Kết luận.....	39
DANH MỤC TÀI LIỆU THAM KHẢO .....	41
PHỤ LỤC .....	42

## DANH MỤC HÌNH ẢNH

Hình 2.1 Mô hình Client – Server.....	6
Hình 2.2 Cấu trúc thư mục Next.js 16 .....	8
Hình 3.1 Sơ đồ Use Case hệ thống quản lý Kỳ thi .....	16
Hình 3.2 Sơ đồ mức 0 (Context Diagram) - Hệ thống quản lý kỳ thi .....	19
Hình 3.3 Sơ đồ mức 1 (Decomposition Diagram) - Hệ thống quản lý kỳ thi.....	20
Hình 3.4 Sơ đồ hoạt động - Đăng nhập .....	21
Hình 3.5 Sơ đồ hoạt động - Nhập điểm .....	22
Hình 3.6 Sơ đồ hoạt động - Tra cứu điểm .....	23
Hình 3.7 Sơ đồ ERD - Hệ thống quản lý kỳ thi .....	24
Hình 3.8 Thiết kế wireframe trang đăng nhập .....	30
Hình 3.9 Thiết kế wireframe trang chủ quản trị viên.....	31
Hình 3.10 Thiết kế wireframe trang nhập điểm.....	31
Hình 3.11 Thiết kế wireframe trang tra cứu điểm (Học viên) .....	32
Hình 4.1: Trang chủ giới thiệu hệ thống.....	33
Hình 4.2: Trang đăng nhập .....	33
Hình 4.3: Trang đăng ký .....	34
Hình 4.4: Trang chủ quản trị viên (Dashboard Admin).....	34
Hình 4.5: Quản lý người dùng .....	35
Hình 4.6: Quản lý môn thi .....	35
Hình 4.7: Quản lý kỳ thi .....	36
Hình 4.8: Dashboard Giáo viên .....	36
Hình 4.9: Trang nhập điểm .....	37
Hình 4.10: Dashboard Học viên .....	37
Hình 4.11: Tra cứu điểm thi.....	38
Hình 4.12: Giao diện trên điện thoại.....	38

## **DANH MỤC BẢNG BIỂU**

Bảng 3.1 NGUOI_DUNG (Users).....	26
Bảng 3.2 MON_THI (Subjects).....	26
Bảng 3.3 KY_THI (Exams) .....	28
Bảng 3.4 DIEM_THI (Scores).....	29
Bảng 3.5 LOP_HOC (Classes) .....	30

## TÓM TẮT ĐỒ ÁN CHUYÊN NGÀNH

**Đề tài:** Xây dựng Website Quản lý các Kỳ thi của Trung tâm Ngoại ngữ – Tin học

**Vấn đề nghiên cứu:** Hiện nay, nhiều trung tâm Ngoại ngữ – Tin học vẫn quản lý kỳ thi bằng phương pháp thủ công như lập danh sách thí sinh trên giấy, nhập điểm bằng Excel, tổng hợp kết quả thủ công. Điều này dẫn đến nhiều hạn chế: tốn thời gian xử lý, dễ xảy ra sai sót khi nhập liệu, khó khăn trong việc tra cứu và thống kê dữ liệu. Vì vậy, việc xây dựng một hệ thống website quản lý kỳ thi là cần thiết để số hóa và tự động hóa quy trình này.

**Hướng tiếp cận:** Áp dụng mô hình Agile-Scrum, kiến trúc Client-Server 3 lớp, thiết kế RESTful API và phân quyền theo vai trò (Admin, Giáo viên, Học viên).

**Giải pháp công nghệ:** Frontend sử dụng Next.js, TypeScript, Tailwind CSS. Backend sử dụng Node.js, Express.js, JWT Authentication. Database sử dụng MongoDB Atlas. Triển khai trên Vercel và Render.

**Kết quả đạt được:** Hoàn thành hệ thống bao gồm: landing page, đăng nhập/đăng ký phân quyền, trang chủ cho 3 vai trò, API CRUD đầy đủ. Hệ thống đã được triển khai thành công tại <https://eduscore.vercel.app>.



## MỞ ĐẦU

### 1. Lý do chọn đề tài

Trong thời đại công nghệ 4.0, việc số hóa công tác quản lý trở nên cấp thiết đối với các tổ chức giáo dục. Hiện nay, nhiều trung tâm Ngoại ngữ – Tin học vẫn quản lý kỳ thi bằng phương pháp thủ công: lập danh sách thí sinh trên giấy, nhập điểm bằng Excel, tổng hợp kết quả thủ công. Điều này gây ra nhiều bất cập như tốn thời gian, dễ sai sót và khó khăn trong tra cứu, thống kê.

Xuất phát từ thực tế đó, em chọn đề tài "Xây dựng Website Quản lý các Kỳ thi của Trung tâm Ngoại ngữ – Tin học" nhằm ứng dụng kiến thức lập trình web để giải quyết vấn đề thực tiễn, đồng thời nâng cao kỹ năng phát triển phần mềm fullstack.

### 2. Mục đích nghiên cứu

Xây dựng hệ thống website quản lý kỳ thi trực tuyến, giúp số hóa quy trình quản lý. Hỗ trợ quản lý thông tin kỳ thi, môn thi, học viên và điểm số một cách chính xác, minh bạch. Cung cấp chức năng tra cứu điểm thi nhanh chóng cho học viên và hỗ trợ thống kê, báo cáo và xuất dữ liệu phục vụ công tác quản lý

### 3. Đối tượng nghiên cứu

- Quy trình quản lý kỳ thi tại các trung tâm Ngoại ngữ – Tin học
- Các công nghệ phát triển web hiện đại: Next.js, Node.js, MongoDB
- Phương pháp thiết kế và xây dựng hệ thống phần mềm theo mô hình Client-Server

### 4. Phạm vi nghiên cứu

- Xây dựng hệ thống với 3 vai trò: Admin (quản trị), Giáo viên (nhập điểm), Học viên (tra cứu điểm)
- Các chức năng chính: Quản lý người dùng, quản lý môn thi, quản lý kỳ thi, quản lý điểm, tra cứu kết quả, thống kê báo cáo
- Áp dụng cho các trung tâm Ngoại ngữ – Tin học quy mô vừa và nhỏ

## **CHƯƠNG 1: TỔNG QUAN**

### **1.1. Giới thiệu bài toán**

Các trung tâm Ngoại ngữ – Tin học thường xuyên tổ chức các kỳ thi chứng chỉ cho học viên. Quy trình quản lý kỳ thi bao gồm nhiều công đoạn: đăng ký thí sinh, tổ chức thi, nhập điểm, công bố kết quả và lưu trữ hồ sơ. Khi thực hiện thủ công, quy trình này tốn nhiều nhân lực, thời gian và dễ phát sinh sai sót.

Hệ thống quản lý kỳ thi trực tuyến ra đời nhằm giải quyết các vấn đề trên, giúp tự động hóa quy trình, giảm thiểu sai sót và nâng cao hiệu quả công tác quản lý.

### **1.2. Thực trạng và vấn đề cần giải quyết**

- Dữ liệu phân tán, lưu trữ trên nhiều file Excel riêng lẻ, khó đồng bộ
- Nhập điểm thủ công dễ xảy ra nhầm lẫn, mất thời gian kiểm tra đối chiếu
- Sinh viên/Học viên phải đến trực tiếp hoặc gọi điện để tra cứu kết quả
- Thiếu công cụ thống kê, báo cáo tổng hợp phục vụ đánh giá chất lượng

### **1.3. Giải pháp đề xuất**

Xây dựng hệ thống website quản lý kỳ thi với các đặc điểm:

- Tập trung dữ liệu trên một nền tảng duy nhất, dễ quản lý và sao lưu
- Phân quyền rõ ràng theo vai trò: Admin quản trị hệ thống, Giáo viên nhập điểm, Sinh viên/Học viên tra cứu kết quả
- Giao diện thân thiện, hỗ trợ đa thiết bị (máy tính, điện thoại, máy tính bảng)
- Hỗ trợ nhập điểm nhanh và xuất báo cáo tự động

### **1.4. Yêu cầu hệ thống**

Yêu cầu chức năng:

- Đăng nhập, đăng ký, phân quyền người dùng
- Quản lý môn thi, kỳ thi, học viên (thêm, sửa, xóa, tìm kiếm)
- Nhập điểm và tra cứu kết quả thi
- Thống kê, báo cáo và xuất dữ liệu

Yêu cầu phi chức năng:

- Bảo mật: mã hóa mật khẩu, xác thực JWT
- Hiệu năng: tải trang nhanh, xử lý đồng thời nhiều người dùng
- Khả dụng: hoạt động ổn định 24/7, triển khai trên cloud

## CHƯƠNG 2: NGHIÊN CỨU LÝ THUYẾT

Trình bày cơ sở lý thuyết, lý luận, giả thiết khoa học và phương pháp nghiên cứu đã được sử dụng trong đề án.

### 2.1 Tổng quan về công nghệ phát triển web

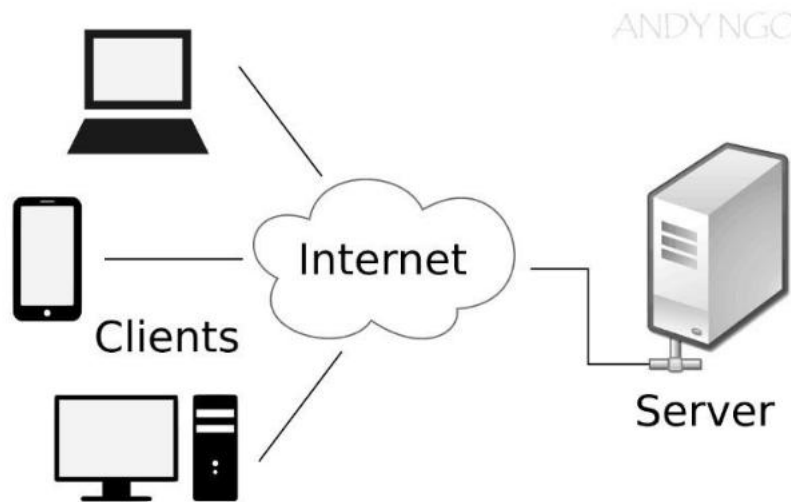
#### 2.1.1 Khái niệm về website và hệ thống web động

Website là tập hợp các trang thông tin gồm văn bản, hình ảnh, video, liên kết,... được hiển thị qua trình duyệt thông qua kết nối Internet. Theo cấu trúc và chức năng, website được chia thành hai loại chủ yếu: website tĩnh và website động. Website động là nền tảng ứng dụng web phổ biến hiện nay với khả năng tương tác với người dùng, cho phép cập nhật, thay đổi nội dung liên tục thông qua cơ sở dữ liệu hoặc hệ thống quản trị. Website động đóng vai trò trọng yếu trong quản lý hoạt động, giới thiệu sản phẩm, giao dịch trực tuyến và cung cấp dịch vụ trực tiếp đến người dùng.

#### 2.1.2 Kiến trúc Client – Server

Kiến trúc Client – Server là mô hình nền tảng của hầu hết các hệ thống web hiện đại. Trong đó, Client thường là trình duyệt web người dùng đảm nhiệm vai trò gửi yêu cầu về tài nguyên hoặc dịch vụ tới Server. Máy chủ xử lý các yêu cầu, truy xuất dữ liệu từ hệ quản trị cơ sở dữ liệu, sau đó trả về kết quả cho client. Kiến trúc này thường triển khai dưới dạng mô hình 3 lớp:

- Lớp trình bày: giao diện người dùng.
- Lớp logic nghiệp vụ: xử lý dữ liệu và hoạt động hệ thống.
- Lớp dữ liệu: truy xuất, lưu trữ dữ liệu.



*Hình 2.1 Mô hình Client – Server*

Giao thức trao đổi phổ biến nhất hiện nay là HTTP hoặc HTTPS, cho phép truyền tải thông tin bảo mật và ổn định.

### **2.1.3 Công nghệ nền tảng**

Công nghệ nền tảng trong phát triển frontend web hiện đại bao gồm nhiều thành phần liên kết chặt chẽ để tạo thành giao diện và chức năng động cho người dùng. HTML là ngôn ngữ đánh dấu tiêu chuẩn, đóng vai trò xác định cấu trúc nội dung của trang web. Bên cạnh đó, CSS thực hiện nhiệm vụ định dạng, thiết kế giao diện, quản lý layout và các hiệu ứng trực quan. JavaScript là ngôn ngữ lập trình phía client giúp bổ sung khả năng tương tác, xử lý logic, kết nối dữ liệu không đồng bộ và quản lý các tác vụ thời gian thực.

Các framework hiện đại như React, Vue.js, Angular cho phép xây dựng ứng dụng web theo kiến trúc component hóa, giúp quản trị mã nguồn tối ưu, dễ dàng bảo trì và mở rộng. Ngoài ra, sự kết hợp với các thư viện của JavaScript hoặc CSS như Bootstrap, Tailwind CSS, Material UI,... nâng cao tốc độ hoàn thiện giao diện cũng như khả năng tương thích trên nhiều loại thiết bị.

Quản lý trạng thái ứng dụng và dữ liệu càng trở nên quan trọng đối với các hệ thống quy mô lớn, do đó các giải pháp quản lý như Redux, Context API, hay các thư viện truy vấn dữ liệu như TanStack Query được ưu tiên cao. Với các ứng dụng yêu cầu tính đồng bộ theo thời gian thực thì các công nghệ như WebSocket hoặc

Socket.IO thường được lựa chọn. Ngoài ra, còn có các thư viện như Chart.js, D3.js giúp hỗ trợ hiển thị và phân tích dữ liệu trực quan.

Việc ứng dụng các công nghệ nền tảng này giúp sản phẩm không những đạt được chất lượng về mặt giao diện, hiệu năng, mà còn đảm bảo khả năng bảo trì, mở rộng lâu dài, đáp ứng tiêu chuẩn công nghiệp hiện hành.

#### **2.1.4 Công nghệ áp dụng trong đề tài**

##### **Frontend:**

- Next.js 16 (React framework) với App Router
- TypeScript để đảm bảo type-safe
- Tailwind CSS v4 cho styling
- Framer Motion cho hiệu ứng chuyển động

##### **Backend:**

- Node.js + Express.js xây dựng RESTful API
- JWT Authentication để xác thực người dùng
- bcrypt mã hóa mật khẩu

##### **Database:**

- MongoDB Atlas (Cloud Database)
- Mongoose ODM để thao tác dữ liệu

##### **Deployment:**

- Vercel (Frontend)
- Render (Backend)
- GitHub cho quản lý mã nguồn

## **2.2 Ngôn ngữ lập trình và Framework sử dụng**

### **2.2.1 Next.js**

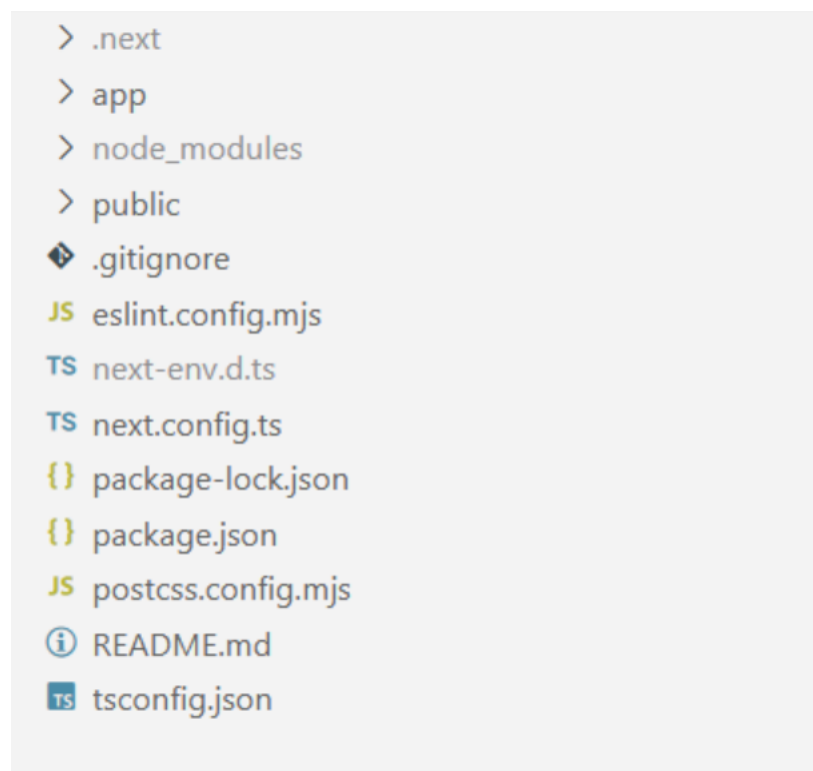
#### **2.2.1.1 Giới thiệu Next.js**

Next.js là một framework mã nguồn mở được xây dựng trên nền tảng React, hỗ trợ phát triển các ứng dụng web full-stack hiện đại. Với Next.js, lập trình viên sử dụng các thành phần React để xây dựng giao diện người dùng, đồng thời tận dụng những tính năng tối ưu hóa và các tiện ích nâng cao mà framework cung cấp.

Next.js tự động cấu hình các công cụ nền tảng thấp như trình biên dịch, giúp đơn giản hóa quy trình thiết lập và phát triển dự án, tạo điều kiện cho lập trình viên tập trung chủ yếu vào việc xây dựng sản phẩm và triển khai tính năng mới một cách nhanh chóng. Framework này phù hợp cho cả cá nhân phát triển lẫn các nhóm lớn nhờ khả năng tạo ra các ứng dụng giàu tính tương tác, động và có hiệu suất cao.

Next.js nổi bật với các chức năng như rendering linh hoạt (server-side, static, incremental static, client-side), routing dựa trên cấu trúc file, tối ưu hóa SEO, hỗ trợ dữ liệu thời gian thực và tích hợp tốt với hệ sinh thái React. Nhờ những đặc điểm này, Next.js ngày càng được lựa chọn rộng rãi để xây dựng ứng dụng web hiệu quả, mở rộng và thân thiện với người dùng hiện đại.

### ***2.2.1.2 Cấu trúc thư mục dự án***



*Hình 2.2 Cấu trúc thư mục Next.js 16*

Dự án Next.js hiện đại có cách sắp xếp thư mục rõ ràng, giúp dễ quản lý code, mở rộng và bảo trì. Cấu trúc này dựa trên quy ước sẵn có, tập trung vào routing và tách biệt chức năng.

### **Thư mục app/**

Đây là thư mục chính cho hệ thống định tuyến mới. Mỗi thư mục con tương ứng với một đường dẫn URL. Các file đặc biệt bao gồm:

- page.js hoặc page.tsx: Nội dung chính của trang.
- layout.js hoặc layout.tsx: Bố cục chung cho route hoặc phần con.
- loading.js: Màn hình tải khi dữ liệu đang xử lý.
- error.js: Xử lý lỗi tùy chỉnh cho route.
- route.js hoặc route.ts: Tạo API endpoint ngay trong app.

### **Thư mục public/**

Nơi lưu tài nguyên tĩnh như hình ảnh, icon, font chữ. File ở đây truy cập trực tiếp qua URL gốc, ví dụ /image.png.

### **Thư mục node\_modules/**

Chứa các thư viện và gói cài đặt qua npm, pnpm hoặc yarn. Không chỉnh sửa thủ công.

### **Thư mục components/**

Lưu các phần giao diện tái sử dụng, như nút bấm, form hoặc header. Giúp tránh lặp code.

### **Thư mục styles/**

Chứa file CSS, SCSS hoặc kiểu dáng toàn cục. Có thể dùng module cho từng phần tử riêng.

### **Các file cấu hình ở gốc dự án**

- next.config.js: Thiết lập tùy chỉnh cho Next.js, như môi trường hoặc plugin.
- package.json: Quản lý gói phụ thuộc và lệnh chạy (build, start).
- .env, .env.local, .env.production, .env.development: Lưu biến môi trường như API key.
- tsconfig.json hoặc jsconfig.json: Cấu hình TypeScript hoặc JavaScript.
- .gitignore: Bỏ qua file không cần theo dõi (như node\_modules).
- Các file khác: next-env.d.ts, README.md.



## Kết luận

Cấu trúc này giúp dự án Next.js dễ mở rộng, kiểm thử và triển khai, phù hợp cho ứng dụng web từ nhỏ đến lớn.

### 2.2.1.3 Cơ chế *Rendering* trong *Next.js*

Next.js là framework dựa trên React, giúp tối ưu tốc độ và trải nghiệm người dùng bằng cách kết hợp nhiều cách hiển thị nội dung. Các cách này cân bằng giữa tải nhanh, dữ liệu mới và khả năng mở rộng dự án.

#### 1. Server-side Rendering (SSR)

SSR tạo nội dung trang trên máy chủ mỗi khi người dùng yêu cầu. Máy chủ lấy dữ liệu mới nhất, tạo HTML hoàn chỉnh và gửi về trình duyệt. Cách này tốt cho trang cần cập nhật thường xuyên, cải thiện tìm kiếm vì nội dung sẵn sàng ngay lập tức.

#### 2. Static Site Generation (SSG)

SSG tạo toàn bộ trang thành file HTML tĩnh ngay lúc xây dựng dự án. File này phục vụ trực tiếp khi người dùng truy cập, giúp tải rất nhanh. Phù hợp cho trang ít thay đổi như blog hoặc trang giới thiệu, nhưng không lý tưởng cho dữ liệu động.

#### 3. Incremental Static Regeneration (ISR)

ISR kết hợp SSG và cập nhật, tái tạo trang tĩnh theo thời gian định sẵn hoặc khi dữ liệu thay đổi. Nó giữ tốc độ nhanh như tĩnh nhưng vẫn cho phép nội dung mới mà không cần xây dựng lại toàn bộ.

#### 4. Client-side Rendering (CSR)

CSR tạo nội dung và tương tác trên trình duyệt sau khi tải trang ban đầu. JavaScript xử lý dữ liệu và cập nhật giao diện. Tốt cho phần cần nhiều hành động người dùng như ứng dụng tương tác, nhưng có thể chậm hơn cho SEO.

#### 5. Partial Prerendering (PPR) và Streaming

PPR tạo trước phần giao diện tĩnh trên máy chủ, trong khi phần động tải dần qua React Suspense. Streaming gửi dữ liệu từng phần, cập nhật trang liên tục mà không chờ toàn bộ, giúp trải nghiệm mượt mà hơn.

## Kết luận

Next.js cho phép kết hợp các cách hiển thị này để tạo ứng dụng nhanh, an toàn, dễ mở rộng và thân thiện với người dùng.

#### **2.2.1.4 Routing và Điều hướng**

Next.js dùng cách định tuyến dựa trên cấu trúc thư mục và file, giúp dễ dàng tạo đường dẫn URL và sắp xếp chức năng ứng dụng một cách rõ ràng. Cách này đơn giản cho dự án nhỏ, nhưng cũng linh hoạt cho dự án lớn, hỗ trợ cả đường dẫn cố định và động.

##### **Định tuyến dựa trên file và thư mục**

Mỗi thư mục hoặc file trong thư mục `app/` sẽ tương ứng với một phần của URL.

File `page.js` hoặc `page.tsx` đại diện cho một trang tại đường dẫn cụ thể.

Ví dụ: File `app/dashboard/settings/page.js` sẽ tạo đường dẫn `/dashboard/settings`. Thư mục lồng nhau giúp tạo đường dẫn phức tạp hơn.

##### **Định tuyến động**

Để tạo đường dẫn thay đổi theo dữ liệu, dùng dấu ngoặc vuông `[]`.

Ví dụ: Thư mục `app/blog/[slug]/page.js` sẽ tạo đường dẫn như `/blog/abc` hoặc `/blog/xyz`.

Trong code trang, lấy giá trị động qua props `params` để sử dụng.

##### **Bố cục lồng nhau**

Next.js cho phép tạo bố cục lồng nhau để chia sẻ giao diện chung cho toàn bộ hoặc một phần ứng dụng.

File `layout.js` trong mỗi thư mục giữ cho giao diện nhất quán và dễ tái sử dụng, tránh lặp code.

##### **Điều hướng giữa các trang**

Sử dụng thành phần `<Link>` của Next.js để chuyển trang mượt mà, tự động tải trước dữ liệu để tránh tải lại toàn bộ.

Với trường hợp phức tạp, dùng hook `useRouter()` để đẩy, thay thế, quay lại trang, hoặc tùy chỉnh hành vi.

- Next.js còn tự động tải trước liên kết hiển thị trên màn hình để tăng tốc độ.
- Tham số tìm kiếm và props đường dẫn
- Trong trang phía máy chủ, lấy tham số URL qua prop searchParams.
- Sử dụng helpers như PageProps hoặc LayoutProps để đảm bảo code an toàn kiểu dữ liệu và dễ quản lý props từ định tuyến.

### **API Routes và xử lý đường dẫn**

Next.js cho phép tạo điểm cuối API ngay trong hệ thống định tuyến bằng file `route.js` hoặc `route.ts`.

Điều này giúp xử lý yêu cầu từ phía máy chủ mà không cần server riêng.

### **Kết luận**

Định tuyến của Next.js giúp sắp xếp code rõ ràng, dễ mở rộng, tối ưu phát triển cho cả trang web tĩnh và ứng dụng động. Nó mang lại trải nghiệm chuyển trang nhanh và liền mạch cho dự án hiện đại.

#### **2.2.1.5 Quản lý tài nguyên tĩnh**

Tài nguyên tĩnh là các file không thay đổi như hình ảnh, font chữ, văn bản hoặc media, giúp xây dựng ứng dụng web hiệu quả. Next.js quản lý chúng qua thư mục `public/` ở gốc dự án, tách biệt rõ ràng với code động để dễ bảo trì và mở rộng.

#### **Cách đặt và truy cập**

Đặt tất cả file tĩnh vào `public/`. Máy chủ phục vụ trực tiếp mà không qua xử lý routing, giúp phản hồi nhanh và tiết kiệm băng thông.

Ví dụ: File `public/assets/img.png` có thể truy cập qua `/assets/img.png` trong code, như `<Image src="/assets/img.png" />`.

Điều này hỗ trợ tích hợp SEO, như `robots.txt` hoặc `favicon.ico`, mà không cần rebuild toàn bộ khi cập nhật.

### **Tối ưu hóa**

File trong public/ không bị rebuild khi triển khai, nên cập nhật độc lập. Next.js đặt header cache mặc định để tránh lưu cache không an toàn, nhưng bạn có thể tùy chỉnh cho hiệu suất tốt hơn.

## Kết luận

Cách quản lý tài nguyên tĩnh của Next.js đảm bảo ổn định, nhanh chóng và dễ mở rộng, nâng cao trải nghiệm người dùng và hiệu năng dự án web.

## 2.3 API

### 2.3.1 Khái niệm về API

API là cơ chế cho phép hai thành phần phần mềm khác nhau giao tiếp và trao đổi dữ liệu thông qua một tập hợp các định nghĩa và giao thức chuẩn. API đóng vai trò như một hợp đồng dịch vụ, định nghĩa rõ cách thức một phần mềm có thể gửi yêu cầu đến một phần mềm hoặc dịch vụ khác, và nhận về phản hồi phù hợp.

Ví dụ điển hình là ứng dụng thời tiết trên điện thoại thông minh, sử dụng API để kết nối với hệ thống dữ liệu dự báo thời tiết, gửi các yêu cầu về vị trí địa lý, ngày giờ và nhận về thông tin dự báo cập nhật theo thời gian thực.

Kiến trúc API hiện đại thường chia thành các loại chính như:

- **API REST:** Giao tiếp qua giao thức HTTP, sử dụng các phương thức GET, POST, PUT, DELETE... Dữ liệu truyền thường là định dạng JSON, nhấn mạnh tính không trạng thái và tương thích tốt với môi trường web, mobile.
- **API WebSocket:** Hỗ trợ giao tiếp hai chiều realtime giữa các ứng dụng, hiệu quả cho các dịch vụ chat, game, notification.
- **API SOAP, RPC:** Các kiểu API truyền thống, sử dụng XML, được dùng cho hệ thống yêu cầu tính đồng bộ khóa chặt.

API giúp tích hợp nhanh các ứng dụng mới với hệ thống hiện tại, đẩy nhanh đổi mới công nghệ, mở rộng khả năng kết nối nhiều nền tảng, đồng thời tăng khả năng bảo trì phần mềm nhờ tách biệt rõ giữa giao diện và logic nội bộ. Các API còn được phân loại theo phạm vi sử dụng: API riêng, API công cộng, API đối tác và API tổng hợp.

### **Kết luận:**

API là nền tảng cho mọi kết nối dữ liệu trong thời đại số hóa, thúc đẩy khả năng tích hợp, mở rộng và tối ưu hóa trải nghiệm người dùng ở mọi cấp độ phát triển phần mềm.

### **2.3.2 Cấu trúc request và response trong API**

Trong môi trường giao tiếp API hiện đại, nhất là API RESTful, mọi hoạt động cần tuân thủ các nguyên tắc cấu trúc request – response chuẩn, đảm bảo tính nhất quán và khả năng tích hợp hệ thống dễ dàng.

#### **- Request:**

**Endpoint:** URL đại diện cho điểm cuối API, ví dụ: /api/v1/users, thường đi kèm tham số truy vấn hoặc ID tài nguyên.

**Phương thức HTTP:** Các verb chuẩn như GET, POST, PUT/PATCH, DELETE.

**Header:** Phân khai báo thông tin metadata yêu cầu.

**Body:** Dữ liệu gửi kèm ở dạng JSON, XML... với các trường dữ liệu phù hợp yêu cầu endpoint.

#### **- Response (Phản hồi):**

**Status Code:** Mã trạng thái HTTP phản ánh kết quả xử lý: 200/201, 400, 401, 403, 404, 500.

**Header:** Metadata phản hồi, thông tin xác thực lại.

**Body:** Nội dung phản hồi ở dạng JSON/XML bao gồm dữ liệu tài nguyên trả về, thông báo lỗi, hoặc thông tin bổ sung cho client.

**Kết luận:** Việc tuân thủ cấu trúc chuẩn request và response API giúp đảm bảo hệ thống backend–frontend giao tiếp ổn định, dễ tích hợp, dễ quản lý và tăng khả năng bảo trì và kiểm thử toàn hệ thống.

## CHƯƠNG 3: HIỆN THỰC HÓA NGHIÊN CỨU

### 3.1. Đặc tả yêu cầu hệ thống

#### 3.1.1. Yêu cầu chức năng

##### Chức năng chung (tất cả người dùng)

Đăng nhập hệ thống bằng email và mật khẩu, đăng xuất khỏi phiên làm việc, đổi mật khẩu và cập nhật thông tin cá nhân.

##### Chức năng Admin (Quản trị viên)

- Quản lý người dùng: thêm, sửa, xóa, tìm kiếm và phân quyền tài khoản.
- Quản lý môn thi: thêm, sửa, xóa, tìm kiếm môn thi.
- Quản lý kỳ thi: thêm, sửa, xóa kỳ thi và cập nhật trạng thái (sắp diễn ra, đang diễn ra, hoàn thành).
- Quản lý học viên: thêm, sửa, xóa thông tin học viên/thí sinh.
- Quản lý điểm thi: xem, sửa, xóa điểm.
- Thống kê báo cáo: xem tổng quan, tỷ lệ đạt/không đạt và xuất báo cáo Excel.

##### Chức năng Giáo viên

Xem danh sách học viên theo kỳ thi. Nhập điểm thủ công qua form hoặc import điểm hàng loạt từ file Excel. Cập nhật và sửa điểm đã nhập.

##### Chức năng Sinh viên/Học viên

Tra cứu điểm các kỳ thi đã tham gia. Xem lịch sử thi và thông tin chi tiết kỳ thi (môn, thời gian, địa điểm).

#### 3.1.2. Yêu cầu phi chức năng

- Bảo mật: Mã hóa mật khẩu bằng bcrypt, xác thực người dùng bằng JWT, phân quyền theo vai trò (RBAC).
- Hiệu năng: Tải trang dưới 3 giây, hỗ trợ đồng thời nhiều người dùng truy cập.
- Khả dụng: Hoạt động ổn định 24/7, triển khai trên nền tảng cloud (Vercel, Render).
- Tương thích: Hỗ trợ các trình duyệt phổ biến (Chrome, Firefox, Safari, Edge), giao diện responsive trên máy tính, tablet và điện thoại.
- Sao lưu: Dữ liệu lưu trữ trên MongoDB Atlas với cơ chế backup tự động.

## 3.2. Phân tích hệ thống

### 3.2.1 Sơ đồ Use Case tổng quát



Hình 3.1 Sơ đồ Use Case hệ thống quản lý Kỳ thi

### 3.2.2 Đặc tả chi tiết các Use Case chính

#### UC01: Đăng nhập

Actor: Admin, Giáo viên, Học viên

Mô tả: Người dùng đăng nhập vào hệ thống để sử dụng các chức năng

Luồng chính:

- Người dùng truy cập trang đăng nhập
- Nhập email và mật khẩu
- Nhấn nút "Đăng nhập"
- Hệ thống xác thực thông tin
- Đăng nhập thành công, chuyển đến trang Dashboard tương ứng với vai trò

Luồng phụ:

- 4a. Sai email hoặc mật khẩu: Hiện thị thông báo lỗi, yêu cầu nhập lại
- 4b. Tài khoản bị khóa: Hiện thị thông báo liên hệ quản trị viên

### **UC02: Quản lý người dùng**

Actor: Admin

Mô tả: Admin quản lý tài khoản người dùng trong hệ thống

Luồng chính:

- Admin chọn menu "Quản lý người dùng"
- Hệ thống hiển thị danh sách người dùng
- Admin thực hiện thao tác: Thêm/Sửa/Xóa/Tìm kiếm
- Hệ thống cập nhật và hiển thị kết quả

Luồng phụ:

- 3a. Thêm mới: Nhập thông tin → Chọn vai trò → Lưu
- 3b. Sửa: Chọn người dùng → Chỉnh sửa thông tin → Lưu
- 3c. Xóa: Chọn người dùng → Xác nhận xóa → Xóa thành công
- 3d. Tìm kiếm: Nhập từ khóa → Hiện thị kết quả phù hợp

### **UC03: Quản lý kỳ thi**

Actor: Admin

Mô tả: Admin quản lý thông tin các kỳ thi

Luồng chính:

- Admin chọn menu "Quản lý kỳ thi"
- Hệ thống hiển thị danh sách kỳ thi
- Admin thực hiện thao tác: Thêm/Sửa/Xóa/Cập nhật trạng thái
- Hệ thống cập nhật và hiển thị kết quả



Luồng phụ:

- 3a. Thêm mới: Nhập tên kỳ thi, chọn môn thi, ngày thi → Lưu
- 3b. Cập nhật trạng thái: Chọn kỳ thi → Đổi trạng thái (Sắp diễn ra/Đang diễn ra/Hoàn thành)

#### **UC04: Nhập điểm**

Actor: Giáo viên

Mô tả: Giáo viên nhập điểm cho học viên

Luồng chính:

- Giáo viên chọn kỳ thi cần nhập điểm
- Hệ thống hiển thị danh sách học viên
- Giáo viên nhập điểm cho từng học viên
- Nhấn "Lưu điểm"
- Hệ thống lưu và hiển thị thông báo thành công

Luồng phụ:

- 3a. Import Excel: Chọn file Excel → Hệ thống đọc và hiển thị dữ liệu → Xác nhận lưu
- 4a. Điểm không hợp lệ: Hiển thị lỗi, yêu cầu nhập lại

#### **UC05: Tra cứu điểm thi**

Actor: Học viên

Mô tả: Học viên tra cứu kết quả thi của mình

Luồng chính:

- Học viên đăng nhập vào hệ thống
- Chọn menu "Tra cứu điểm"
- Hệ thống hiển thị danh sách kỳ thi đã tham gia
- Học viên chọn kỳ thi cần xem
- Hệ thống hiển thị chi tiết điểm thi

Luồng phụ:

- 3a. Chưa có kỳ thi nào: Hiển thị thông báo "Bạn chưa tham gia kỳ thi nào"

#### **UC06: Thống kê báo cáo**

Actor: Admin

Mô tả: Admin xem thống kê tổng quan về kỳ thi và kết quả

Luồng chính:

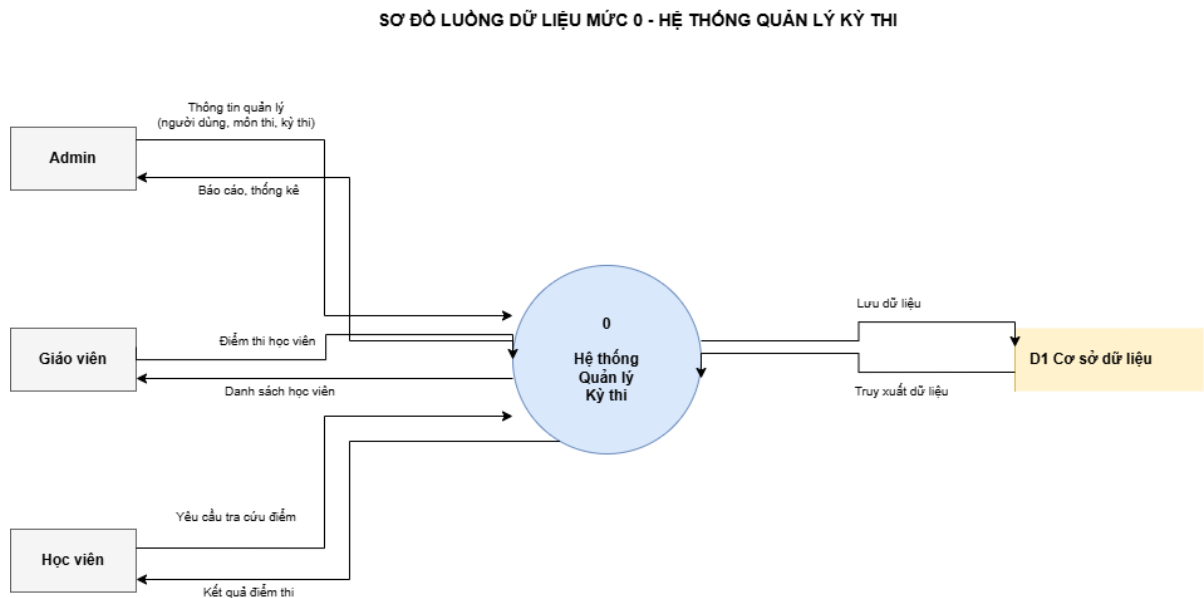
- Admin chọn menu "Thống kê"
- Hệ thống hiển thị Dashboard thống kê: tổng số kỳ thi, học viên, tỷ lệ đạt/không đạt
- Admin chọn kỳ thi cụ thể để xem chi tiết
- Hệ thống hiển thị biểu đồ và số liệu chi tiết

Luồng phụ:

- 3a. Xuất Excel: Chọn "Xuất báo cáo" → Hệ thống tạo file Excel → Tải về

### 3.3. Thiết kế hệ thống

#### 3.3.1 Sơ đồ mức 0 (Context Diagram)



Hình 3.2 Sơ đồ mức 0 (Context Diagram) - Hệ thống quản lý kỳ thi

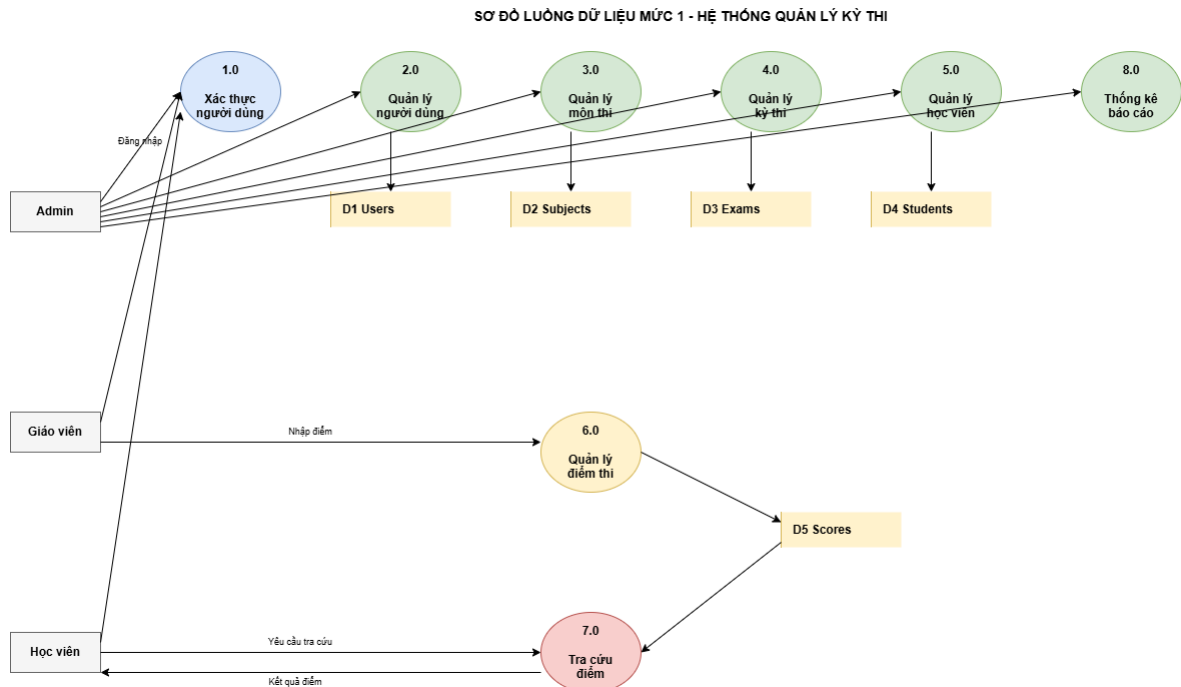
#### Mô tả sơ đồ luồng dữ liệu

##### Sơ đồ mức 0 (Context Diagram)

Sơ đồ mức 0 thể hiện tổng quan hệ thống với một tiến trình trung tâm "Hệ thống Quản lý Kỳ thi" tương tác với 3 thực thể ngoài: Admin, Giáo viên và Học viên. Admin gửi thông tin quản lý và nhận báo cáo thống kê. Giáo viên gửi điểm thi và

nhận danh sách học viên. Học viên gửi yêu cầu tra cứu và nhận kết quả điểm. Tất cả dữ liệu được lưu trữ và truy xuất từ cơ sở dữ liệu.

### 3.3.2 Sơ đồ mức 1 (Decomposition Diagram)



Hình 3.3 Sơ đồ mức 1 (Decomposition Diagram) - Hệ thống quản lý kỳ thi

#### Mô tả sơ đồ luồng dữ liệu

Sơ đồ mức 1 phân rã tiến trình trung tâm thành 8 tiến trình con:

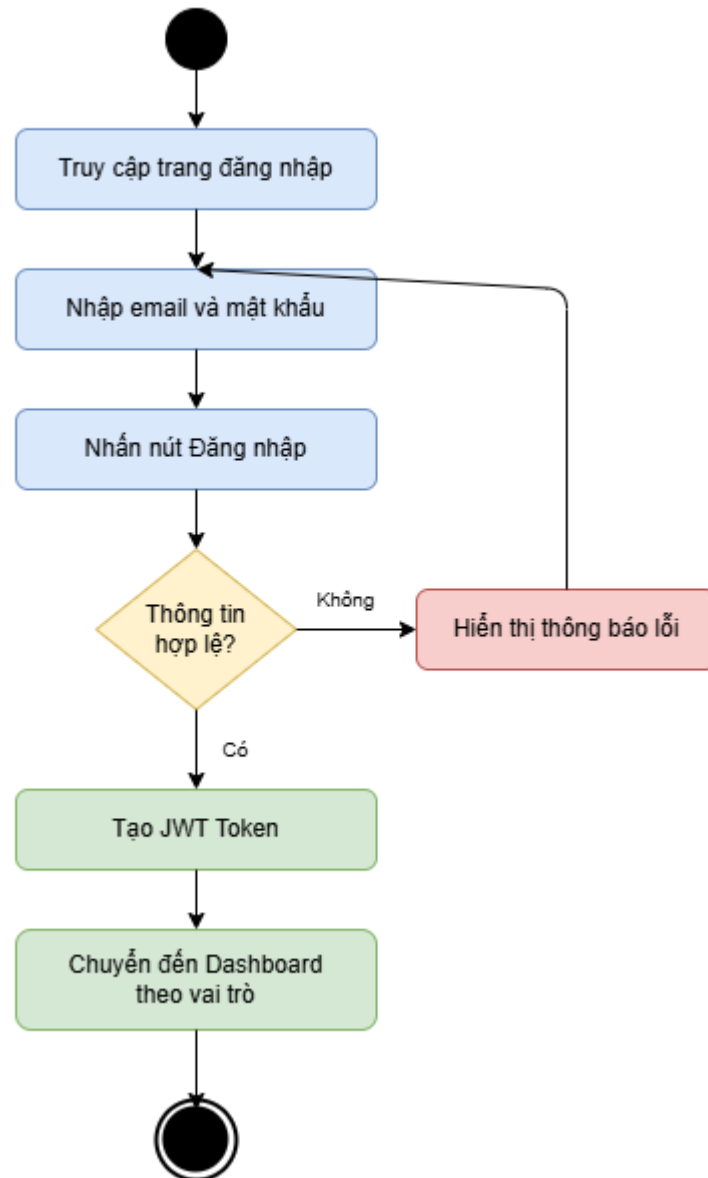
- 1.0 Xác thực người dùng: Xử lý đăng nhập, đăng xuất cho tất cả người dùng
- 2.0 Quản lý người dùng: Admin thêm, sửa, xóa tài khoản
- 3.0 Quản lý môn thi: Admin quản lý danh mục môn thi
- 4.0 Quản lý kỳ thi: Admin tạo và cập nhật kỳ thi
- 5.0 Quản lý học viên: Admin quản lý thông tin học viên
- 6.0 Quản lý điểm thi: Giáo viên nhập và cập nhật điểm
- 7.0 Tra cứu điểm: Học viên xem kết quả thi
- 8.0 Thống kê báo cáo: Admin xem thống kê và xuất báo cáo

Dữ liệu được lưu trong 5 kho: Users, Subjects, Exams, Students, Scores.

### 3.3.3 Sơ đồ hoạt động (Activity Diagram)

#### Sơ đồ hoạt động - Đăng nhập

### SƠ ĐỒ HOẠT ĐỘNG - ĐĂNG NHẬP

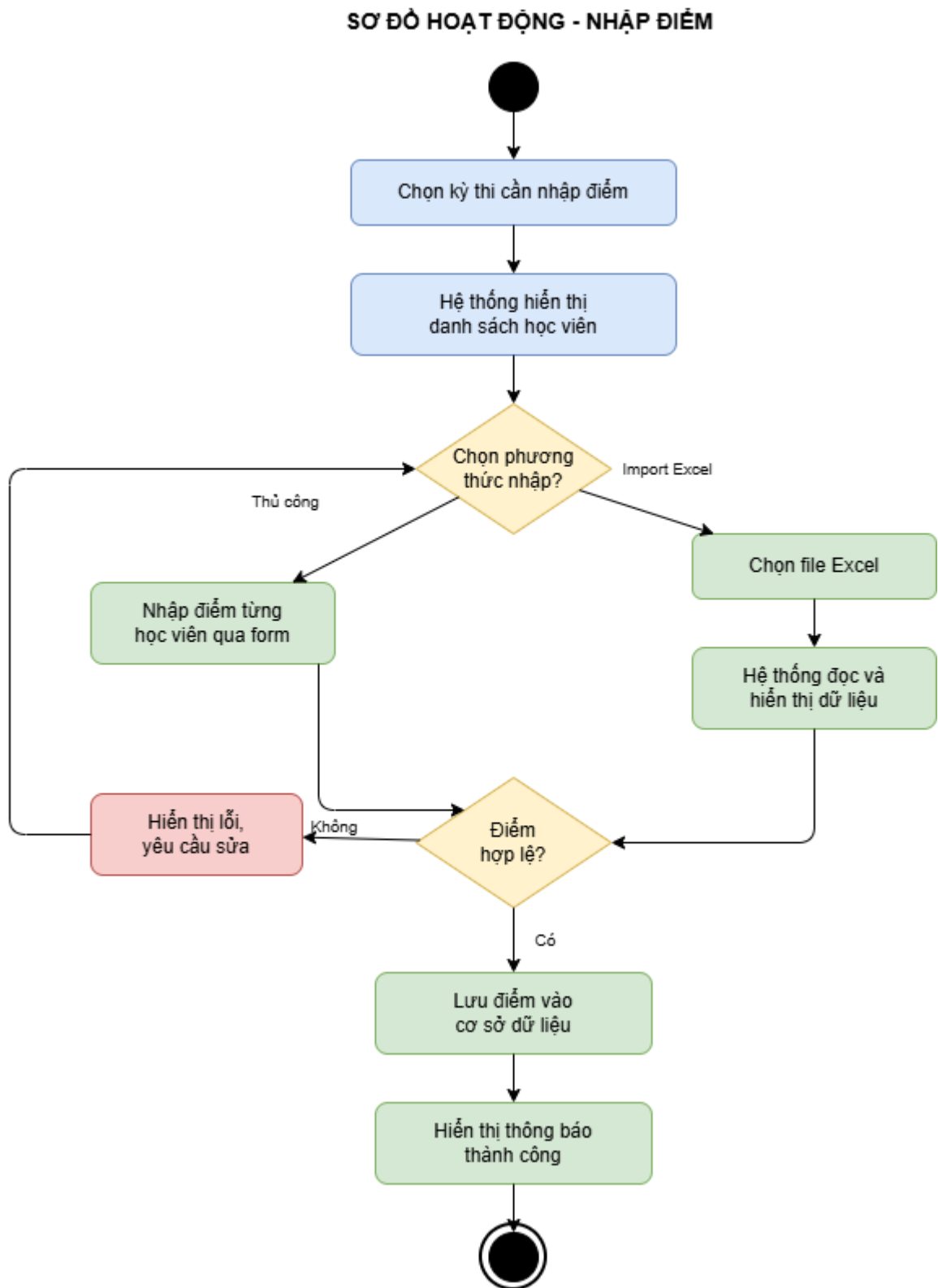


Hình 3.4 Sơ đồ hoạt động - Đăng nhập

### Mô tả sơ đồ hoạt động

Người dùng truy cập trang đăng nhập, nhập email và mật khẩu, sau đó nhấn nút "Đăng nhập". Hệ thống kiểm tra thông tin: nếu hợp lệ thì tạo JWT Token và chuyển đến Dashboard tương ứng với vai trò (Admin/Giáo viên/Học viên); nếu không hợp lệ thì hiển thị thông báo lỗi và yêu cầu nhập lại.

### Sơ đồ hoạt động - Nhập điểm



Hình 3.5 Sơ đồ hoạt động - Nhập điểm

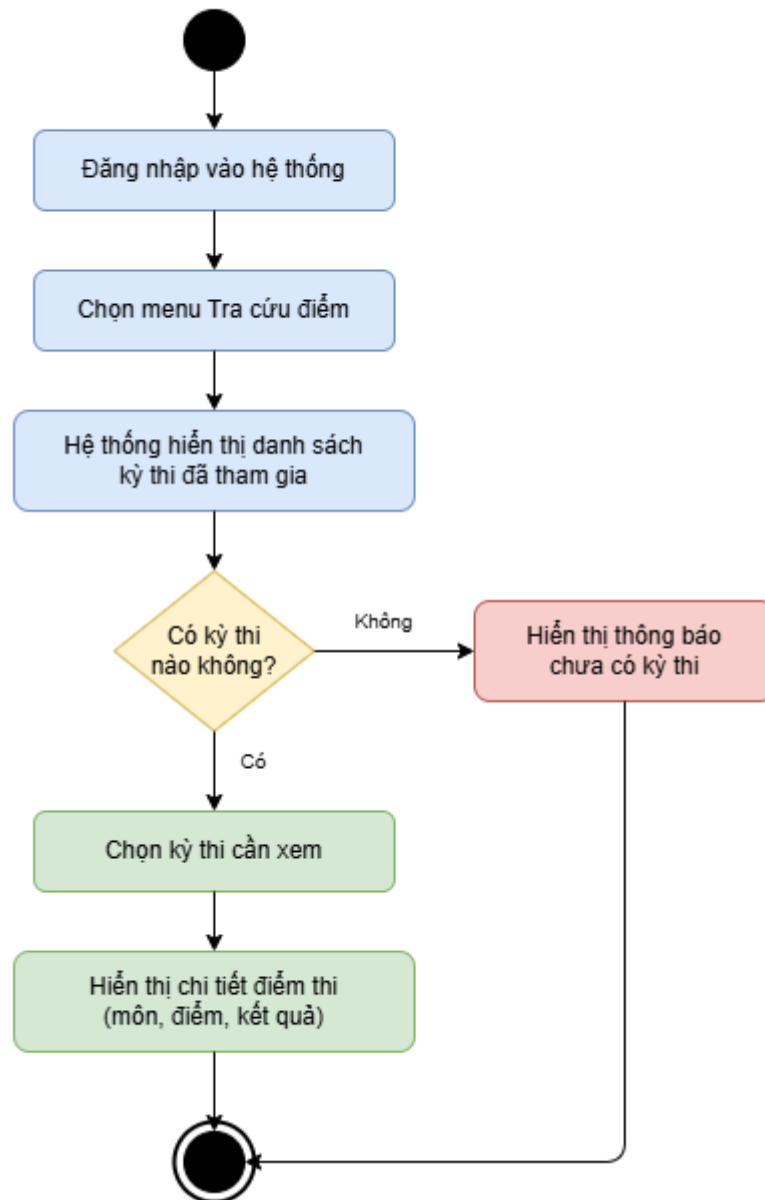
### Mô tả sơ đồ hoạt động

Giáo viên chọn kỳ thi cần nhập điểm, hệ thống hiển thị danh sách học viên. Giáo viên chọn phương thức nhập: nhập thủ công từng học viên qua form hoặc import

từ file Excel. Hệ thống kiểm tra điểm: nếu hợp lệ (0-10) thì lưu vào cơ sở dữ liệu và hiển thị thông báo thành công; nếu không hợp lệ thì hiển thị lỗi và yêu cầu sửa lại.

### Sơ đồ hoạt động - Tra cứu điểm

#### SƠ ĐỒ HOẠT ĐỘNG - TRA CỨU ĐIỂM



Hình 3.6 Sơ đồ hoạt động - Tra cứu điểm

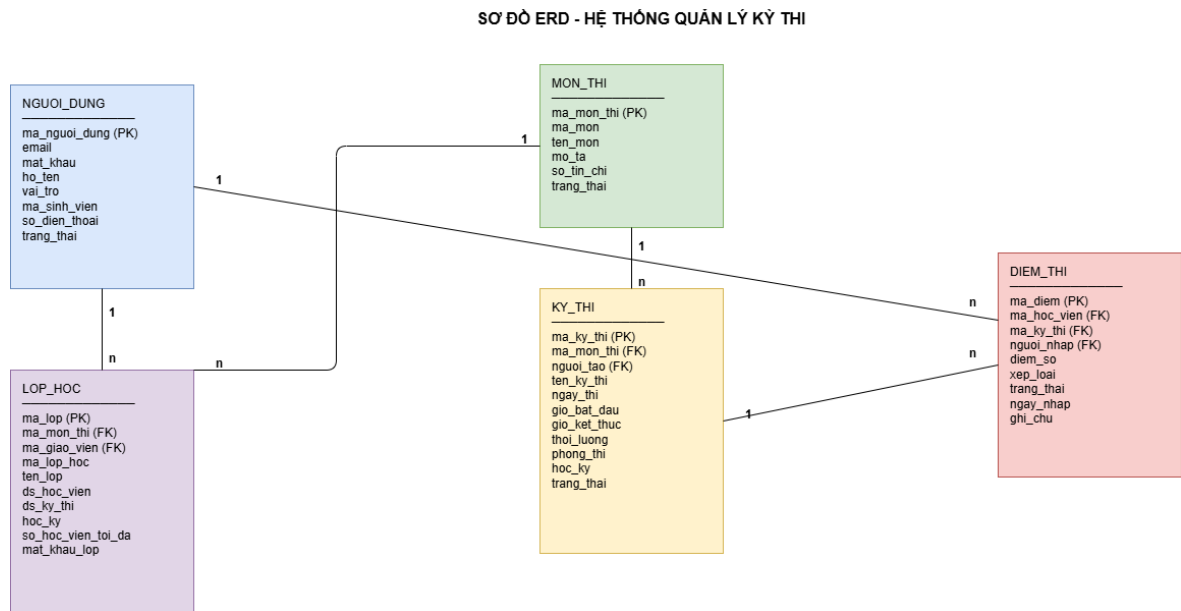
### Mô tả sơ đồ hoạt động

Học viên đăng nhập vào hệ thống, chọn menu "Tra cứu điểm". Hệ thống hiển thị danh sách các kỳ thi đã tham gia. Nếu có kỳ thi, học viên chọn kỳ thi cần xem và

hệ thống hiển thị chi tiết điểm (môn thi, điểm số, kết quả đạt/không đạt). Nếu chưa có kỳ thi nào, hệ thống hiển thị thông báo "Bạn chưa tham gia kỳ thi nào".

### 3.4. Thiết kế cơ sở dữ liệu

#### 3.4.1 Sơ đồ ERD



Hình 3.7 Sơ đồ ERD - Hệ thống quản lý kỳ thi

#### Mô tả cấu trúc cơ sở dữ liệu

##### Bảng NGUOI\_DUNG

ma\_nguoi\_dung(PK), email, mat\_khau, ho\_ten, vai\_tru  
(admin/giao\_vien/hoc\_vien), ma\_sinh\_vien, so\_dien\_thoai, trang\_thai

##### Bảng MON\_THI

ma\_mon\_thi (PK), ma\_mon, ten\_mon, mo\_ta, so\_tin\_chi, trang\_thai

##### Bảng KY\_THI

ma\_ky\_thi (PK), ma\_mon\_thi (FK), nguoi\_tao (FK), ten\_ky\_thi, ngay\_thi,  
gio\_bat\_dau, gio\_ket\_thuc, thoi\_luong, phong\_thi, hoc\_ky, trang\_thai

##### Bảng DIEM\_THI

ma\_diem (PK), ma\_hoc\_vien (FK), ma\_ky\_thi (FK), nguoi\_nhap (FK), diem\_so, xep\_loai, trang\_thai, ngay\_nhap, ghi\_chu

### **Bảng LOP\_HOC**

ma\_lop (PK), ma\_mon\_thi (FK), ma\_giao\_vien (FK), ma\_lop\_hoc, ten\_lop, ds\_hoc\_vien, ds\_ky\_thi, hoc\_ky, so\_hoc\_vien\_toi\_da, mat\_khau\_lop

### **Mối quan hệ:**

MON\_THI (1) → (n) KY\_THI

KY\_THI (1) → (n) DIEM\_THI

NGUOI\_DUNG (1) → (n) DIEM\_THI

MON\_THI (1) → (n) LOP\_HOC

NGUOI\_DUNG (1) → (n) LOP\_HOC

### **3.4.2 Bảng thuộc tính**

Thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
ma_nguoi_dung	ObjectId	PK	Mã định danh người dùng
email	String	Unique, Required	Địa chỉ email đăng nhập
mat_khau	String	Required	Mật khẩu (mã hóa bằng bcrypt)
ho_ten	String	Required	Họ và tên



vai_tro	Enum	Required	admin / teacher / student
ma_sinh_vien	String	Unique	Mã sinh viên (chỉ áp dụng học viên)
so_dien_thoai	String		Số điện thoại
trang_thai	Boolean	Default: true	Trạng thái hoạt động

*Bảng 3.1 NGUOI\_DUNG (Users)*

Thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
ma_mon_thi	ObjectId	PK	Mã định danh môn thi
ma_mon	String	Unique, Required	Mã môn (VD: ENG101)
ten_mon	String	Required	Tên môn thi
mo_ta	String		Mô tả môn thi
so_tin_chi	Number	Min: 1, Max: 10	Số tín chỉ
trang_thai	Boolean	Default: true	Trạng thái hoạt động

*Bảng 3.2 MON\_THI (Subjects)*

Thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
ma_ky_thi	ObjectId	PK	Mã định danh kỳ thi
ma_mon_thi	ObjectId	FK → MON_THI	Môn thi
nguo_i_tao	ObjectId	FK → NGUOI_DUNG	Admin tạo kỳ thi
ten_ky_thi	String	Required	Tên kỳ thi
ngay_thi	Date	Required	Ngày thi
gio_bat_dau	String	Default: "08:00"	Giờ bắt đầu
gio_ket_thuc	String	Default: "10:00"	Giờ kết thúc
thoi_luong	Number	Min: 15, Max: 300	Thời gian thi (phút)
phong_thi	String		Phòng thi
hoc_ky	String	Required	Học kỳ (VD: HK1 2024–2025)
nam_hoc	String	Required	Năm học

trang_thai	Enum	Default: upcoming	upcoming / ongoing / completed / cancelled
------------	------	-------------------	--

*Bảng 3.3 KY\_THI (Exams)*

Thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
ma_diem	ObjectId	PK	Mã định danh điểm
ma_hoc_vien	ObjectId	FK → NGUOI_DUNG	Học viên
ma_ky_thi	ObjectId	FK → KY_THI	Kỳ thi
nguai_nhap	ObjectId	FK → NGUOI_DUNG	Giáo viên nhập điểm
diem_so	Number	Required, Min: 0, Max: 10	Điểm số
xep_loai	Enum		A / B / C / D / F (tự động tính)
trang_thai	Enum	Default: entered	pending / entered / verified / published
ngay_nhap	Date	Default: now	Ngày nhập điểm

nguoai_duyet	ObjectId	FK → NGUOI_DUNG	Admin duyệt điểm
ghi_chu	String		Ghi chú

*Bảng 3.4 DIEM\_THI (Scores)*

Thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
ma_lop	ObjectId	PK	Mã định danh lớp
ma_mon_thi	ObjectId	FK → MON_THI	Môn học
ma_giao_vien	ObjectId	FK → NGUOI_DUNG	Giáo viên phụ trách
ma_lop_hoc	String	Unique, Required	Mã lớp (VD: ENG101-A)
ten_lop	String	Required	Tên lớp
ds_hoc_vien	[ObjectId]	FK → NGUOI_DUNG	Danh sách học viên
ds_ky_thi	[ObjectId]	FK → KY_THI	Danh sách kỳ thi

hoc_ky	String	Required	Học kỳ
nam_hoc	String	Required	Năm học
lich_hoc	String		Lịch học
phong_hoc	String		Phòng học
so_hoc_vien_toi_da	Number	Default: 50	Số học viên tối đa
mat_khau_lop	String	Required	Mật khẩu tham gia lớp
trang_thai	Boolean	Default: true	Trạng thái hoạt động

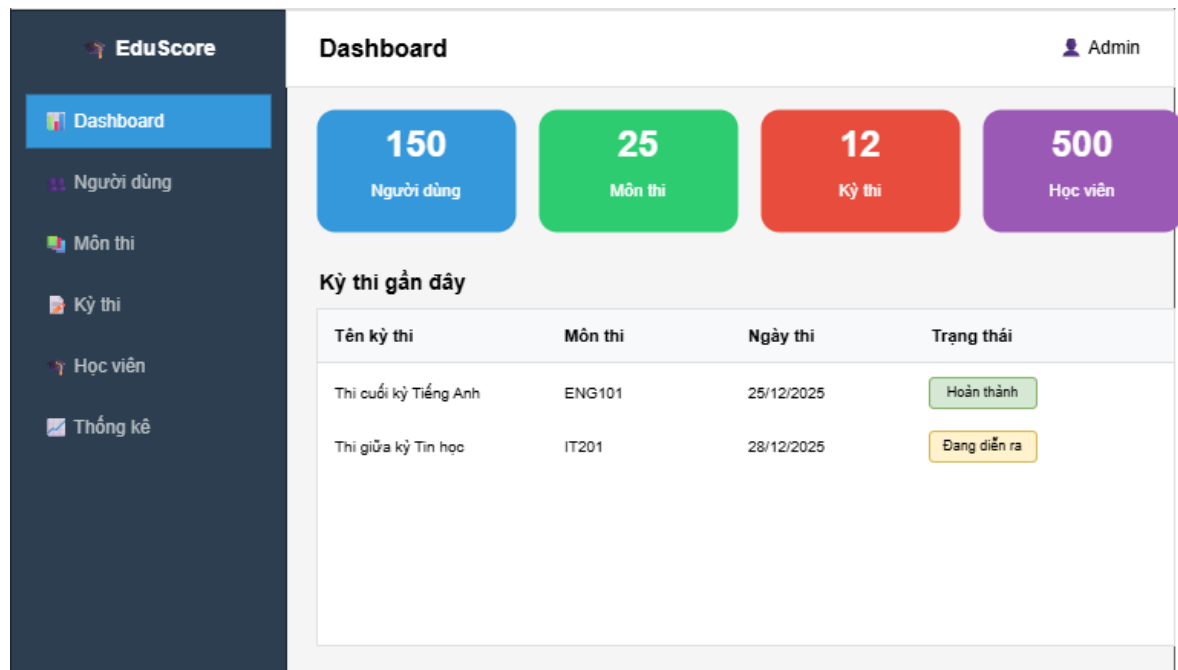
Bảng 3.5 LOP\_HOC (Classes)

### 3.5. Thiết kế giao diện

#### Wireframe 1: Trang đăng nhập

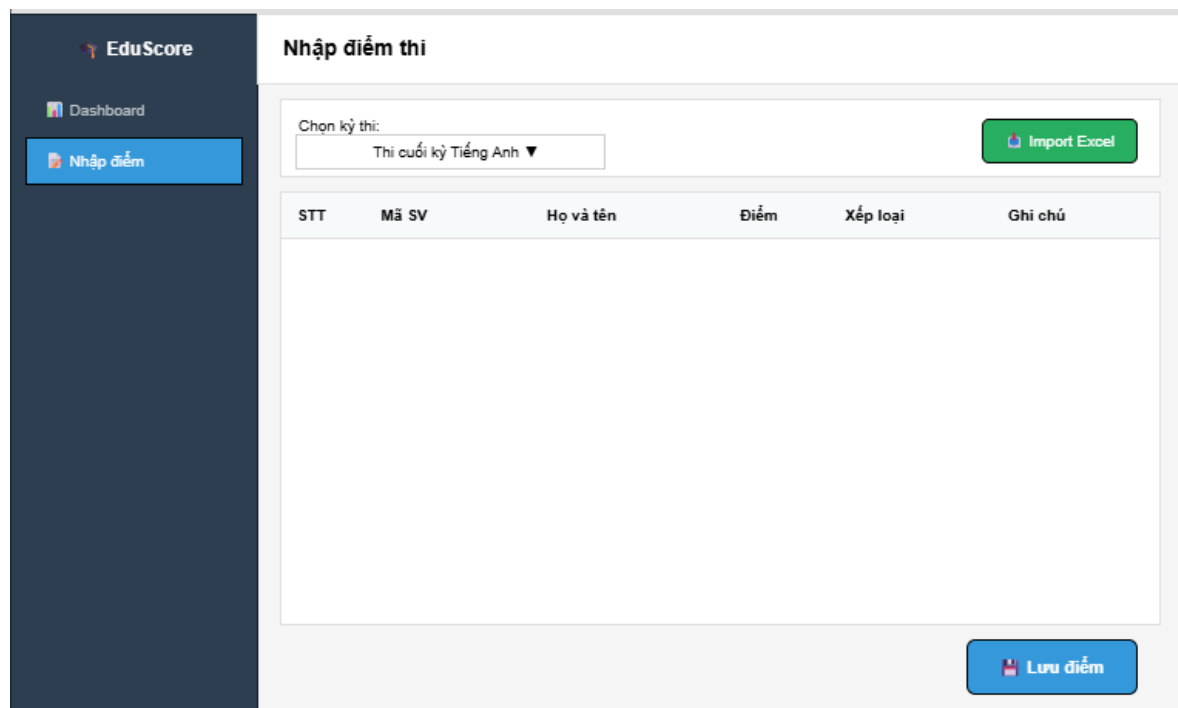
Hình 3.8 Thiết kế wireframe trang đăng nhập

## Wireframe 2: Trang chủ quản trị viên (Dashboard Admin )



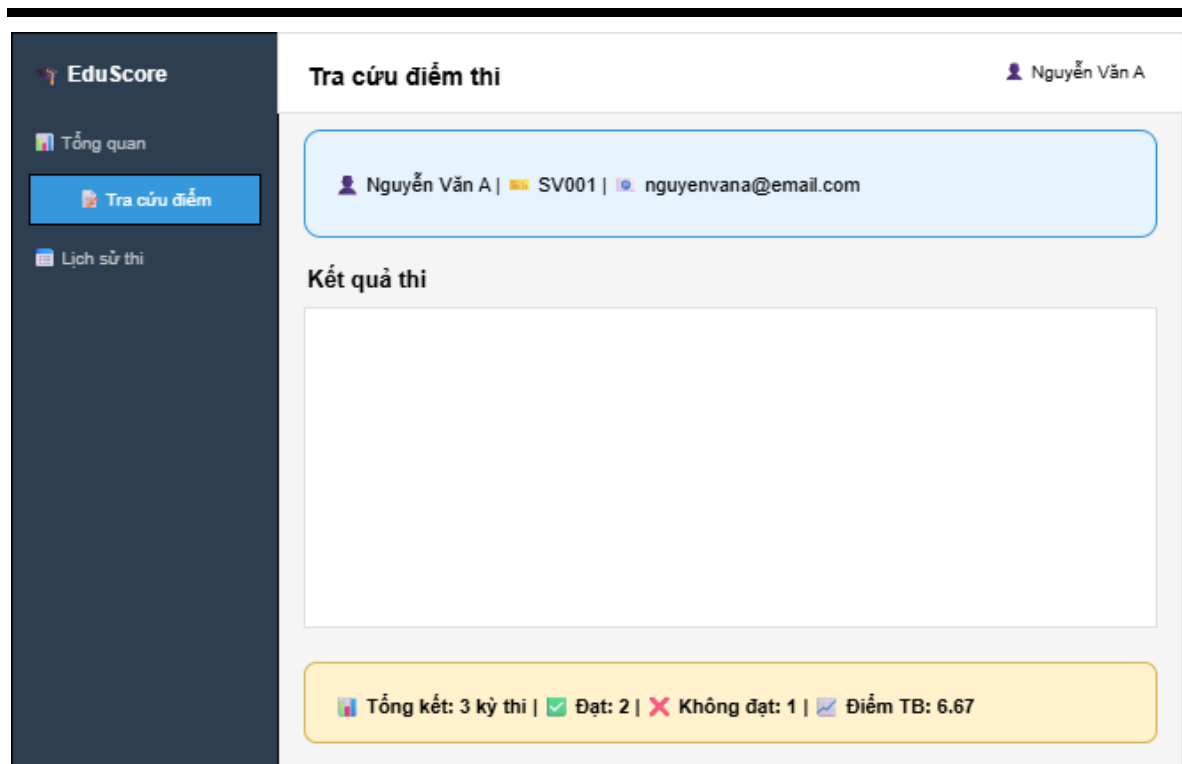
Hình 3.9 Thiết kế wireframe trang chủ quản trị viên

## Wireframe 3: Trang nhập điểm



Hình 3.10 Thiết kế wireframe trang nhập điểm

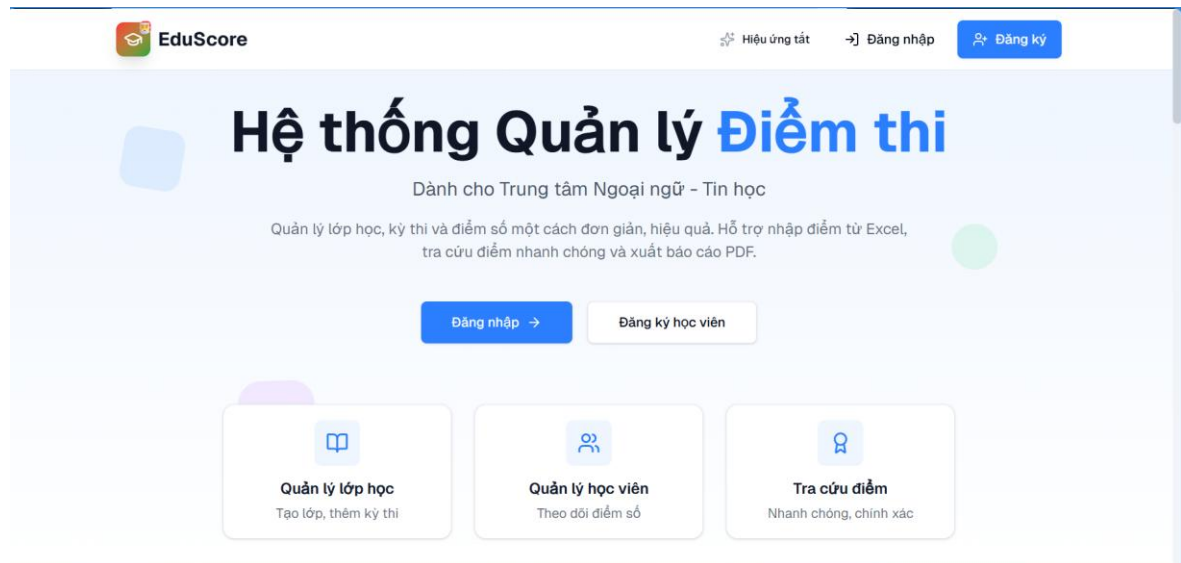
## Wireframe 4: Trang tra cứu điểm (Học viên)



Hình 3.11 Thiết kế wireframe trang tra cứu điểm (Học viên)

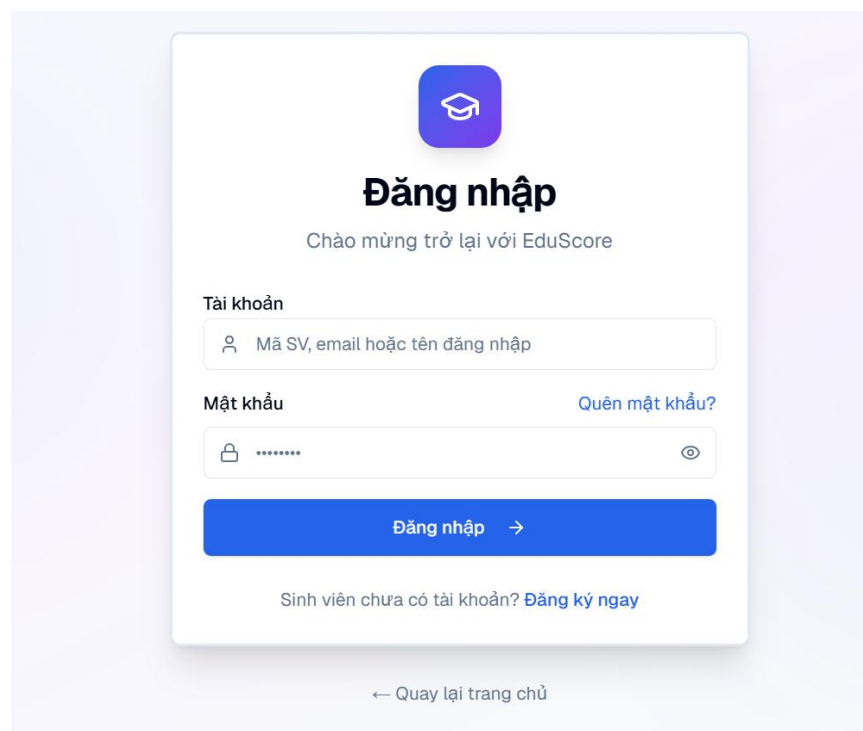
## CHƯƠNG 4: KẾT QUẢ NGHIÊN CỨU

### 4.1. Giao diện chung



Hình 4.1: Trang chủ giới thiệu hệ thống

Trang chủ giới thiệu tổng quan về hệ thống EduScore với các tính năng nổi bật, 3 vai trò người dùng và nút đăng nhập/đăng ký.



Hình 4.2: Trang đăng nhập

Người dùng nhập email và mật khẩu để đăng nhập. Hệ thống tự động phân quyền và chuyển đến Dashboard tương ứng.



**Đăng ký tài khoản**  
Đăng ký tài khoản dành cho sinh viên

Mã sinh viên  
123456789  
Email tự động: 123456789@gmail.com

Họ và tên  
Nguyễn Văn An

Số điện thoại (tùy chọn)  
0901234567

Mật khẩu  
Tối thiểu 6 ký tự

Xác nhận mật khẩu

**Đăng ký** →

Đã có tài khoản? [Đăng nhập](#)

\* Lưu ý: Chức năng đăng ký chỉ dành cho sinh viên.  
Giáo viên và Quản trị viên liên hệ phòng đào tạo để được cấp tài khoản.

[← Quay lại trang chủ](#)

Hình 4.3: Trang đăng ký

Người dùng mới tạo tài khoản với thông tin: họ tên, email, mật khẩu và vai trò.

## 4.2. Giao diện Admin

**EduScore Admin Panel**

**Dashboard Admin**  
Tổng quan hệ thống quản lý

**1** Học viên  
**1** Giáo viên  
**3** Lớp học  
**3** Môn học  
**3** Kỳ thi  
**0** Đang thi

Kỳ thi sắp tới: **0**  
Đang diễn ra: **0**  
Đã kết thúc: **3**

**Kỳ thi gần đây**  
Danh sách các kỳ thi mới nhất

Anh văn không chuyên 2  
27/12/2025 • 08:00 - 10:00  
Đã kết thúc

Tin Học cơ bản  
26/12/2025 • 18:20 - 19:00  
Đã kết thúc

**Thao tác nhanh**  
Truy cập nhanh các chức năng

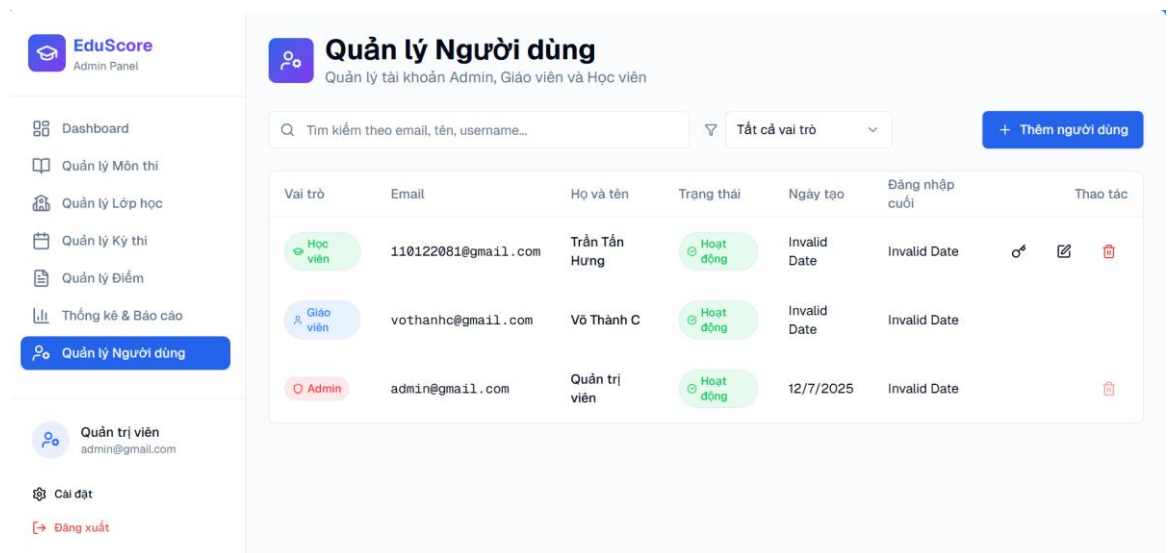
[Quản lý kỳ thi](#)  
[Quản lý lớp học](#)  
[Quản lý môn học](#)

Quản trị viên  
admin@gmail.com

Cài đặt  
[Đăng xuất](#)

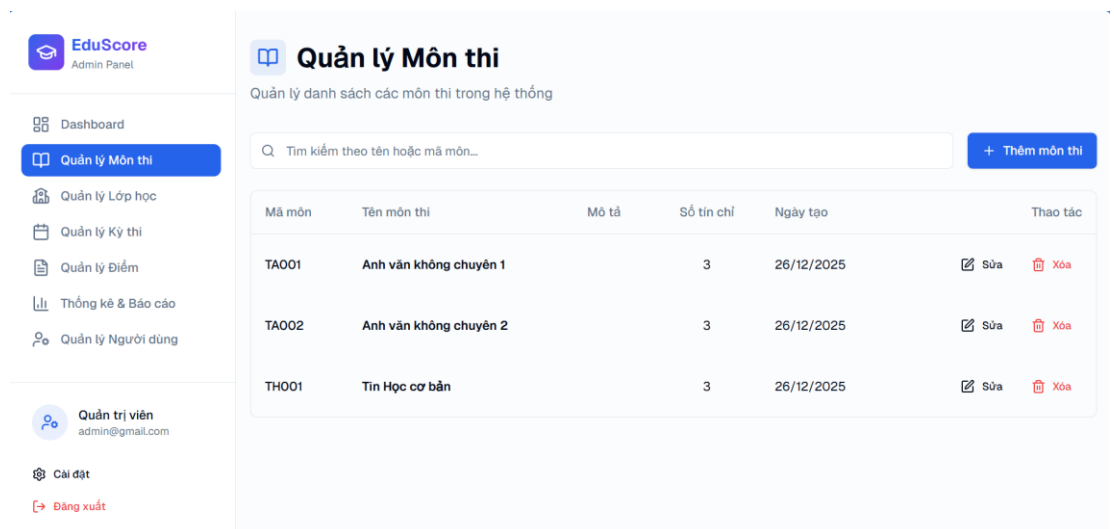
Hình 4.4: Trang chủ quản trị viên (Dashboard Admin)

Hiện thị thông kê tổng quan: số người dùng, môn thi, kỳ thi, học viên và danh sách kỳ thi gần đây.



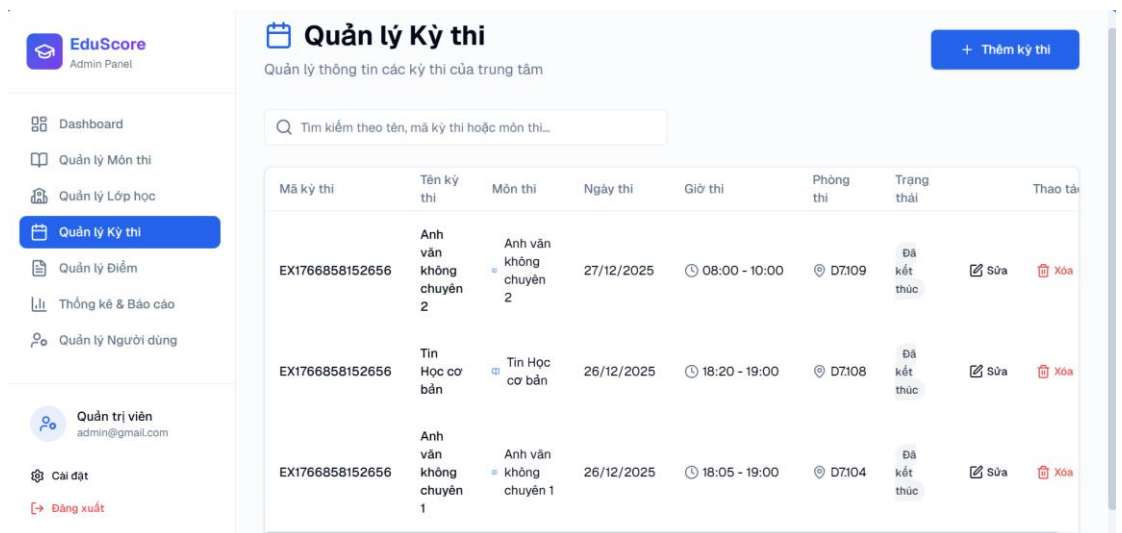
Hình 4.5: Quản lý người dùng

Admin thêm, sửa, xóa và tìm kiếm tài khoản người dùng. Phân quyền theo vai trò.



Hình 4.6: Quản lý môn thi

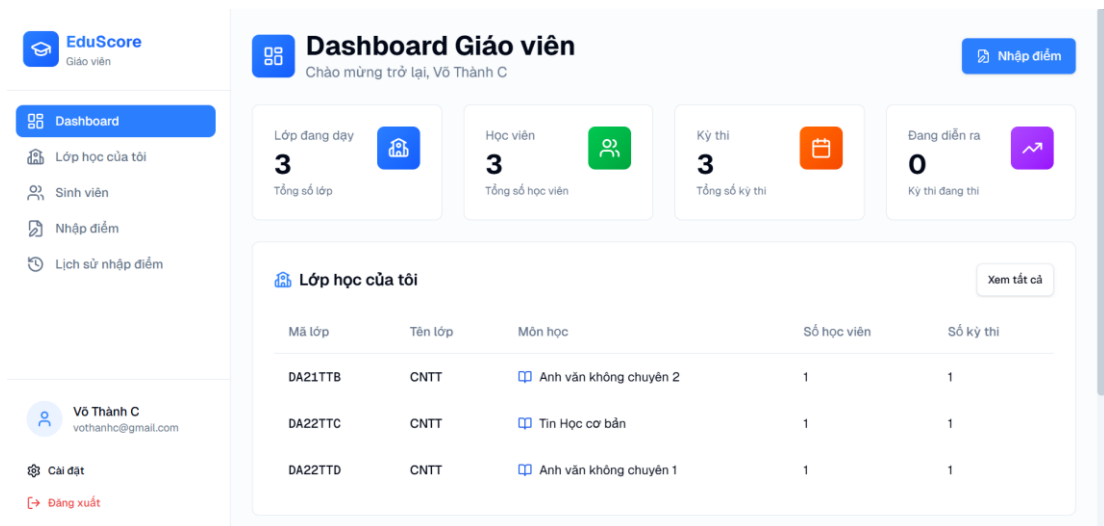
Admin quản lý danh sách môn thi với mã môn, tên môn và số tín chỉ.



Hình 4.7: Quản lý kỳ thi

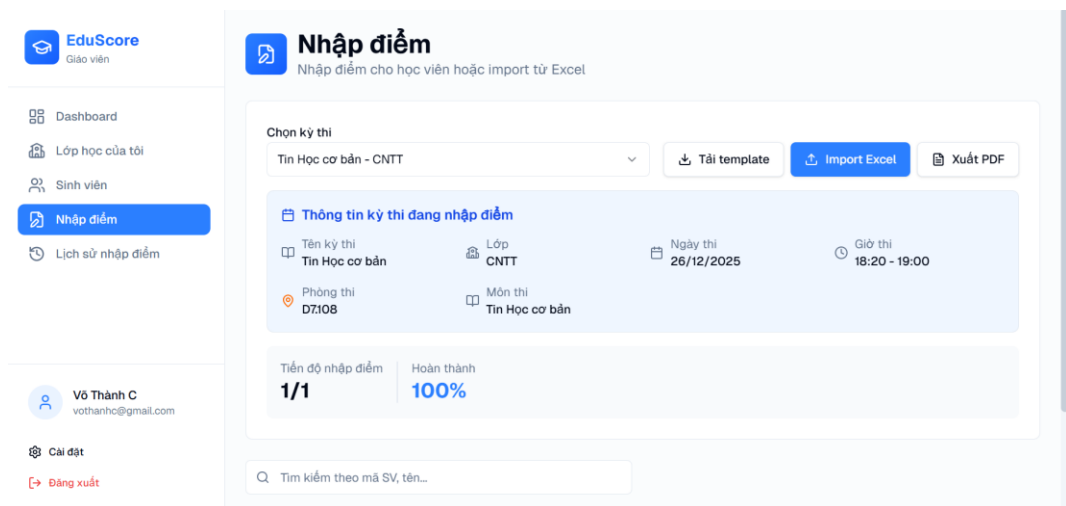
Admin tạo kỳ thi mới, cập nhật trạng thái và xem danh sách kỳ thi.

### 4.3. Giao diện Giáo viên



Hình 4.8: Dashboard Giáo viên

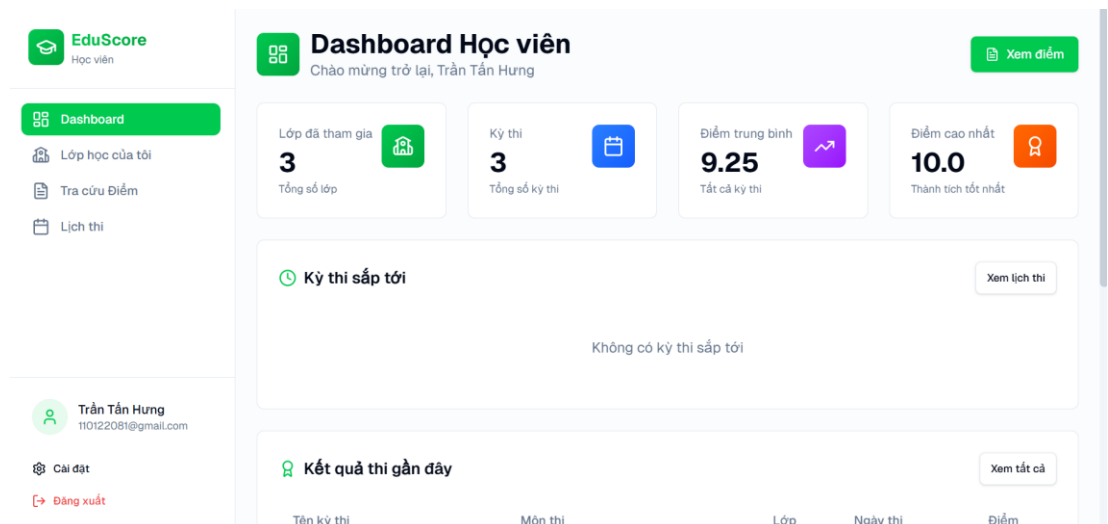
Hiện thị danh sách lớp phụ trách và các kỳ thi cần nhập điểm.



Hình 4.9: Trang nhập điểm

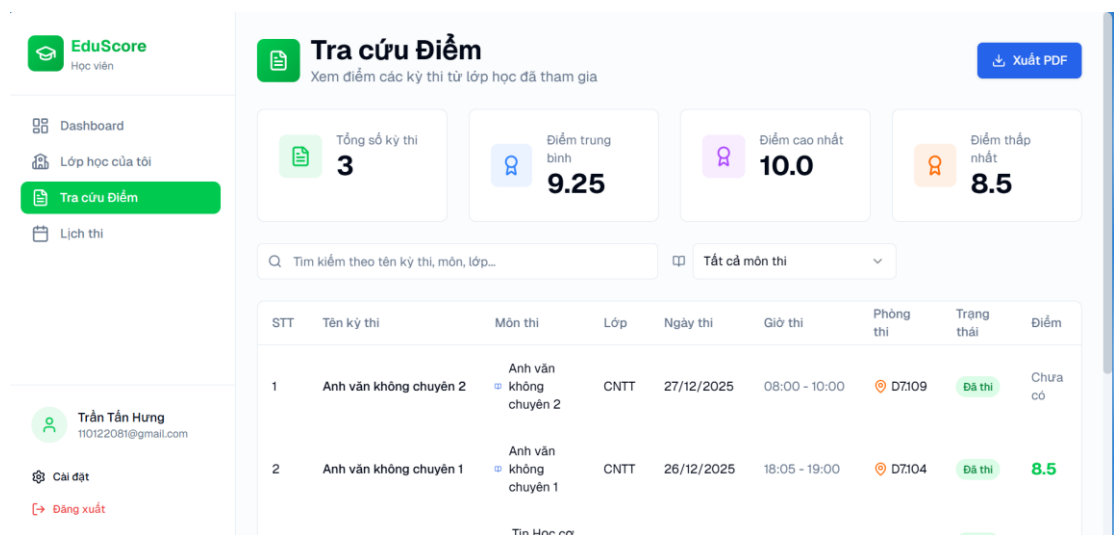
Giáo viên chọn kỳ thi, nhập điểm cho từng học viên hoặc import từ file Excel.

#### 4.4. Giao diện Học viên



Hình 4.10: Dashboard Học viên

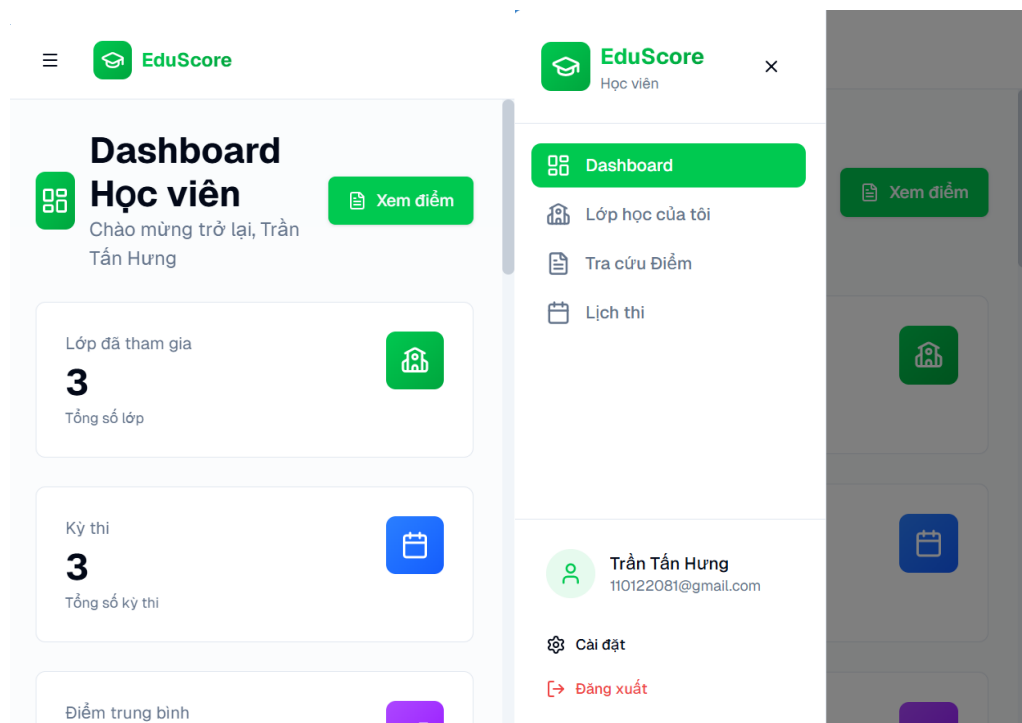
Hiển thị thông tin cá nhân và tổng quan kết quả học tập.



Hình 4.11: Tra cứu điểm thi

Học viên xem kết quả các kỳ thi đã tham gia với điểm số và xếp loại.

#### 4.5. Giao diện Responsive



Hình 4.12: Giao diện trên điện thoại

Hệ thống hiển thị tốt trên thiết bị di động với menu thu gọn và bố cục phù hợp.

## CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

### 5.1. Kết luận

#### Kết quả đạt được:

- Đồ án đã hoàn thành xây dựng hệ thống website quản lý kỳ thi EduScore với các kết quả chính:
- Xây dựng thành công hệ thống với 3 vai trò người dùng: Admin, Giáo viên, Học viên
- Hoàn thiện các chức năng cốt lõi: đăng nhập/đăng ký, quản lý người dùng, quản lý môn thi, quản lý kỳ thi, nhập điểm và tra cứu điểm
- Thiết kế giao diện hiện đại, responsive, tương thích đa thiết bị
- Triển khai thành công hệ thống lên môi trường production (Vercel + Render)

#### Đóng góp của đề tài:

- Số hóa quy trình quản lý kỳ thi, giảm thiểu công việc thủ công
- Cung cấp giải pháp tra cứu điểm trực tuyến cho học viên
- Áp dụng các công nghệ web hiện đại: Next.js, Node.js, MongoDB
- Xây dựng hệ thống bảo mật với JWT Authentication và mã hóa mật khẩu

#### Hạn chế:

- Chưa hoàn thiện chức năng import điểm từ Excel
- Chưa có chức năng xuất báo cáo PDF/Excel
- Chưa tích hợp gửi email thông báo kết quả
- Chưa có module thống kê biểu đồ chi tiết

### 5.2. Hướng phát triển

#### Hoàn thiện các chức năng còn thiếu:

- Phát triển chức năng import/export điểm từ file Excel
- Xây dựng module thống kê với biểu đồ trực quan (Chart.js)
- Tích hợp gửi email thông báo kết quả thi cho học viên
- Bổ sung lịch sử chỉnh sửa điểm để theo dõi thay đổi

#### Mở rộng tính năng:

- Phát triển chức năng thi trắc nghiệm trực tuyến
- Xây dựng hệ thống chấm điểm tự động
- Tạo chứng chỉ điện tử cho học viên đạt yêu cầu
- Phát triển ứng dụng mobile (React Native)

**Nâng cao hệ thống:**

- Tối ưu hiệu năng để hỗ trợ số lượng người dùng lớn hơn
- Bổ sung tính năng sao lưu và phục hồi dữ liệu
- Tích hợp đăng nhập qua Google, Facebook
- Triển khai hệ thống thông báo realtime (WebSocket)

## DANH MỤC TÀI LIỆU THAM KHẢO

- [1] Vercel Inc. (2024). Next.js Documentation. <https://nextjs.org/docs>
- [2] Meta Platforms Inc. (2024). React Documentation. <https://react.dev>
- [3] OpenJS Foundation. (2024). Node.js Documentation. <https://nodejs.org/docs>
- [4] OpenJS Foundation. (2024). Express.js Documentation. <https://expressjs.com>
- [5] MongoDB Inc. (2024). MongoDB Documentation.  
<https://www.mongodb.com/docs>
- [6] Automattic Inc. (2024). Mongoose Documentation.  
<https://mongoosejs.com/docs>
- [7] Tailwind Labs. (2024). Tailwind CSS Documentation.  
<https://tailwindcss.com/docs>
- [8] Auth0 Inc. (2024). JSON Web Token Introduction. <https://jwt.io/introduction>
- [9] Microsoft. (2024). TypeScript Documentation.  
<https://www.typescriptlang.org/docs>



## **PHỤ LỤC**