

NHẬN XÉT CỦA THÀNH VIÊN HỘI ĐỒNG

This image shows a full page of a document template designed for handwriting practice or general writing. It consists of approximately 28 evenly spaced horizontal dotted lines across the entire width of the page. The background is plain white, and there are no margins, headers, footers, or other markings present.

Vĩnh Long, ngày tháng năm

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

LỜI CẢM ƠN

MỤC LỤC

MỞ ĐẦU.....	10
1. Lý do chọn đề tài.....	10
2. Mục tiêu nghiên cứu.....	10
3. Đối tượng nghiên cứu.....	10
4. Phạm vi nghiên cứu.....	11
CHƯƠNG 1: TỔNG QUAN.....	12
1.1. Giới thiệu.....	12
1.2. Tình hình thực tế.....	12
CHƯƠNG 2. NGHIÊN CỨU LÝ THUYẾT.....	13
2.1. Tổng quan về ứng dụng Web hiện đại.....	13
2.1.1. Kiến trúc Client-Server trong ứng dụng web.....	13
2.1.2. Single Page Application (SPA).....	14
2.2. Lập trình Component-base.....	14
2.2.1. Khái niệm Component.....	14
2.2.2. Ưu điểm của Component-Based.....	15
2.3. Quản lý trạng thái trong ứng dụng web.....	16
2.3.1. Khái niệm State Management.....	16
2.3.2. Context API trong React.....	16
2.3.3. Ưu điểm, hạn chế của Context API.....	17
2.4. TypeScript và Type safety.....	18
2.4.1. Giới thiệu TypeScript.....	18
2.4.2. Hệ thống kiểu trong TypeScript.....	18
2.4.3. Lợi ích của TypeScript trong dự án thực tế.....	20
2.5. Kiến trúc của RESTful API.....	20
2.5.1. Khái niệm REST.....	20

2.5.2. Thiết kế RESTful API.....	21
CHƯƠNG 3. HIỆN THỰC HÓA NGHIÊN CỨU	22
3.1. Phân tích bài toán	22
3.1.1. Bối cảnh và động cơ nghiên cứu.....	22
3.1.2. Các vấn đề cốt lõi cần giải quyết	22
3.2. Phân tích yêu cầu.....	23
3.2.1. Yêu cầu chức năng.....	23
3.2.2. Yêu cầu phi chức năng.....	27
3.3. Thiết kế giao diện	28
3.3.1. Cấu trúc hệ thống.....	28
3.3.2. Thiết kế các màn hình chính	29
3.4. Thiết kế dữ liệu.....	36
3.4.1. Mô hình ERD	36
3.4.2. Mô hình Use Case.....	43
3.4.3. Thiết kế xử lý	46
CHƯƠNG 4. KẾT QUẢ NGHIÊN CỨU	47
4.1. Dữ liệu thực nghiệm	47
4.2. Kết quả các chức năng chính.....	47
4.2.1. Chức năng Xác thực và Đăng nhập	47
4.2.2. Chức năng Quản lý Hồ sơ Gia sư	48
4.2.3. Chức năng Tìm kiếm và Lọc Gia sư.....	48
4.2.4. Chức năng Dashboard.....	49
CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.....	51
5.1. Kết luận	51
5.1.1. Kết quả đạt được	51
5.1.2. Đóng góp mới	51

5.1.3. Hạn chế	52
5.2. Hướng phát triển.....	52
DANH MỤC TÀI LIỆU THAM KHẢO	55
PHỤ LỤC.....	56

DANH MỤC HÌNH ẢNH

Hình 2.1 Mô hình Client-Server.....	13
Hình 2.2 Mô hình Provider – Consumer trong Context API	17
Hình 3.1 Sơ đồ cấu trúc hệ thống.....	28
Hình 3.2 Màn hình trang chủ	29
Hình 3.3 Màn hình Đăng nhập/Đăng ký	30
Hình 3.4 Màn hình danh sách gia sư.....	31
Hình 3.5 Màn hình chi tiết gia sư.....	32
Hình 3.6 Màn hình dashboard phụ huynh.....	33
Hình 3.7 Màn hình Dashboard Gia sư	34
Hình 3.8 Màn hình Dashboard Admin.....	35
Hình 3.9 Mô hình ERD	36
Hình 3.10 Mô hình UseCase	43
Hình 3.11 Mô hình DFD mức 0	46
Hình 3.12 Mô hình DFD mức 1	46

DANH MỤC BẢNG BIỂU

Bảng 2.1 So sánh SPA và Multi-Page Application (MPA)	14
Bảng 2.2 Bảng mô tả HTTP Methods	21
Bảng 3.1 Danh sách các thực thể và mối kết hợp	37
Bảng 3.2 Thực thể người_dùng	38
Bảng 3.3 Thực thể họ_sơ_sinh_vien	38
Bảng 3.4 Thực thể họ_sơ_gia_sư	39
Bảng 3.5 Thực thể môn_học	40
Bảng 3.6 Thực thể gia_sư_môn_học	40
Bảng 3.7 Thực thể lịch_day	41
Bảng 3.8 Thực thể bài_dạy	41
Bảng 3.9 Thực thể đơn_ung_tuyen	42
Bảng 3.10 Thực thể danh_gia	42
Bảng 3.11 Tổng quan các Use Case	44
Bảng 3.12 Mô tả chi tiết Use Case chính	44
Bảng 4.1 Tổng quan dữ liệu thực nghiệm	47
Bảng 4.2 Kết quả test chức năng xác thực và đăng nhập	47
Bảng 4.3 Kết quả test chức năng Quản lý Hồ sơ Gia sư	48
Bảng 4.4 Chức năng Tìm kiếm và Lọc Gia sư	49
Bảng 4.5 Kết quả kiểm thử chức năng Dashboard	49

TÓM TẮT ĐỒ ÁN CHUYÊN NGÀNH

Đề tài tập trung giải quyết vấn đề thiếu nền tảng kết nối uy tín giữa sinh viên Đại học Trà Vinh và phụ huynh có nhu cầu tìm gia sư cho học sinh từ lớp 1 đến lớp 12. Việc không thể xác minh danh tính gia sư gây ra rủi ro về chất lượng giảng dạy, trong khi sinh viên lại gặp khó khăn trong việc tiếp cận các cơ hội dạy kèm phù hợp.

Hệ thống được xây dựng dựa trên kiến trúc Three-tier Architecture gồm Frontend (React + TypeScript), Backend (Node.js + Express) và Database (MySQL). Quy trình xác thực MSSV được triển khai theo mô hình kiểm duyệt thủ công, đảm bảo gia sư được xác minh trước khi hoạt động. Hệ thống áp dụng phân quyền dựa trên vai trò (RBAC) cho ba nhóm người dùng: Phụ huynh, Gia sư và Quản trị viên.

Cơ sở dữ liệu gồm 12 bảng với các ràng buộc toàn vẹn và quan hệ chặt chẽ giữa người dùng, hồ sơ sinh viên, hồ sơ gia sư. Giao diện người dùng được thiết kế theo hướng Desktop-first, hỗ trợ quản lý bài đăng, ứng tuyển, chat realtime và thông báo đẩy. Quản trị viên có dashboard riêng để phê duyệt tài khoản và giám sát hoạt động.

Kết quả đạt được cho thấy hệ thống đã hoàn thiện giao diện, mô hình dữ liệu, quy trình xác minh và 15 trường hợp kiểm thử cốt lõi. Hệ thống đáp ứng các yêu cầu ban đầu và sẵn sàng bước sang giai đoạn tích hợp API và triển khai thực tế.

MỞ ĐẦU

1. Lý do chọn đề tài

Hiện nay, nhu cầu tìm kiếm gia sư cho học sinh từ lớp 1 đến lớp 12 ngày càng tăng, đặc biệt là tại các khu vực đô thị và các tỉnh có nhiều trường đại học như Trà Vinh. Tuy nhiên, quá trình kết nối giữa phụ huynh và gia sư phần lớn vẫn diễn ra thủ công thông qua mạng xã hội, các nhóm tự phát hoặc giới thiệu cá nhân, dẫn đến nhiều hạn chế như: khó xác minh danh tính gia sư, thiếu tính minh bạch, không có hệ thống quản lý hồ sơ và đánh giá uy tín.

Bên cạnh đó, sinh viên Đại học Trà Vinh (TVU) có nhu cầu tìm kiếm công việc làm thêm phù hợp với chuyên môn và thời gian. Công việc gia sư là lựa chọn phổ biến nhưng thiếu một nền tảng chuyên nghiệp để sinh viên thể hiện năng lực, xây dựng hồ sơ và tiếp cận phụ huynh.

Vì vậy, việc thiết kế và xây dựng một nền tảng tìm kiếm – kết nối gia sư dành riêng cho sinh viên TVU và phụ huynh là cần thiết, vừa hỗ trợ sinh viên tìm việc làm thêm, vừa giúp phụ huynh tiếp cận gia sư uy tín có xác minh danh tính. Đây cũng là một bài toán thực tế có ý nghĩa xã hội và phù hợp với xu hướng công nghệ hiện nay.

2. Mục tiêu nghiên cứu

- Xây dựng một hệ thống kết nối gia sư – phụ huynh rõ ràng, minh bạch và dễ sử dụng.
- Phát triển cơ chế xác thực danh tính sinh viên thông qua MSSV nhằm tăng độ tin cậy của gia sư.
- Thiết kế giao diện thân thiện, tối ưu trải nghiệm cho phụ huynh và gia sư.
- Xây dựng cơ sở dữ liệu và quy trình quản lý thông tin đồng bộ, hỗ trợ tìm kiếm và ghép nối nhanh chóng.
- Tạo công cụ để phụ huynh có thể đăng nhu cầu học, còn gia sư có thể ứng tuyển và xây dựng hồ sơ năng lực.

3. Đối tượng nghiên cứu

- Quy trình đăng ký, xác minh và quản lý gia sư.
- Tương tác giữa phụ huynh và gia sư thông qua hệ thống.

- Mô hình hoạt động và phân quyền xử lý của từng loại người dùng (Phụ huynh, Gia sư, Quản trị viên).
- Cơ sở dữ liệu, luồng xử lý, tính năng và giao diện người dùng của nền tảng.

4. Phạm vi nghiên cứu

Phạm vi của đề tài được giới hạn như sau:

Hệ thống chỉ phục vụ việc kết nối phụ huynh với gia sư là sinh viên Đại học Trà Vinh.

Chỉ tập trung vào các nghiệp vụ chính: đăng ký tài khoản, xác minh MSSV, đăng bài tìm gia sư, ứng tuyển, xem hồ sơ, và quản lý dữ liệu.

Tính năng giao tiếp nâng cao (video call, thanh toán trực tuyến, AI gợi ý) chưa triển khai trong phạm vi đồ án.

Hệ thống được xây dựng ở mức độ prototype hoàn chỉnh (chạy mô phỏng hoặc có thể demo thực tế), chưa triển khai thương mại.

CHƯƠNG 1: TỔNG QUAN

1.1. Giới thiệu

Trong bối cảnh công nghệ số ngày càng phát triển, nhu cầu sử dụng các nền tảng trực tuyến để kết nối giữa người có nhu cầu và người cung cấp dịch vụ trở nên phổ biến. Đối với lĩnh vực giáo dục, nhu cầu tìm kiếm gia sư cho học sinh từ lớp 1 đến lớp 12 có xu hướng tăng mạnh, đặc biệt tại các khu vực có đông sinh viên. Tuy nhiên, đa số phụ huynh vẫn gặp nhiều khó khăn khi lựa chọn gia sư do thiếu thông tin xác thực và không có công cụ đánh giá chất lượng đáng tin cậy. Ngược lại, sinh viên, đối tượng có nhu cầu tìm việc làm thêm lại thiếu một nền tảng chính thống để tiếp cận phụ huynh.

Trong bối cảnh đó, việc xây dựng một nền tảng trực tuyến hỗ trợ tìm kiếm và kết nối gia sư – phụ huynh một cách chuyên nghiệp, minh bạch và có quy trình xác minh rõ ràng là cần thiết. Đề tài hướng đến việc giải quyết các bất cập của mô hình kết nối truyền thống và áp dụng công nghệ để tối ưu hóa trải nghiệm người dùng.

1.2. Tình hình thực tế

Hiện nay, việc tìm kiếm gia sư chủ yếu diễn ra thông qua các kênh như:

- Nhóm mạng xã hội (Facebook, Zalo): thông tin rời rạc, không kiểm chứng.
- Trung tâm gia sư: chi phí cao, nhiều nơi thiếu minh bạch.
- Giới thiệu cá nhân: phụ thuộc vào mối quan hệ và không đảm bảo chất lượng.

Những hình thức này tiềm ẩn rủi ro về uy tín của gia sư, độ phù hợp chuyên môn và tính bảo mật thông tin. Đồng thời, sinh viên cũng gặp khó khăn khi không có nơi thể hiện kỹ năng, kinh nghiệm hoặc xây dựng hồ sơ cá nhân độc lập.

Từ góc độ giáo dục và xã hội, một nền tảng hỗ trợ sinh viên TVU tìm việc làm thêm đúng năng lực vừa giúp cải thiện thu nhập, vừa tạo cơ hội rèn luyện kỹ năng mềm. Điều này cũng hỗ trợ phụ huynh tiếp cận nguồn gia sư uy tín, có xác minh danh tính rõ ràng.

CHƯƠNG 2. NGHIÊN CỨU LÝ THUYẾT

2.1. Tổng quan về ứng dụng Web hiện đại

2.1.1. Kiến trúc Client-Server trong ứng dụng web

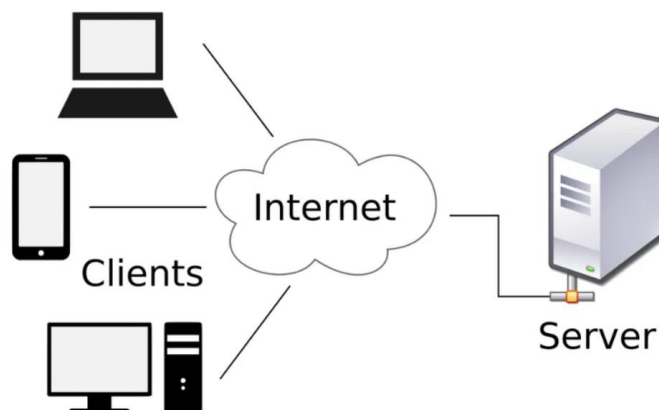
Kiến trúc Client-Server là mô hình phân tán trong đó các nhiệm vụ được chia sẻ giữa các nhà cung cấp tài nguyên (server) và người yêu cầu dịch vụ (client). Trong bối cảnh phát triển ứng dụng web, mô hình này đóng vai trò nền tảng cho hầu hết các hệ thống hiện đại.

Đặc điểm của kiến trúc Client-Server:

- Phân tách trách nhiệm rõ ràng: Client chịu trách nhiệm về giao diện người dùng và logic hiển thị, trong khi server xử lý logic nghiệp vụ và quản lý dữ liệu.
- Giao tiếp qua giao thức chuẩn: Client và server trao đổi thông tin thông qua các giao thức Internet như HTTP/HTTPS, đảm bảo tính tương thích và khả năng mở rộng.
- Tính độc lập về công nghệ: Client và server có thể được phát triển bằng các công nghệ khác nhau mà không ảnh hưởng đến nhau, miễn là tuân thủ giao thức giao tiếp đã định nghĩa.

Ưu điểm của mô hình:

- Bảo mật nâng cao: Dữ liệu nhạy cảm được lưu trữ tập trung tại server, giảm thiểu nguy cơ rò rỉ thông tin.
- Quản lý tập trung: Việc cập nhật, bảo trì hệ thống được thực hiện tại server mà không yêu cầu can thiệp vào từng client.
- Khả năng mở rộng: Có thể tăng cường năng lực xử lý của server (vertical scaling) hoặc thêm nhiều server (horizontal scaling) khi lượng người dùng tăng cao.



Hình 2.1 Mô hình Client-Server

2.1.2. Single Page Application (SPA)

Single Page Application là mô hình ứng dụng web trong đó toàn bộ nội dung được tải trong một trang HTML duy nhất và các tương tác tiếp theo được xử lý động thông qua JavaScript mà không cần tải lại trang.

Đặc điểm kỹ thuật của SPA:

- Client-side Routing: Điều hướng giữa các "trang" được thực hiện hoàn toàn trên trình duyệt thông qua JavaScript, không gửi yêu cầu HTTP mới đến server.
- Dynamic Content Loading: Nội dung được cập nhật bằng cách gọi API và thay đổi DOM (Document Object Model) thay vì tải lại toàn bộ trang.
- State Management: Trạng thái ứng dụng được quản lý trong bộ nhớ của trình duyệt, cho phép chuyển đổi nhanh chóng giữa các view.

Bảng 2.1 So sánh SPA và Multi-Page Application (MPA)

Tiêu chí	SPA	MPA
Tốc độ chuyển trang	Rất nhanh (không reload)	Chậm hơn (reload toàn bộ)
SEO	Khó khăn hơn (cần SSR)	Tốt (HTML sẵn có)
Initial Load	Chậm (tải nhiều JS)	Nhanh (chỉ tải trang đầu)
User Experience	Mượt mà, như native app	Giật lag khi chuyển trang
Độ phức tạp	Cao (cần framework)	Thấp (HTML truyền thống)

Ứng dụng của SPA:

- Ứng dụng tương tác cao (dashboard, admin panel)
- Hệ thống thời gian thực (chat, collaboration tools)
- Ứng dụng giống native mobile app

2.2. Lập trình Component-base

2.2.1. Khái niệm Component

Component-Based Development (CBD) là phương pháp phát triển phần mềm trong đó hệ thống được xây dựng từ các thành phần độc lập, có thể tái sử dụng. Trong ngữ cảnh phát triển giao diện người dùng, component là đơn vị đóng gói bao gồm cả logic xử lý và giao diện hiển thị.

Đặc điểm của component:

- Encapsulation (Đóng gói): Component ẩn đi chi tiết triển khai bên trong, chỉ expose các interface cần thiết thông qua props.

- Reusability (Tái sử dụng): Một component được thiết kế tốt có thể sử dụng ở nhiều nơi khác nhau trong ứng dụng hoặc thậm chí giữa các dự án.

- Composability (Khả năng kết hợp): Các component nhỏ có thể được kết hợp để tạo thành component lớn hơn, phức tạp hơn.

- Independence (Độc lập): Mỗi component hoạt động độc lập, giảm thiểu phụ thuộc lẫn nhau (loose coupling).

Phân loại component:

Trong thực tế phát triển, component được phân loại theo chức năng:

1. Presentational Components: Chỉ chịu trách nhiệm hiển thị UI, nhận dữ liệu qua props, không chứa logic nghiệp vụ. Ví dụ: Button, Input, Card.

2. Container Components: Chứa logic nghiệp vụ, quản lý state, gọi API, sau đó truyền dữ liệu xuống presentational components.

3. Layout Components: Định nghĩa cấu trúc bố cục chung như Header, Sidebar, Footer.

4. Higher-Order Components (HOC): Component nhận component khác làm input và trả về component mới với các tính năng bổ sung.

2.2.2. Ưu điểm của Component-Based

Về khía cạnh kỹ thuật:

- Maintainability (Dễ bảo trì): Khi cần sửa đổi, chỉ cần tập trung vào component liên quan mà không ảnh hưởng toàn bộ hệ thống.

- Testability (Dễ kiểm thử): Mỗi component có thể được test độc lập với input/output cụ thể, tăng độ tin cậy của hệ thống.

- Scalability (Khả năng mở rộng): Dễ dàng thêm component mới hoặc thay thế component cũ mà không cần refactor toàn bộ codebase.

Về khía cạnh quản lý dự án:

- Parallel Development: Nhiều developer có thể làm việc đồng thời trên các component khác nhau mà không xung đột.

- Code Reusability: Giảm thời gian phát triển và số lượng bug nhờ tái sử dụng component đã được kiểm chứng.

2.3. Quản lý trạng thái trong ứng dụng web

2.3.1. Khái niệm State Management

State là dữ liệu động trong ứng dụng, thay đổi theo thời gian dựa trên hành động của người dùng hoặc các sự kiện hệ thống. State Management là việc quản lý và đồng bộ hóa các trạng thái này xuyên suốt ứng dụng.

Phân loại State:

1. Local State: Trạng thái chỉ sử dụng trong một component cụ thể. Ví dụ: giá trị input form, trạng thái toggle button.
2. Global State: Trạng thái cần truy cập từ nhiều component khác nhau. Ví dụ: thông tin người dùng đã đăng nhập, theme (dark/light mode).
3. Server State: Dữ liệu đồng bộ từ server, có tính chất cache. Ví dụ: danh sách bài đăng, thông tin gia sư.
4. URL State: Trạng thái được lưu trong URL query parameters hoặc route params. Ví dụ: bộ lọc tìm kiếm.

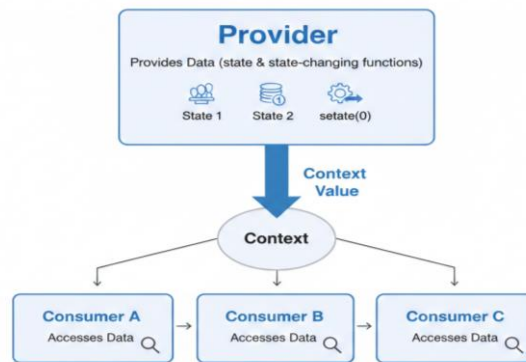
2.3.2. Context API trong React

Context API là cơ chế tích hợp sẵn (built-in) trong React, cho phép truyền dữ liệu xuyên suốt cây component mà không cần phải truyền props qua từng cấp độ trung gian.

Nguyên lý hoạt động:

Context API hoạt động dựa trên pattern Provider-Consumer:

1. Provider: Component cung cấp dữ liệu (state và các hàm thay đổi state).
2. Consumer: Các component con có thể truy cập dữ liệu từ Provider gần nhất.



Hình 2.2 Mô hình Provider – Consumer trong Context API

Khi nào nên sử dụng Context API:

- Dữ liệu cần truy cập từ nhiều component ở các cấp độ khác nhau
- Muốn tránh "prop drilling" (truyền props qua nhiều cấp trung gian)
- Các global state đơn giản như theme, locale, authenticated user

Ví dụ minh họa cấu trúc AuthContext:

```
// Định nghĩa interface cho authentication context
interface AuthContextType {
  user: User | null           // Thông tin người dùng
  isAuthenticated: boolean    // Trạng thái đăng nhập
  login: (credentials) => void // Hàm đăng nhập
  logout: () => void           // Hàm đăng xuất
}

// Tạo context với giá trị mặc định undefined
Const AuthContext = createContext<AuthContextType>
undefined>(undefined)
```

2.3.3. Ưu điểm, hạn chế của Context API

Ưu điểm:

- Tích hợp sẵn trong React, không cần cài thêm thư viện
- Giao diện lập trình (API) đơn giản, dễ học, phù hợp với ứng dụng nhỏ đến trung bình
- Hỗ trợ TypeScript support tốt

Hạn chế:

- Giảm hiệu năng nếu giá trị của context thay đổi thường xuyên
- Không có công cụ dành cho lập trình viên (devtools) như Redux

-
- Thiếu mẫu kiến trúc middleware để xử lý tác vụ bất đồng bộ

2.4. TypeScript và Type safety

2.4.1. Giới thiệu TypeScript

TypeScript là một superset của JavaScript được phát triển bởi Microsoft, bổ sung tính năng static typing (kiểu tĩnh) vào ngôn ngữ JavaScript động. Anders Hejlsberg, kiến trúc sư chính của TypeScript, mô tả nó như một "JavaScript that scales" - JavaScript có khả năng mở rộng cho các dự án lớn.

Động cơ ra đời của TypeScript:

JavaScript là ngôn ngữ kiểu động (dynamically typed), nghĩa là kiểu dữ liệu của biến chỉ được xác định khi chương trình chạy (runtime). Điều này dẫn đến:

- Khó phát hiện lỗi sai kiểu dữ liệu (type mismatch) trong quá trình phát triển
- Thiếu hỗ trợ gợi ý mã (IntelliSense) và tự hoàn thành (autocomplete) trong trình soạn thảo hoặc IDE

- Khó tái cấu trúc mã (refactor code) trong các dự án lớn
- Tài liệu (documentation) dễ thiếu chính xác hoặc không đồng bộ

TypeScript giải quyết những vấn đề này bằng cách:

- Kiểm tra kiểu khi biên dịch (compile-time type checking): phát hiện lỗi trước khi chạy chương trình
- Hỗ trợ IDE nâng cao: tự hoàn thành, điều hướng đến định nghĩa (go-to-definition), công cụ tái cấu trúc mã
- Mã tự mô tả (self-documenting code): các kiểu dữ liệu chính là tài liệu
- Cộng tác tốt hơn: các thành viên trong nhóm hiểu rõ cấu trúc dữ liệu và hợp đồng API

2.4.2. Hệ thống kiểu trong TypeScript

2.4.2.1 Kiểu nguyên thủy (Primitive Types)

TypeScript hỗ trợ các kiểu dữ liệu cơ bản của JavaScript nhưng bổ sung kiểu tĩnh (static typing), giúp phát hiện lỗi ngay khi viết mã.

Ví dụ:

```
let id: number = 123
let email: string = "user@example.com"
let isActive: boolean = true
let tags: string[] = ["react", "typescript"]
```

2.4.2.2 Interface và Type Aliases

Interface dùng để định nghĩa cấu trúc của một đối tượng, đảm bảo tính thống nhất trong toàn bộ ứng dụng.

Ví dụ:

```
// Interface cho User entity
interface User {
  id: string
  email: string
  name: string
  role: 'phu_huynh' | 'gia_su' | 'admin' // Union type
  createdAt: Date
  phone?: string // Optional property
}
```

2.4.2.3 Union Types và Literal Types

Union types cho phép một biến có thể nhận nhiều kiểu khác nhau. Literal types giới hạn biến vào một số giá trị cố định.

Ví dụ:

```
type UserRole = 'phu_huynh' | 'gia_su' | 'admin' // Chỉ 3
giá trị hợp lệ
type Status = 'pending' | 'approved' | 'rejected'
```

2.4.2.4 Generic Types

Generics cho phép viết mã tái sử dụng, áp dụng được cho nhiều kiểu dữ liệu khác nhau.

Ví dụ:

```
interface ApiResponse<T> {
  success: boolean
  data?: T
  error?: string
}

// Sử dụng với kiểu cụ thể
```

```
const userResponse: ApiResponse<User> = await fetchUser()  
const tutorsResponse: ApiResponse<Tutor[]> = await  
fetchTutors()
```

2.4.3. Lợi ích của TypeScript trong dự án thực tế

- Giảm khoảng 15% số lượng lỗi: Nhờ cơ chế kiểm tra kiểu (type checking) giúp phát hiện sai sót ngay từ sớm.
- Tăng khoảng 20% năng suất làm việc: Do có tính năng gợi ý mã (autocomplete), điều hướng, và công cụ tái cấu trúc mã (refactoring tools).
- Cải thiện chất lượng mã nguồn: Buộc lập trình viên phải suy nghĩ rõ ràng về kiểu dữ liệu và cấu trúc hệ thống.

2.5. Kiến trúc của RESTful API

2.5.1. Khái niệm REST

REST (Representational State Transfer) là một phong cách kiến trúc (architectural style) cho các hệ thống phân tán, được Roy Fielding đề xuất trong luận án tiến sĩ năm 2000. REST không phải là một giao thức, mà là một tập hợp các ràng buộc kiến trúc nhằm xây dựng hệ thống dễ mở rộng, nhất quán và hiệu quả.

Những ràng buộc của REST:

1. Client–Server: Tách biệt phần giao diện người dùng và phần lưu trữ dữ liệu để tăng khả năng mở rộng.
2. Stateless: Mỗi request từ phía client phải tự chứa đủ thông tin để server xử lý; server không lưu trạng thái phiên (session state).
3. Cacheable: Mỗi response phải cho biết có được phép cache hay không nhằm tối ưu hiệu năng.
4. Uniform Interface: Chuẩn hoá cách giao tiếp giữa client và server giúp hệ thống đơn giản, dễ hiểu và dễ mở rộng.
5. Layered System: Kiến trúc có thể có thêm các lớp trung gian như load balancer, proxy, cache... mà client không cần biết.
6. Code on Demand (tùy chọn): Server có thể gửi mã thực thi (như JavaScript) cho client nếu cần.

2.5.2. Thiết kế RESTful API

Trong kiến trúc RESTful, các phương thức HTTP được sử dụng để biểu diễn các thao tác CRUD trên tài nguyên. Mỗi phương thức mang một ngữ nghĩa riêng, giúp chuẩn hoá cách thức mà client tương tác với server. Cụ thể:

GET: Thực hiện thao tác truy xuất dữ liệu (Read). Đây là phương thức idempotent và safe, nghĩa là việc gửi lặp lại nhiều lần không làm thay đổi trạng thái của hệ thống.

POST: Dùng để tạo mới một tài nguyên (Create). Phương thức này không mang tính idempotent, do mỗi lần thực hiện có thể tạo ra một thực thể mới.

PUT: Thực hiện cập nhật toàn bộ một tài nguyên (Update). Phương thức này là idempotent, đảm bảo kết quả không thay đổi dù được gọi nhiều lần.

PATCH: Cập nhật một phần của tài nguyên. Tương tự PUT, phương thức này cũng idempotent.

DELETE: Xóa một tài nguyên cụ thể. Đây là phương thức idempotent, do việc xóa lặp lại không làm thay đổi kết quả cuối cùng.

Bảng 2.2 Bảng mô tả HTTP Methods

HTTP Method	Hành động	Đặc tính	Mô tả
GET	Read	Idempotent, Safe	Lấy dữ liệu
POST	Create	Non-idempotent	Tạo mới resource
PUT	Update (toàn bộ)	Idempotent	Ghi đè toàn bộ resource
PATCH	Update (một phần)	Idempotent	Cập nhật một phần resource
DELETE	Delete	Idempotent	Xóa resource

CHƯƠNG 3. HIỆN THỰC HÓA NGHIÊN CỨU

3.1. Phân tích bài toán

3.1.1. Bối cảnh và động cơ nghiên cứu

Trong những năm gần đây, nhu cầu học thêm và tìm kiếm gia sư tại Việt Nam tăng mạnh do áp lực học tập và xu hướng cá thể hóa quá trình học của học sinh. Theo các khảo sát giáo dục, ước tính có khoảng 70% học sinh THPT tại các đô thị lớn tham gia các hình thức học thêm. Tuy nhiên, thị trường gia sư hiện nay vẫn tồn tại nhiều bất cập liên quan đến tính minh bạch thông tin, sự an toàn khi kết nối và chất lượng dịch vụ.

Đối với phụ huynh, việc tìm kiếm một gia sư phù hợp gặp nhiều khó khăn do thiếu thông tin xác thực về trình độ, kinh nghiệm và danh tính của người dạy. Các kênh tìm kiếm phổ biến như mạng xã hội hoặc giới thiệu cá nhân thiếu tính kiểm chứng và dễ tiềm ẩn rủi ro.

Đối với sinh viên, đặc biệt là sinh viên Trường Đại học Trà Vinh (TVU), nhu cầu tìm kiếm việc làm thêm phù hợp với chuyên môn và an toàn cũng ngày càng lớn. Tuy nhiên, cơ hội tiếp cận phụ huynh bị hạn chế do không có nền tảng tập trung, trong khi các trung tâm gia sư truyền thống thường thu phí cao hoặc thiếu minh bạch.

Thực trạng này cho thấy cần thiết phải xây dựng một hệ thống trực tuyến có khả năng kết nối trực tiếp, minh bạch và hiệu quả giữa phụ huynh và sinh viên TVU.

3.1.2. Các vấn đề cốt lõi cần giải quyết

Từ bối cảnh thực tế, có thể nhận diện các vấn đề chính sau:

1. Thiếu cơ chế xác thực danh tính và trình độ gia sư: Phụ huynh không có công cụ kiểm chứng sinh viên có đúng là đang theo học tại TVU hay không.
2. Thông tin giữa phụ huynh và gia sư không minh bạch: Hồ sơ gia sư trên các nền tảng không chính thức thường thiếu chi tiết và không có đánh giá đáng tin cậy.
3. Kênh kết nối phân tán và thiếu hệ thống: Việc trao đổi qua mạng xã hội dễ gây thất lạc thông tin, khó quản lý lịch dạy và không đảm bảo tính an toàn.

4. Thiếu nền tảng hỗ trợ sinh viên tiếp cận công việc một cách chuyên nghiệp: Sinh viên không có nơi xây dựng hồ sơ học thuật, kỹ năng và kinh nghiệm để thể hiện với phụ huynh.

Những vấn đề này đặt ra nhu cầu cấp thiết về một nền tảng chuyên biệt dành riêng cho cộng đồng sinh viên và phụ huynh tại Trà Vinh.

3.2. Phân tích yêu cầu

3.2.1. Yêu cầu chức năng

3.2.1.1 Yêu cầu chức năng dành cho phụ huynh

a) Quản lý tài khoản và hồ sơ cá nhân (RF-PH-01)

Hệ thống cho phép phụ huynh thực hiện các thao tác liên quan đến tài khoản:

- Đăng ký tài khoản mới với thông tin bắt buộc: họ tên, email, số điện thoại, mật khẩu

- Đăng nhập vào hệ thống bằng email và mật khẩu
- Cập nhật thông tin cá nhân (họ tên, địa chỉ, số điện thoại, ảnh đại diện)
- Đổi mật khẩu và quên mật khẩu
- Đăng xuất khỏi hệ thống

b) Đăng bài tìm gia sư (RF-PH-02)

Phụ huynh có thể tạo bài đăng tìm gia sư với các thông tin:

- Môn học cần tìm (lựa chọn từ danh sách môn học có sẵn)
- Trình độ học sinh (Tiểu học, THCS, THPT, Đại học)
- Địa điểm dạy (địa chỉ cụ thể hoặc khu vực)
- Lịch dạy mong muốn (số buổi/tuần, khung giờ cụ thể)
- Mức học phí đề xuất (VNĐ/buổi hoặc VNĐ/tháng)
- Mô tả chi tiết yêu cầu và thông tin học sinh

Hệ thống cho phép chỉnh sửa hoặc xóa bài đăng chưa có gia sư nhận và thay đổi trạng thái bài đăng (Đang tìm/Đã đủ gia sư/Đã đóng).

c) Tìm kiếm và xem thông tin gia sư (RF-PH-03)

Phụ huynh có thể:

- Xem danh sách gia sư đã được xác thực trong hệ thống
- Tìm kiếm gia sư theo tên hoặc từ khóa
- Lọc gia sư theo các tiêu chí: môn học, khu vực giảng dạy, khoảng học phí, điểm đánh giá trung bình

-
- Sắp xếp kết quả theo độ phù hợp, đánh giá cao nhất, hoặc học phí
 - Xem chi tiết hồ sơ gia sư bao gồm: thông tin cá nhân, trình độ học vấn, kinh nghiệm, các môn có thể dạy, lịch rảnh, đánh giá từ phụ huynh khác

d) Quản lý đơn ứng tuyển (RF-PH-04)

Đối với mỗi bài đăng, phụ huynh có thể:

- Xem danh sách các đơn ứng tuyển từ gia sư
- Xem thông tin chi tiết của từng ứng viên (hồ sơ, kinh nghiệm, đánh giá)
- Chấp nhận hoặc từ chối đơn ứng tuyển
- Gửi tin nhắn trao đổi với ứng viên trước khi quyết định
- Sau khi chấp nhận, hệ thống tự động cập nhật trạng thái bài đăng và thông báo cho gia sư.

e) Đánh giá gia sư (RF-PH-05)

Sau khi kết thúc khóa học, phụ huynh được yêu cầu đánh giá gia sư:

- Cho điểm từ 1 đến 5 sao
- Viết nhận xét chi tiết về chất lượng giảng dạy, thái độ, hiệu quả học tập
- Xem lại các đánh giá đã đưa ra trong quá khứ
- Chỉnh sửa đánh giá trong vòng 7 ngày sau khi gửi

f) Trao đổi và nhận thông báo (RF-PH-06)

Hệ thống hỗ trợ:

- Nhắn tin trực tiếp với gia sư thông qua hệ thống chat tích hợp
- Nhận thông báo khi có đơn ứng tuyển mới
- Nhận thông báo khi có tin nhắn mới từ gia sư
- Quản lý và đánh dấu đã đọc, chưa đọc thông báo

3.2.1.2 Yêu cầu chức năng dành cho gia sư

a) Đăng ký và xác thực tài khoản sinh viên (RF-GS-01)

Sinh viên TVU muốn trở thành gia sư phải thực hiện quy trình xác thực:

- Đăng ký tài khoản với email, mật khẩu, và mã số sinh viên (MSSV) 9 chữ số
- Upload hình ảnh thẻ sinh viên TVU còn hiệu lực
- Upload bảng điểm hoặc giấy tờ liên quan (tùy chọn)
- Gửi yêu cầu xác thực đến quản trị viên
- Chờ phê duyệt từ quản trị viên (thường trong vòng 24-48 giờ)

- Chỉ sau khi được xác thực, sinh viên mới có thể tạo hồ sơ gia sư và ứng tuyển công việc.

b) Quản lý hồ sơ gia sư (RF-GS-02)

Sau khi được xác thực, gia sư có thể:

- Tạo và hoàn thiện hồ sơ cá nhân với thông tin:
- Thông tin học vấn: trường, khoa, ngành, năm học, GPA
- Kinh nghiệm dạy học (nếu có)
- Chứng chỉ, giải thưởng học tập
- Giới thiệu bản thân (mô tả ngắn gọn về phong cách dạy học)
- Ảnh chân dung chuyên nghiệp
- Đăng ký các môn học có thể dạy (được chọn từ danh sách hệ thống)
- Thiết lập mức học phí mong muốn cho từng môn
- Cập nhật hồ sơ bất kỳ lúc nào

c) Quản lý lịch dạy (RF-GS-03)

Gia sư có thể:

- Thiết lập lịch rảnh trong tuần (chọn các khung giờ có thể dạy)
- Xem lịch dạy đã nhận theo dạng lịch tuần/tháng
- Cập nhật hoặc thay đổi lịch rảnh khi cần thiết
- Đánh dấu các buổi học đã hoàn thành
- Ghi chú về tiến độ và nội dung đã dạy cho từng buổi học

d) Tìm kiếm và ứng tuyển công việc (RF-GS-04)

Gia sư có thể:

- Xem danh sách các bài đăng tìm gia sư đang mở
- Tìm kiếm và lọc bài đăng theo:
 - + Môn học phù hợp với khả năng
 - + Khu vực gần nơi ở hoặc trường học
 - + Mức học phí chấp nhận được
 - + Lịch dạy không trùng với lịch học
- Nộp đơn ứng tuyển với:
 - + Giới thiệu bản thân và động lực dạy học
 - + Kinh nghiệm liên quan đến môn học/trình độ
 - + Đề xuất học phí (nếu bài đăng cho phép thương lượng)

+ Lịch có thể dạy cụ thể

- Xem trạng thái đơn ứng tuyển (Đang chờ/Được chấp nhận/Bị từ chối)
- Rút đơn ứng tuyển nếu chưa được phản hồi

e) Quản lý công việc đang nhận (RF-GS-05)

Sau khi được phụ huynh chấp nhận, gia sư có thể:

- Xem danh sách học sinh đang dạy
- Xem thông tin chi tiết từng học sinh (địa chỉ, lịch học, yêu cầu đặc biệt)
- Cập nhật tiến độ học tập của học sinh
- Ghi chú về từng buổi học (nội dung đã dạy, bài tập, nhận xét)
- Xem đánh giá và phản hồi từ phụ huynh

f) Trao đổi và nhận thông báo (RF-GS-06)

Hệ thống hỗ trợ:

- Nhắn tin trực tiếp với phụ huynh
- Nhận thông báo khi đơn ứng tuyển được chấp nhận hoặc từ chối
- Nhận thông báo khi có bài đăng mới phù hợp với hồ sơ (dựa trên môn học và khu vực)
- Nhận nhắc lịch dạy sắp tới

3.2.1.3 Yêu cầu chức năng dành cho quản trị viên

a) Quản lý người dùng (RF-AD-01)

Quản trị viên có quyền:

- Xem danh sách tất cả người dùng (phụ huynh và gia sư)
- Tìm kiếm người dùng theo tên, email, số điện thoại, vai trò
- Xem chi tiết thông tin tài khoản và lịch sử hoạt động
- Kích hoạt hoặc vô hiệu hóa tài khoản vi phạm quy định
- Đặt lại mật khẩu cho người dùng khi có yêu cầu

b) Xác thực tài khoản gia sư (RF-AD-02)

Đây là chức năng quan trọng nhất của quản trị viên:

- Xem danh sách yêu cầu xác thực từ sinh viên (sắp xếp theo thời gian gửi)
- Kiểm tra tính hợp lệ của:
 - + Mã số sinh viên (MSSV) - phải đúng định dạng 9 chữ số
 - + Hình ảnh thẻ sinh viên TVU - phải rõ ràng, còn hiệu lực
 - + Thông tin trên thẻ phải khớp với thông tin đăng ký

- + Phê duyệt yêu cầu nếu thông tin hợp lệ

- + Từ chối yêu cầu với lý do cụ thể (MSSV không hợp lệ, ảnh thẻ không rõ, thông tin không khớp)

- Gửi thông báo kết quả đến sinh viên

c) Quản lý nội dung (RF-AD-03)

Quản trị viên giám sát nội dung trên nền tảng:

- Xem danh sách tất cả bài đăng tìm gia sư
- Ẩn hoặc xóa bài đăng vi phạm (nội dung không phù hợp, spam, lừa đảo)

Quản lý danh mục môn học:

- Thêm môn học mới
- Chỉnh sửa tên môn học
- Xóa môn học không còn sử dụng
- Quản lý danh sách khu vực giảng dạy

d) Quản lý đánh giá (RF-AD-04)

Quản trị viên có thể:

- Xem tất cả đánh giá trong hệ thống
- Lọc đánh giá theo gia sư, phụ huynh, điểm số, thời gian
- Ẩn hoặc xóa đánh giá không phù hợp (spam, ngôn từ không phù hợp, giả mạo)

- Xử lý báo cáo từ người dùng về đánh giá gian lận

- Khôi phục đánh giá đã ẩn nếu cần

3.2.2. Yêu cầu phi chức năng

3.2.2.1 Yêu cầu về hiệu năng (NFR-01)

- Thời gian phản hồi trung bình cho các thao tác thông thường ≤ 2 giây
- Thời gian tải trang không quá 3 giây với kết nối Internet tốc độ trung bình
- Hỗ trợ đồng thời tối thiểu 500 người dùng trực tuyến
- Database có khả năng xử lý tối thiểu 10,000 bản ghi/bảng mà không ảnh hưởng hiệu năng

3.2.2.2 Yêu cầu về tính khả dụng (NFR-02)

- Hệ thống hoạt động ổn định 99.5% thời gian (downtime < 1.8 ngày/năm)
- Giao diện responsive, tương thích với các thiết bị: Desktop, Tablet, Mobile

- Hỗ trợ các trình duyệt phổ biến: Chrome, Firefox, Safari, Edge (phiên bản mới nhất)

- Có cơ chế backup dữ liệu tự động hàng ngày

3.2.2.3 Yêu cầu về bảo mật (NFR-03)

- Mật khẩu được mã hóa bằng thuật toán bcrypt trước khi lưu vào database
- Xác thực người dùng bằng JWT (JSON Web Token)
- Phiên đăng nhập hết hạn sau 7 ngày không hoạt động
- Dữ liệu nhạy cảm (MSSV, số điện thoại) được mã hóa khi lưu trữ
- Ngăn chặn các tấn công: SQL Injection, XSS, CSRF
- Giới hạn số lần đăng nhập sai (5 lần/30 phút) để chống brute-force

3.2.2.4 Yêu cầu về tính dễ sử dụng (NFR-04)

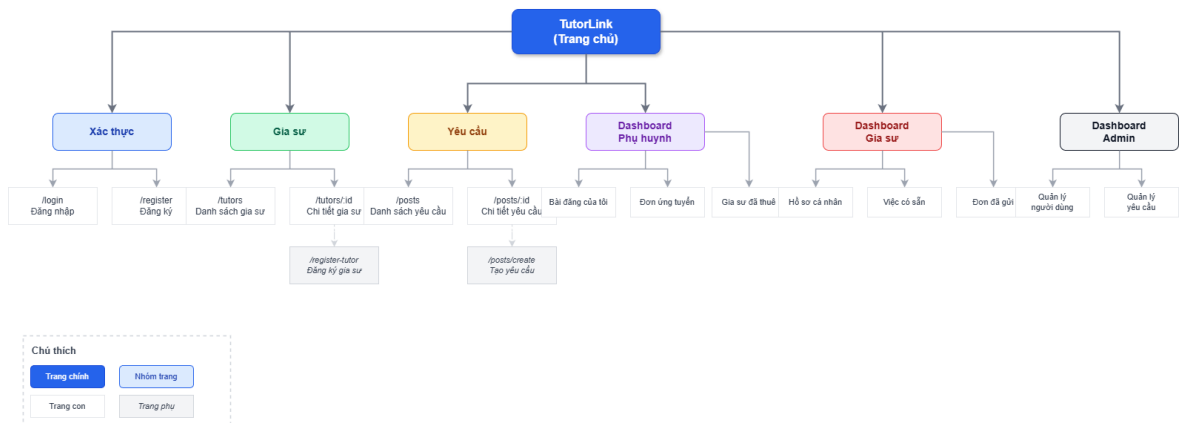
- Giao diện trực quan, dễ hiểu cho người dùng không chuyên
- Có hướng dẫn sử dụng (tooltips, FAQ) cho các chức năng chính
- Thông báo lỗi rõ ràng, hướng dẫn người dùng cách khắc phục
- Form nhập liệu có validation và gợi ý ngay lập tức
- Sử dụng tiếng Việt làm ngôn ngữ chính

3.2.2.5 Yêu cầu về khả năng mở rộng (NFR-05)

- Kiến trúc module hóa, dễ dàng thêm chức năng mới
- Database được thiết kế chuẩn hóa, dễ mở rộng
- Code tuân theo coding convention, dễ bảo trì
- API được thiết kế RESTful, có thể tích hợp với ứng dụng di động sau này

3.3. Thiết kế giao diện

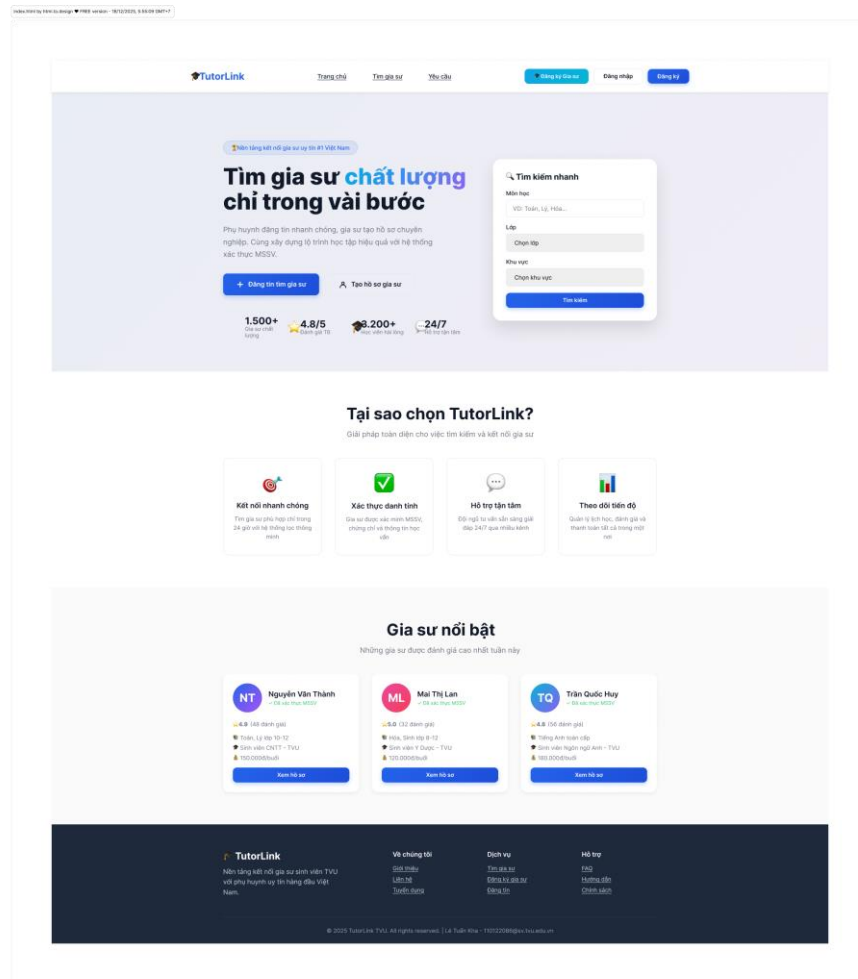
3.3.1. Cấu trúc hệ thống



Hình 3.1 Sơ đồ cấu trúc hệ thống

3.3.2. Thiết kế các màn hình chính

3.3.2.1 Màn hình trang chủ



Hình 3.2 Màn hình trang chủ

Trang chủ là điểm tiếp xúc đầu tiên giữa hệ thống và người dùng, có vai trò giới thiệu tổng quan dịch vụ và định hướng thao tác. Giao diện được thiết kế gồm 05 khu vực chính, sắp xếp theo chiều dọc từ trên xuống dưới nhằm đảm bảo tính trực quan và dễ sử dụng.

Thanh điều hướng (Navigation Bar) được cố định ở đầu trang, bao gồm logo TutorLink, các mục điều hướng (Trang chủ, Tìm gia sư, Yêu cầu) và các nút chức năng Đăng ký Gia sư, Đăng nhập, Đăng ký, giúp người dùng truy cập nhanh vào các chức năng quan trọng.

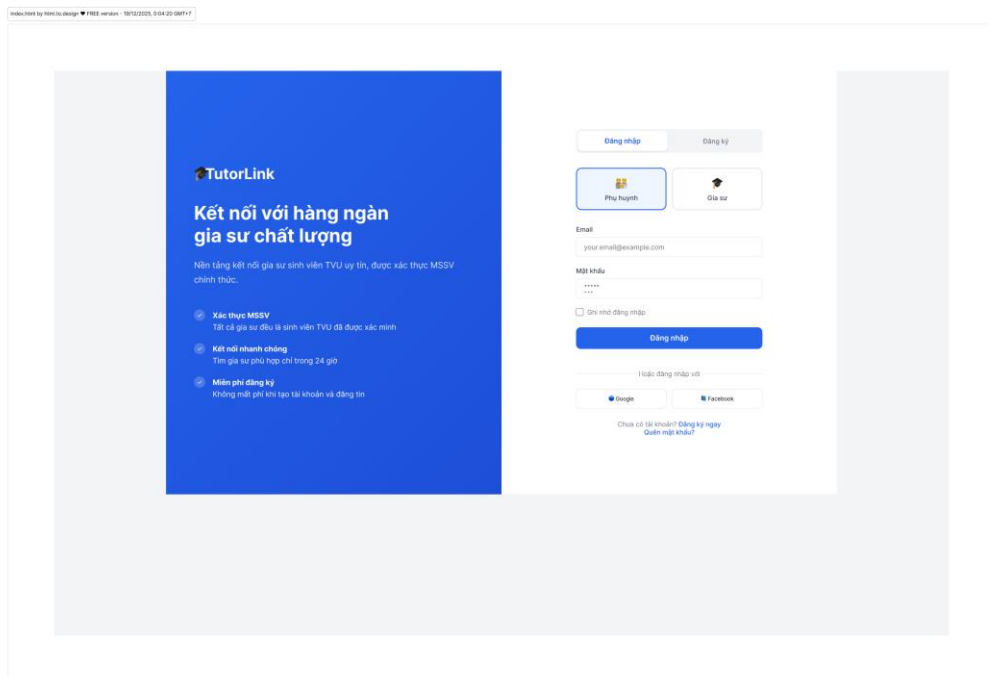
Khu vực giới thiệu chính (Hero Section) sử dụng bố cục lưới hai cột, trong đó cột trái trình bày tiêu đề, mô tả ngắn, các nút kêu gọi hành động và các chỉ số thống kê tiêu biểu. Cột phải là thẻ tìm kiếm nhanh cho phép người dùng tra cứu gia sư theo môn học, lớp học và khu vực.

Khu vực lợi ích (Benefits Section) trình bày các ưu điểm nổi bật của hệ thống dưới dạng lưới bốn cột, mỗi thẻ gồm biểu tượng, tiêu đề và mô tả ngắn gọn, giúp người dùng nhanh chóng nắm bắt giá trị cốt lõi của TutorLink.

Khu vực gia sư nổi bật (Featured Tutors Section) hiển thị ba gia sư tiêu biểu theo bố cục ba cột, mỗi thẻ bao gồm thông tin cơ bản, đánh giá và nút xem hồ sơ, nhằm tăng độ tin cậy và khả năng tương tác của người dùng.

Chân trang (Footer) cung cấp thông tin đơn vị phát triển, các liên kết nhanh và nội dung bản quyền. Toàn bộ giao diện tuân theo hệ thống lưới 12 cột, với chiều rộng tối đa 1120px và sử dụng nền gradient nhẹ cho Hero Section nhằm tạo điểm nhấn thị giác.

3.3.2.2 Màn hình đăng nhập/dăng ký



Hình 3.3 Màn hình Đăng nhập/Đăng ký

Màn hình xác thực được thiết kế theo bố cục chia đôi (split-screen) tỷ lệ 50/50, nhằm tối ưu không gian hiển thị và tạo ấn tượng thị giác ban đầu.

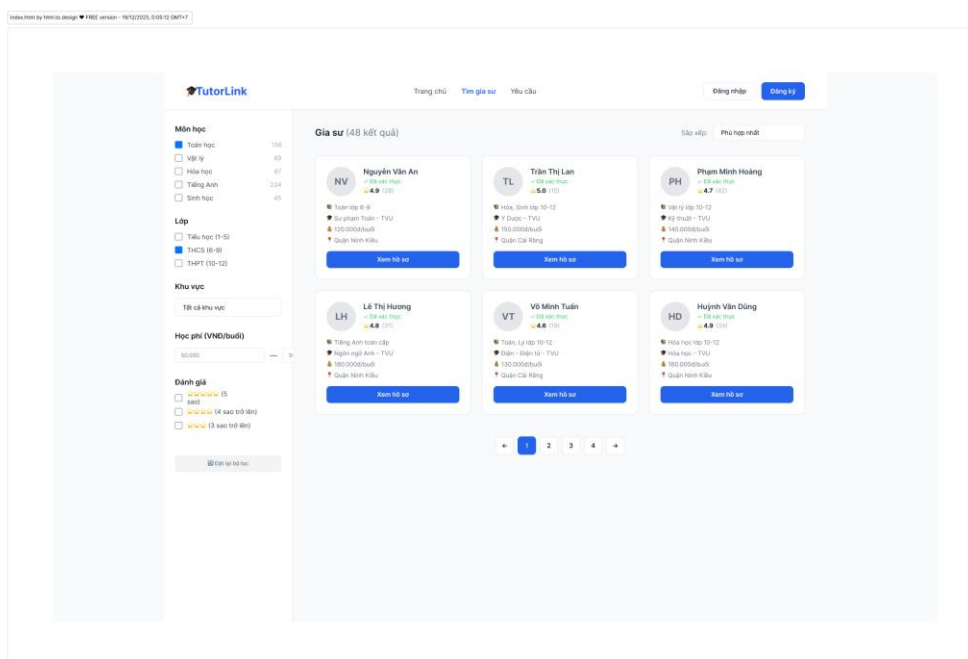
Khu vực bên trái sử dụng nền màu xanh chủ đạo (#2563EB) với chữ màu trắng, hiển thị logo TutorLink, tiêu đề thể hiện giá trị hệ thống, các lợi ích chính dưới dạng gạch đầu dòng và một số chỉ số thống kê tiêu biểu (số lượng gia sư và mức đánh giá trung bình).

Khu vực bên phải có nền trắng, chứa biểu mẫu xác thực với hai tab Đăng nhập và Đăng ký, đồng thời cho phép người dùng lựa chọn vai trò (Phụ huynh hoặc Gia sư). Biểu mẫu đăng nhập bao gồm các trường thông tin cơ bản như email và

mật khẩu, tùy chọn ghi nhớ đăng nhập, nút gửi dữ liệu và liên kết khôi phục mật khẩu.

Khi chuyển sang chế độ đăng ký, biểu mẫu được mở rộng thêm các trường thông tin cá nhân, trong đó trường mã số sinh viên chỉ hiển thị khi người dùng lựa chọn vai trò gia sư nhằm phục vụ mục đích xác thực.

3.3.2.3 Màn hình danh sách gia sư



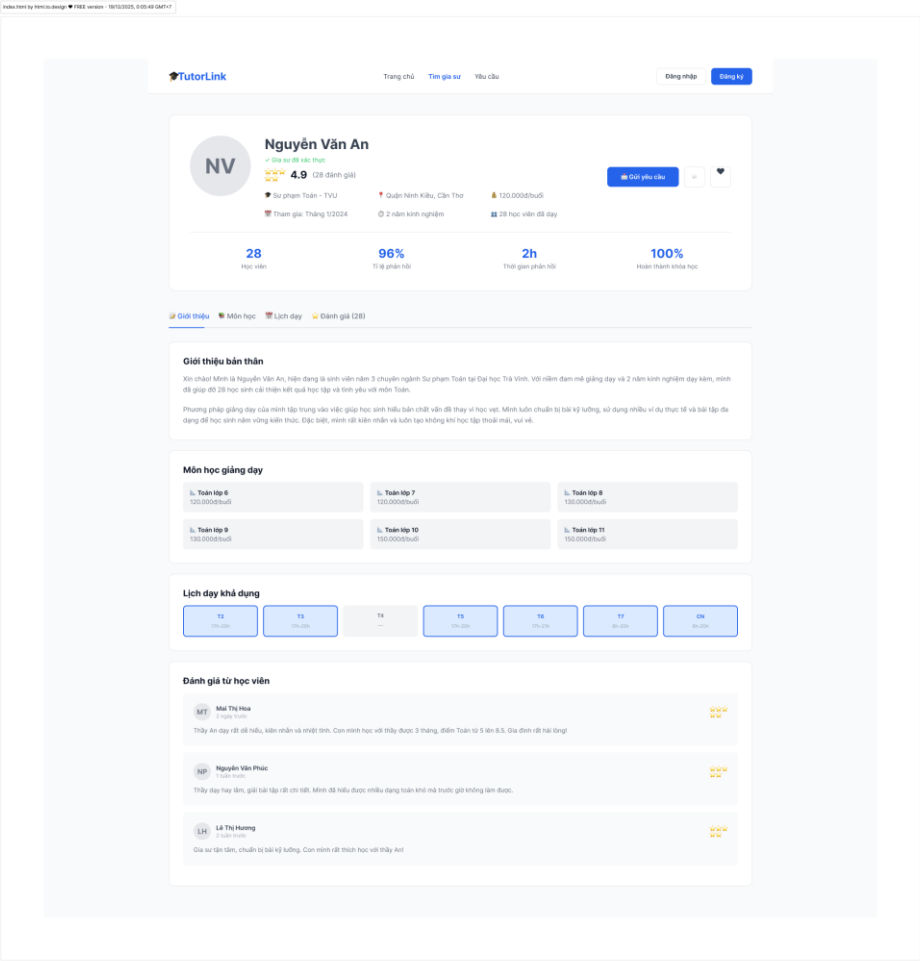
Hình 3.4 Màn hình danh sách gia sư

Màn hình danh sách gia sư bao gồm khu vực bộ lọc và khu vực hiển thị kết quả chính. Ở đầu trang, phần giới thiệu cung cấp nhãn tiêu đề, tiêu đề chính, mô tả ngắn và các thẻ thống kê tổng quan, giúp người dùng nhanh chóng nắm bắt tình trạng dữ liệu trong hệ thống.

Bố cục tổng thể sử dụng lưới hai cột, trong đó sidebar bộ lọc đặt bên trái với chiều rộng cố định và khả năng cố định khi cuộn trang, còn khu vực kết quả nằm bên phải và chiếm phần lớn không gian hiển thị. Sidebar bao gồm các nhóm bộ lọc dạng thu gọn, cho phép người dùng lọc gia sư theo môn học, cấp lớp, mức học phí, khu vực và kinh nghiệm giảng dạy, đồng thời cung cấp các nút thao tác áp dụng và đặt lại bộ lọc.

Khu vực kết quả bắt đầu bằng thanh tiêu đề hiển thị số lượng gia sư tìm thấy và tùy chọn sắp xếp. Danh sách gia sư được trình bày dưới dạng lưới nhiều cột thông qua các thẻ thông tin gia sư, kèm theo thành phần phân trang nhằm hỗ trợ điều hướng giữa các tập dữ liệu lớn.

3.3.2.4 Màn hình chi tiết gia sư



Hình 3.5 Màn hình chi tiết gia sư

Màn hình chi tiết gia sư được tổ chức theo cấu trúc ba tầng, bao gồm Profile Header, Tab Navigation và Content Layout, nhằm trình bày thông tin một cách rõ ràng, có hệ thống và dễ tiếp cận.

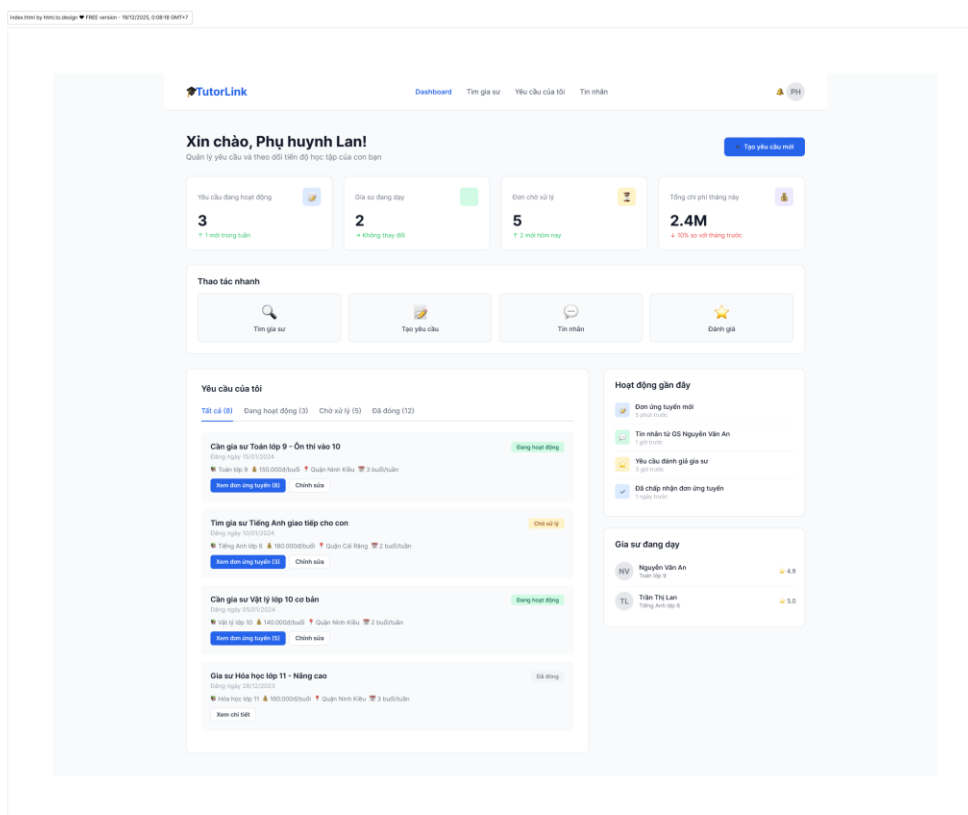
Phần Profile Header hiển thị các thông tin nhận diện chính như ảnh đại diện, tên gia sư, trạng thái xác thực, mức đánh giá, khu vực hoạt động và các nút hành động quan trọng.

Thanh điều hướng theo tab (Tab Navigation) cho phép chuyển đổi giữa các nhóm thông tin chính gồm giới thiệu, môn học và lịch dạy, đánh giá và thành tích.

Phần Content Layout được chia thành hai cột, trong đó khu vực nội dung chính hiển thị thông tin tương ứng với tab được chọn, còn sidebar cung cấp các thông tin bổ trợ và được cố định khi cuộn trang. Nội dung chính bao gồm phần giới thiệu bản thân, kinh nghiệm và phương pháp giảng dạy, danh sách môn học kèm lịch dạy, cũng như tổng hợp đánh giá và nhận xét từ người học. Sidebar hiển thị

thông tin cá nhân, kênh liên hệ và các chỉ số thống kê liên quan đến hoạt động giảng dạy.

3.3.2.5 Màn hình Dashboard Phụ huynh



Hình 3.6 Màn hình dashboard phụ huynh

Dashboard phụ huynh được thiết kế theo mô hình giao diện quản trị hiện đại (admin template), gồm bốn khu vực chức năng chính nhằm hỗ trợ người dùng theo dõi tổng quan và quản lý hoạt động hiệu quả.

Phần Page Header hiển thị biểu tượng minh họa, tiêu đề bảng điều khiển và thông tin chào mừng kèm tên, email người dùng, giúp xác nhận danh tính và tạo cảm giác cá nhân hóa.

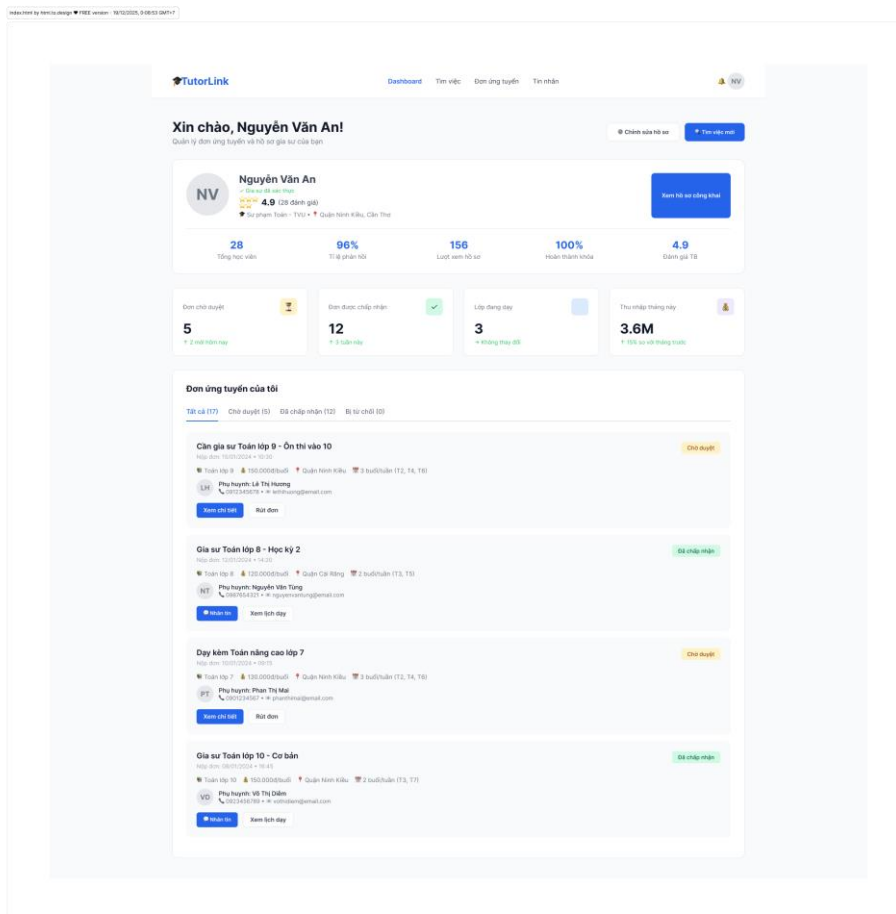
Khu vực Stats Grid trình bày các chỉ số quan trọng dưới dạng lưới bốn thẻ, phản ánh tình trạng bài đăng, nhu cầu tuyển dụng, số lượng đơn ứng tuyển và gia sư đang làm việc.

Quick Actions cung cấp các nút thao tác nhanh cho hai chức năng cốt lõi là đăng tin tìm gia sư và tìm kiếm gia sư, hỗ trợ người dùng truy cập trực tiếp vào các tác vụ thường xuyên sử dụng.

Phần Main Content được bố cục theo lưới hai cột, trong đó khu vực nội dung chính sử dụng điều hướng theo tab để quản lý bài đăng, đơn ứng tuyển và danh sách

gia sư đã thuê. Khu vực phụ bên cạnh hiển thị các hoạt động gần đây dưới dạng dòng thời gian, giúp người dùng theo dõi diễn biến và trạng thái tương tác trong hệ thống.

3.3.2.6 Màn hình Dashboard Gia sư



Hình 3.7 Màn hình Dashboard Gia sư

Dashboard gia sư được xây dựng dựa trên cấu trúc tương tự dashboard phụ huynh, tuy nhiên nội dung và thành phần được điều chỉnh phù hợp với vai trò gia sư, nhằm hỗ trợ tìm kiếm việc làm, theo dõi đơn ứng tuyển và quản lý hoạt động giảng dạy.

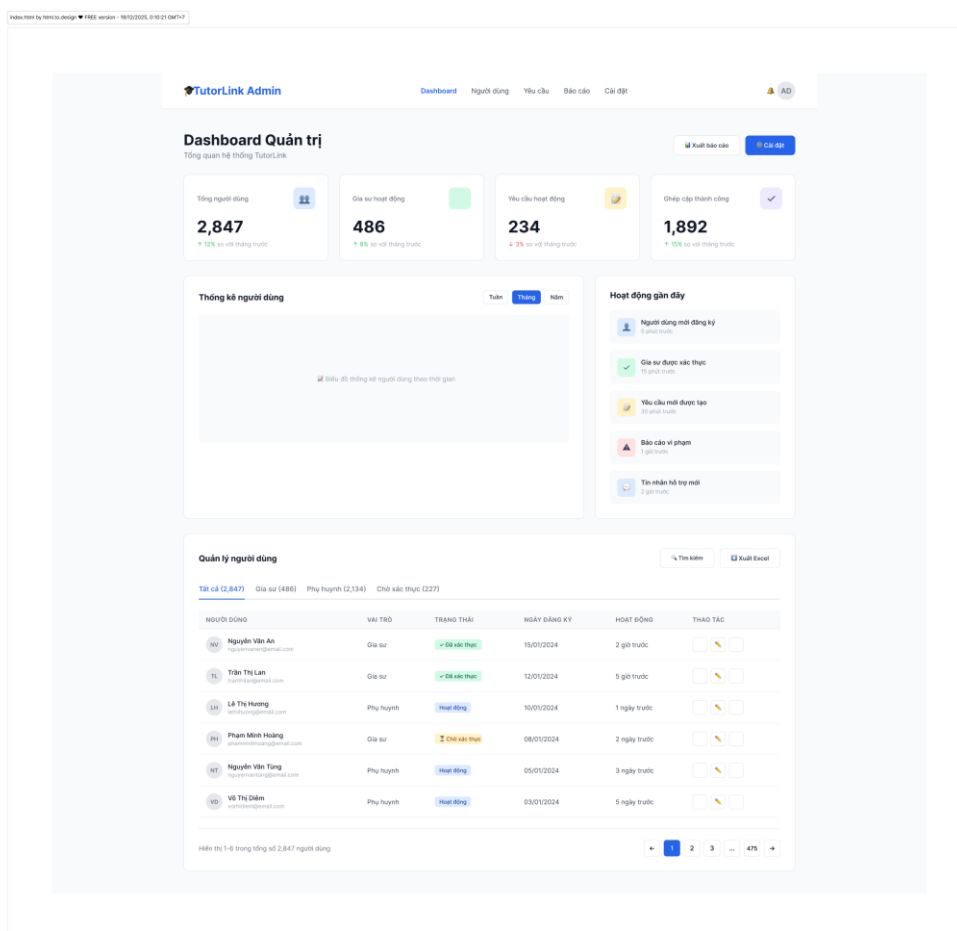
Phần Page Header hiển thị biểu tượng minh họa, tiêu đề bảng điều khiển và thông tin chào mừng người dùng. Điểm nổi bật là nhãn xác thực gia sư kèm mã số sinh viên, giúp tăng mức độ tin cậy và khẳng định uy tín cá nhân trong hệ thống.

Một thành phần đặc trưng của dashboard gia sư là Profile Summary Card, hiển thị ảnh đại diện, tên và các thông tin học tập – giảng dạy quan trọng dưới dạng lưới hai cột. Thẻ này cung cấp nút chỉnh sửa hồ sơ, khuyến khích gia sư cập nhật thông tin nhằm nâng cao khả năng được tuyển chọn.

Khu vực Stats Grid trình bày các chỉ số phản ánh hiệu suất hoạt động của gia sư, bao gồm số đơn đã gửi, số đơn được chấp nhận, lớp học đang giảng dạy và thu nhập trong tháng. Các chỉ số này giúp gia sư theo dõi tiến trình và kết quả làm việc một cách trực quan.

Phần nội dung chính sử dụng điều hướng theo tab, bao gồm danh sách các lớp học đang tuyển để ứng tuyển, danh sách đơn đã gửi kèm trạng thái xử lý và lịch giảng dạy được trình bày dưới dạng lịch kết hợp danh sách lớp sắp tới.

3.3.2.7 Màn hình Dashboard Admin



Hình 3.8 Màn hình Dashboard Admin

Dashboard quản trị viên là màn hình phức tạp nhất của hệ thống, tích hợp nhiều thông tin tổng hợp và chức năng quản trị nhằm hỗ trợ việc giám sát và vận hành toàn bộ nền tảng.

Thanh điều hướng mở rộng cung cấp các mục quản lý chính như bảng điều khiển, người dùng, yêu cầu, báo cáo và cài đặt, đồng thời tích hợp khu vực thông báo và thông tin tài khoản quản trị viên để thuận tiện trong quá trình theo dõi hệ thống.

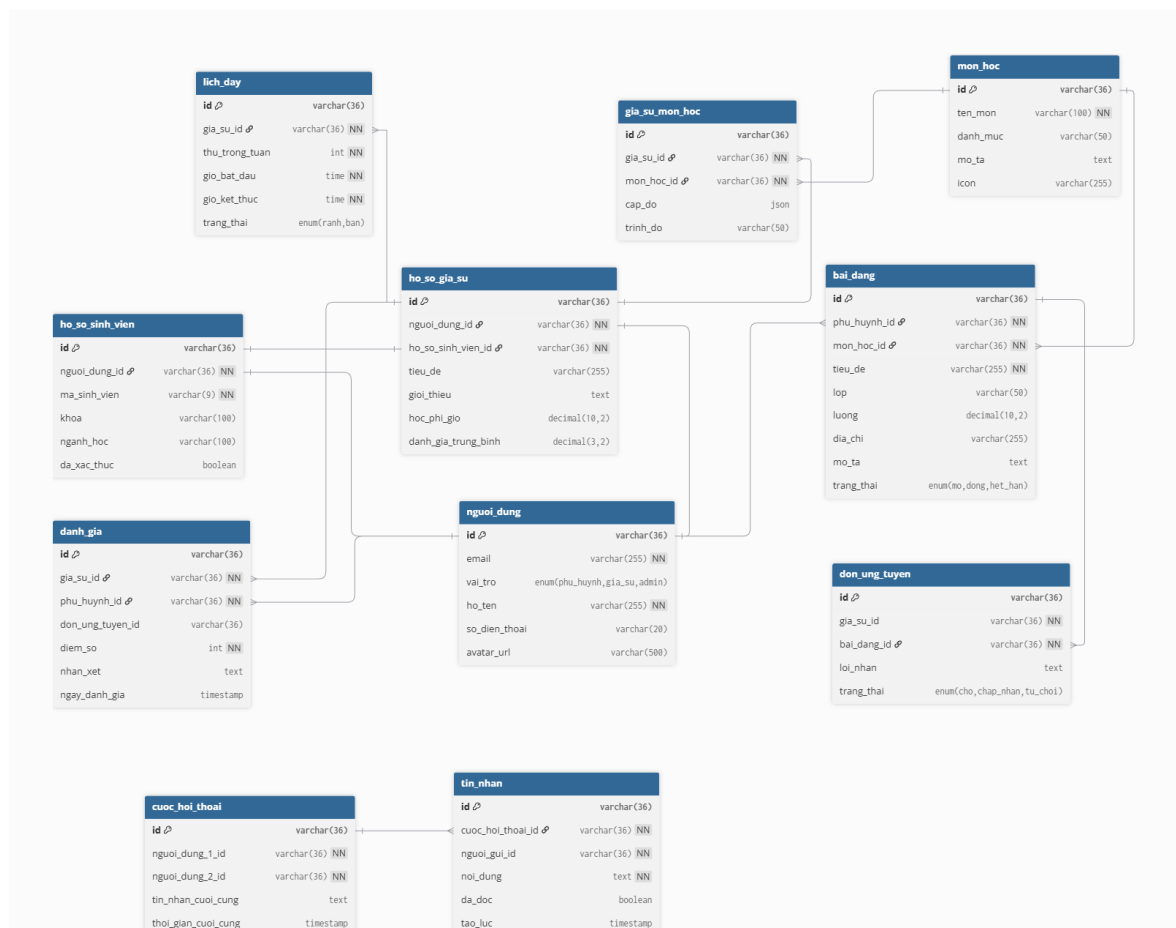
Phần Page Header hiển thị tiêu đề bảng điều khiển, mô tả tổng quan hệ thống và các nút thao tác nhanh cho việc xuất báo cáo và cấu hình hệ thống. Khu vực Stats Grid trình bày các chỉ số quan trọng của hệ thống kèm xu hướng tăng giảm, giúp quản trị viên nhanh chóng đánh giá tình trạng hoạt động.

Khu vực biểu đồ và hoạt động được bố trí theo lưới hai cột, trong đó biểu đồ thống kê thể hiện sự thay đổi số lượng người dùng theo thời gian, còn danh sách hoạt động gần đây cung cấp thông tin về các sự kiện mới phát sinh trong hệ thống.

Phần quản lý chi tiết chiếm toàn bộ chiều rộng giao diện, bao gồm chức năng quản lý người dùng với điều hướng theo tab, bảng dữ liệu chi tiết và phân trang. Cấu trúc này cho phép quản trị viên vừa nắm bắt bức tranh tổng thể của hệ thống, vừa thực hiện các tác vụ quản lý chi tiết một cách hiệu quả.

3.4. Thiết kế dữ liệu

3.4.1. Mô hình ERD



Hình 3.9 Mô hình ERD

3.4.1.1 Danh sách các thực thể và mối kết hợp

Bảng 3.1 Danh sách các thực thể và mối kết hợp

STT	Thực thể 1	Thực thể 2	Loại	Mô tả
1	nguoi_dung	ho_so_sinh_vien	1:1	Người dùng vai trò sinh viên tạo hồ sơ sinh viên để xác thực MSSV.
2	ho_so_sinh_vien	ho_so_gia_su	1:1	Hồ sơ sinh viên đã xác thực đăng ký làm gia sư (quan hệ bắt buộc).
3	ho_so_gia_su	mon_hoc	N:N	Gia sư đăng ký dạy nhiều môn, thông qua bảng gia_su_mon_hoc.
4	ho_so_gia_su	lich_day	1:N	Gia sư đăng ký nhiều khung lịch rảnh trong tuần.
5	nguoi_dung	bai_dang	1:N	Phụ huynh đăng nhiều bài tìm gia sư.
6	mon_hoc	bai_dang	1:N	Mỗi bài đăng chỉ định một môn học cần tìm gia sư.
7	ho_so_gia_su	don_ung_tuyen	1:N	Gia sư nộp nhiều đơn ứng tuyển cho các bài đăng.
8	bai_dang	don_ung_tuyen	1:N	Bài đăng nhận nhiều đơn ứng tuyển từ gia sư.
9	ho_so_gia_su	danh_gia	1:N	Gia sư nhận nhiều đánh giá từ phụ huynh.
10	nguoi_dung	danh_gia	1:N	Phụ huynh viết nhiều đánh giá cho các gia sư.
11	nguoi_dung	cuoc_hoi_thoai	N:N	Người dùng tham gia nhiều cuộc hội thoại (tự tham chiếu).
12	cuoc_hoi_thoai	tin_nhan	1:N	Cuộc hội thoại chứa nhiều tin nhắn.
13	nguoi_dung	tin_nhan	1:N	Người dùng gửi nhiều tin nhắn trong các cuộc hội thoại.
14	nguoi_dung	thong_bao	1:N	Người dùng nhận nhiều thông báo từ hệ thống.

3.4.1.2 Chi tiết các thực thể

Tên thực thể: `nguoi_dung`

Mô tả: Lưu trữ thông tin tài khoản của tất cả người dùng trong hệ thống, bao gồm phụ huynh, sinh viên, gia sư và quản trị viên.

Bảng 3.2 Thực thể `nguoi_dung`

STT	Thuộc tính	Mô tả	Kiểu dữ liệu	Ràng buộc
1	id	Mã định danh duy nhất của người dùng	VARCHAR(36)	PRIMARY KEY
2	email	Email đăng nhập	VARCHAR(255)	
3	mat_khau	Mật khẩu đã mã hóa (bcrypt)	VARCHAR(255)	
4	ho_ten	Họ và tên đầy đủ	VARCHAR(255)	
5	so_dien_thoai	Số điện thoại liên hệ	VARCHAR(15)	
6	dia_chi	Địa chỉ cư trú	TEXT	
7	anh_dai_dien	URL ảnh đại diện	VARCHAR(500)	
8	vai_tro	Vai trò trong hệ thống	ENUM	

Tên thực thể: `ho_so_sinh_vien`

Mô tả: Lưu trữ thông tin sinh viên TVU và trạng thái xác thực MSSV. Chỉ sinh viên được xác thực mới có thể đăng ký làm gia sư.

Bảng 3.3 Thực thể `ho_so_sinh_vien`

STT	Thuộc tính	Mô tả	Kiểu dữ liệu	Ràng buộc
1	id	Mã định danh duy nhất	VARCHAR(36)	PRIMARY KEY
2	nguoi_dung_id	Liên kết đến bảng người dùng	VARCHAR(36)	
3	ma_sinh_vien	Mã số sinh viên TVU (9 chữ số)	VARCHAR(9)	
4	khoa	Khoa/Viện đang theo học	VARCHAR(255)	

5	nganh	Ngành học	VARCHAR(255)	
6	nam_hoc	Năm học hiện tại	VARCHAR(20)	
7	gpa	Điểm trung bình tích lũy	DECIMAL(3,2)	
8	hinh_the_sinh_vien	URL ảnh thẻ sinh viên	VARCHAR(500)	
9	giay_to_xac_thuc	URL giấy tờ khác (bảng điểm...)	TEXT	
10	da_xac_thuc	Trạng thái xác thực từ admin	BOOLEAN	
11	ngay_xac_thuc	Ngày được admin xác thực	TIMESTAMP	

Tên thực thể: ho_so_gia_su

Mô tả: Lưu trữ hồ sơ gia sư, bao gồm kinh nghiệm, giới thiệu, và thống kê đánh giá. Chỉ sinh viên đã xác thực mới tạo được bảng này.

Bảng 3.4 Thực thể ho_so_gia_su

STT	Thuộc tính	Mô tả	Kiểu dữ liệu	Ràng buộc
1	id	Mã định danh duy nhất	VARCHAR(36)	PRIMARY KEY
2	ho_so_sinh_vien_id	Liên kết đến hồ sơ sinh viên	VARCHAR(36)	
3	kinh_nghiem	Mô tả kinh nghiệm dạy học	TEXT	
4	chung_chi	Danh sách chứng chỉ, giải thưởng	TEXT	
5	mo_ta	Giới thiệu bản thân và phong cách dạy	TEXT	
6	trang_thai	Trạng thái nhận lớp	ENUM	
7	diem_trung_binh	Điểm đánh giá trung bình	DECIMAL(2,1)	
8	so_danh_gia	Tổng số đánh giá	INT	

		nhận được		
9	created_at	Thời gian tạo hồ sơ gia sư	TIMESTAMP	
10	updated_at	Thời gian cập nhật cuối	TIMESTAMP	

Tên thực thể: mon_hoc

Mô tả: Danh mục các môn học có thể được dạy kèm trong hệ thống, do quản trị viên quản lý.

Bảng 3.5 Thực thể mon_hoc

STT	Thuộc tính	Mô tả	Kiểu dữ liệu	Ràng buộc
1	id	Mã định danh duy nhất	VARCHAR(36)	PRIMARY KEY
2	ten_mon	Tên môn học	VARCHAR(255)	
3	mo_ta	Mô tả chi tiết môn học	TEXT	
4	trinh_do	Các trình độ áp dụng	ENUM	
5	created_at	Thời gian tạo	TIMESTAMP	
6	updated_at	Thời gian cập nhật cuối	TIMESTAMP	

Tên thực thể: gia_su_mon_hoc

Mô tả: Bảng trung gian lưu trữ các môn học mà gia sư có thể dạy, kèm học phí và kinh nghiệm cho từng môn.

Bảng 3.6 Thực thể gia_su_mon_hoc

STT	Thuộc tính	Mô tả	Kiểu dữ liệu	Ràng buộc
1	id	Mã định danh duy nhất	VARCHAR(36)	PRIMARY KEY
2	ho_so_gia_su_id	Liên kết đến gia sư	VARCHAR(36)	
3	mon_hoc_id	Liên kết đến môn học	VARCHAR(36)	
4	hoc_phi_mong_muon	Học phí mong muốn (VNĐ/buổi)	DECIMAL(10,2)	

5	kinh_nghiem	Kinh nghiệm dạy môn này	TEXT	
6	created_at	Thời gian đăng ký môn	TIMESTAMP	

Tên thực thể: lich_day

Mô tả: Lưu trữ lịch rảnh của gia sư trong tuần, giúp phụ huynh biết khung giờ có thể đặt lịch.

Bảng 3.7 Thực thể lich_day

STT	Thuộc tính	Mô tả	Kiểu dữ liệu	Ràng buộc
1	id	Mã định danh duy nhất	VARCHAR(36)	PRIMARY KEY
2	ho_so_gia_su_id	Liên kết đến gia sư	VARCHAR(36)	
3	thu	Thứ trong tuần	ENUM	
4	gio_bat_dau	Giờ bắt đầu	TIME	
5	gio_ket_thuc	Giờ kết thúc	TIME	
6	trang_thai	Trạng thái khung giờ	ENUM	
7	created_at	Thời gian tạo	TIMESTAMP	
8	updated_at	Thời gian cập nhật cuối	TIMESTAMP	

Tên thực thể: bai_dang

Mô tả: Lưu trữ các bài đăng tìm gia sư do phụ huynh tạo, bao gồm yêu cầu chi tiết về môn học, địa điểm, lịch dạy.

Bảng 3.8 Thực thể bai_dang

STT	Thuộc tính	Mô tả	Kiểu dữ liệu	Ràng buộc
1	id	Mã định danh duy nhất	VARCHAR(36)	PRIMARY KEY
2	nguai_dung_id	Phụ huynh tạo bài đăng	VARCHAR(36)	
3	mon_hoc_id	Môn học cần tìm	VARCHAR(36)	

		gia sư		
4	tieu_de	Tiêu đề bài đăng	VARCHAR(500)	
5	mo_ta	Mô tả chi tiết yêu cầu	TEXT	
6	trinh_do	Trình độ học sinh	ENUM	
7	dia_diem	Địa chỉ dạy học	VARCHAR(500)	
8	hoc_phi	Mức học phí đề xuất (VNĐ/buổi)	DECIMAL(10,2)	

Tên thực thể: don_ung_tuyen

Mô tả: Lưu trữ đơn ứng tuyển của gia sư cho các bài đăng tìm gia sư.

Bảng 3.9 Thực thể don_ung_tuyen

STT	Thuộc tính	Mô tả	Kiểu dữ liệu	Ràng buộc
1	id	Mã định danh duy nhất	VARCHAR(36)	PRIMARY KEY
2	bai_dang_id	Bài đăng được ứng tuyển	VARCHAR(36)	
3	ho_so_gia_su_id	Gia sư nộp đơn	VARCHAR(36)	
4	gioi_thieu	Giới thiệu bản thân	TEXT	
5	hoc_phi_de_xuat	Học phí đề xuất (VNĐ/buổi)	DECIMAL(10,2)	
6	trang_thai	Trạng thái đơn	ENUM	
7	ngay_phan_hoi	Ngày phụ huynh phản hồi	TIMESTAMP	
8	created_at	Thời gian nộp đơn	TIMESTAMP	

Tên thực thể: danh_gia

Mô tả: Lưu trữ đánh giá của phụ huynh về gia sư sau khi kết thúc khóa học.

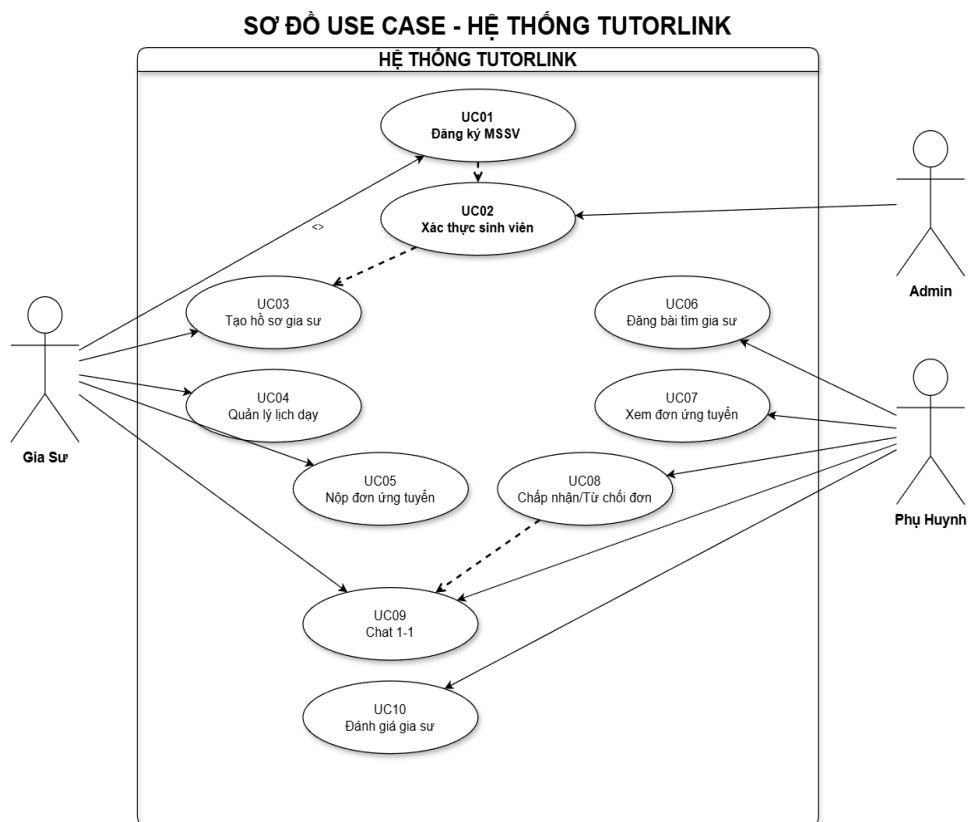
Bảng 3.10 Thực thể danh_gia

STT	Thuộc tính	Mô tả	Kiểu dữ liệu	Ràng buộc
-----	------------	-------	--------------	-----------

1	id	Mã định danh duy nhất	VARCHAR(36)	PRIMARY KEY, UUID
2	ho_so_gia_su_id	Gia sư được đánh giá	VARCHAR(36)	
3	nguoi_dung_id	Phụ huynh đánh giá	VARCHAR(36)	
4	diem_so	Điểm đánh giá	INT	
5	nhan_xet	Nhận xét chi tiết	TEXT	
6	trang_thai	Trạng thái hiển thị	ENUM	
7	created_at	Thời gian đánh giá	TIMESTAMP	
8	updated_at	Thời gian chỉnh sửa cuối	TIMESTAMP	

3.4.2. Mô hình Use Case

3.4.2.1 Sơ đồ Use Case tổng quan



Hình 3.10 Mô hình UseCase

Bảng 3.11 Tổng quan các Use Case

Mã UC	Tên Use Case	Actor	Mô tả	Độ ưu tiên
UC-01	Đăng ký tài khoản sinh viên	Sinh viên TVU	Đăng ký với MSSV để trở thành gia sư	Cao
UC-02	Xác thực MSSV	Quản trị viên	Phê duyệt/Từ chối yêu cầu xác thực MSSV	Cao
UC-03	Tạo hồ sơ gia sư	Gia sư	Tạo hồ sơ để nhận việc dạy kèm	Cao
UC-04	Quản lý lịch dạy	Gia sư	Thiết lập lịch rảnh trong tuần	Trung bình
UC-05	Xem và ứng tuyển bài đăng	Gia sư	Tìm và nộp đơn ứng tuyển bài đăng	Cao
UC-06	Đăng bài tìm gia sư	Phụ huynh	Tạo bài đăng để tìm gia sư phù hợp	Cao
UC-07	Xem danh sách gia sư	Phụ huynh	Tìm kiếm và xem hồ sơ gia sư	Trung bình
UC-08	Quản lý đơn ứng tuyển	Phụ huynh	Xem, chấp nhận/từ chối đơn ứng tuyển	Cao
UC-09	Nhắn tin	Phụ huynh, Gia sư	Chat 1-1 giữa phụ huynh và gia sư	Trung bình
UC-10	Đánh giá gia sư	Phụ huynh	Đánh giá gia sư (1-5 sao) sau khi học	Trung bình

3.4.2.2 Mô tả chi tiết Use Case chính

Bảng 3.12 Mô tả chi tiết Use Case chính

Mã UC	Tiền điều kiện	Hậu điều kiện	Luồng chính	Luồng thay thế
UC-01 Đăng ký SV	Có MSSV, email @st.tvu.edu.vn	Tài khoản tạo thành công, chờ xác thực	1. Nhập email, mật khẩu, họ tên, MSSV 2. Upload ảnh thẻ sinh viên 3. Hệ thống validate và lưu	3a. Email sai định dạng → Lỗi 3b. Email đã tồn tại → Lỗi

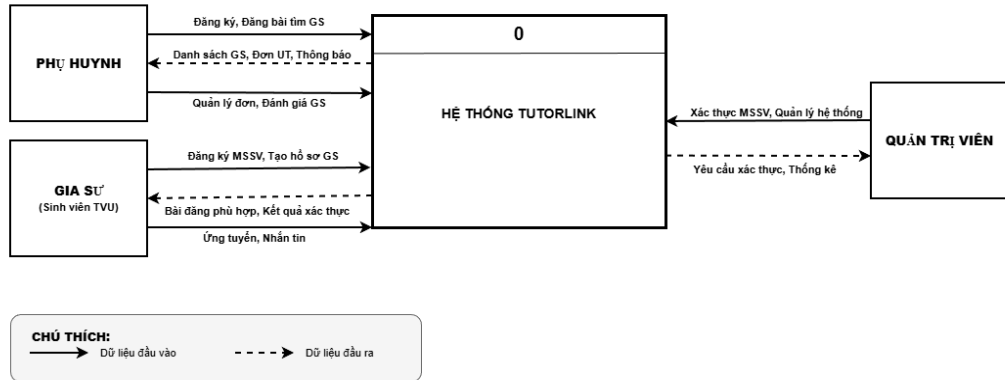
			DB 4. Hiện thị "Chờ admin xác thực"	
UC-02 Xác thực MSSV	Có yêu cầu chờ duyệt	SV được phê duyệt/từ chối, nhận thông báo	1. Admin vào "Yêu cầu xác thực" 2. Kiểm tra MSSV và ảnh thẻ 3. Chọn Phê duyệt/Từ chối 4. Cập nhật DB và gửi thông báo	3a. MSSV không hợp lệ → Từ chối 3b. Ảnh không rõ → Yêu cầu upload lại
UC-06 Đăng bài	Phụ huynh đã đăng nhập	Bài đăng được tạo, GS phù hợp nhận TB	1. Nhập: môn học, địa điểm, học phí, lịch 2. Click "Đăng bài" 3. Hệ thống tìm GS phù hợp 4. Gửi thông báo đến 10 GS	1a. Thiếu thông tin → Highlight trường thiếu
UC-08 Quản lý đơn	Có ≥ 1 đơn ứng tuyển	Đơn được xử lý, GS nhận TB	1. Xem danh sách đơn ứng tuyển 2. Click vào đơn xem hồ sơ GS 3. Chọn Chấp nhận/Từ chối 4. Gửi thông báo, đóng bài nếu chấp nhận	3a. Muốn nhắn tin trước → Click "Nhắn tin"
UC-10 Đánh giá GS	Đã thuê GS, chưa đánh giá	Đánh giá được lưu, điểm TB cập nhật	1. Chọn gia sư cần đánh giá 2. Chọn số sao (1-5) + viết nhận xét 3. Hệ thống lưu và tính điểm TB	2a. Không chọn sao → Lỗi 3a. Đã đánh giá → "Đã đánh giá rồi"

			4. Gửi thông báo cho GS	
--	--	--	-------------------------	--

3.4.3. Thiết kế xử lý

3.4.3.1 Mô hình DFD mức 0

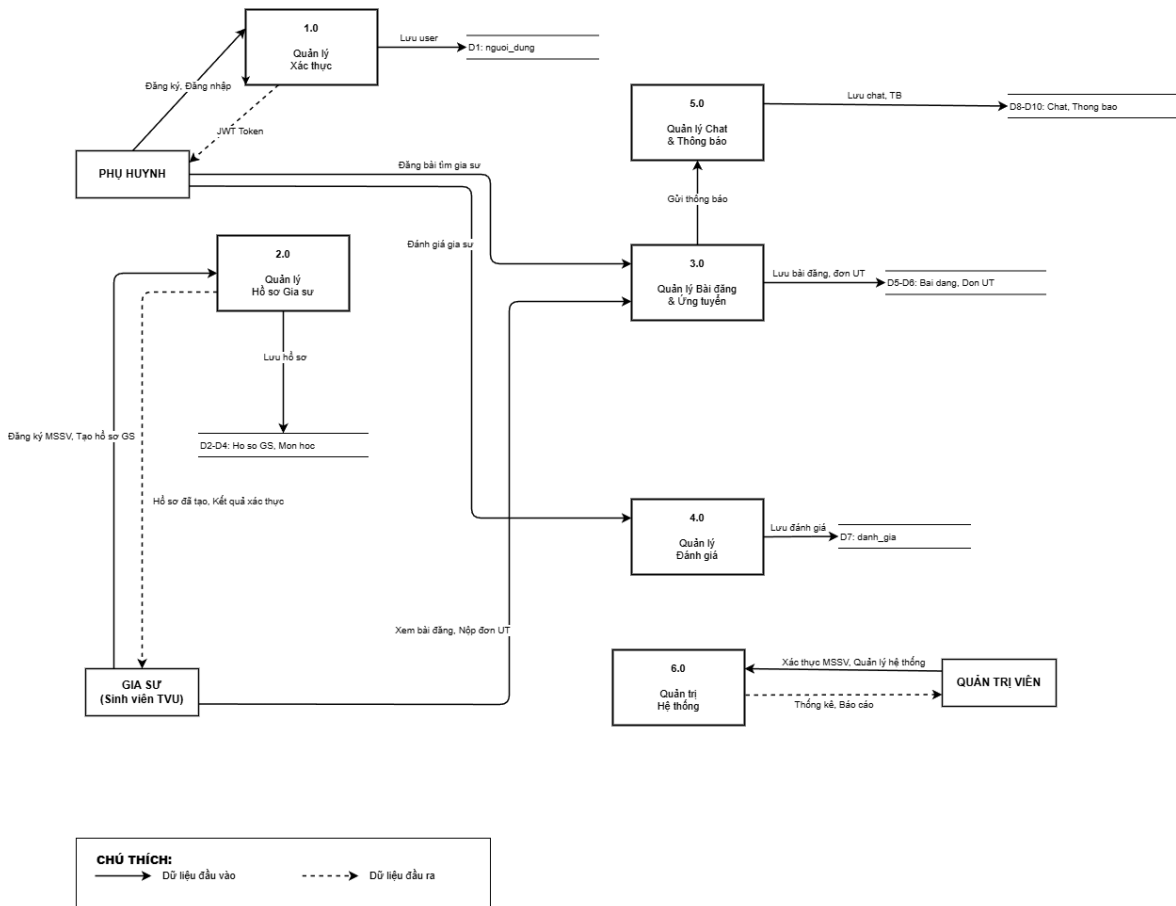
SƠ ĐỒ DFD MỨC 0 - HỆ THỐNG TUTORLINK



Hình 3.11 Mô hình DFD mức 0

3.4.3.2 Mô hình DFD mức 1

SƠ ĐỒ DFD MỨC 1 - HỆ THỐNG TUTORLINK



Hình 3.12 Mô hình DFD mức 1

CHƯƠNG 4. KẾT QUẢ NGHIÊN CỨU

4.1. Dữ liệu thực nghiệm

Hệ thống đã được chuẩn bị dữ liệu thực nghiệm đầy đủ để kiểm thử các chức năng:

Bảng 4.1 Tổng quan dữ liệu thực nghiệm

Loại dữ liệu	Số lượng	Trạng thái
Tài khoản Gia sư	10	Đã xác thực MSSV
Tài khoản Phụ huynh	5	Hoạt động
Tài khoản Admin	1	Hoạt động
Bài đăng tìm gia sư	20	17 đang tuyển, 3 đã tìm được
Đơn ứng tuyển	35	12 chấp nhận, 5 từ chối, 18 chờ
Đánh giá	45	Rating TB: 4.72/5.0

4.2. Kết quả các chức năng chính

4.2.1. Chức năng Xác thực và Đăng nhập

Mô tả: Module xác thực cho phép người dùng đăng ký, đăng nhập và phân quyền theo vai trò.

Kết quả kiểm thử

Bảng 4.2 Kết quả test chức năng xác thực và đăng nhập

Test Case	Mô tả	Input	Kết quả mong đợi	Kết quả thực tế
TC-AUTH-01	Đăng ký phụ huynh hợp lệ	Email, password, tên, SĐT	Tạo tài khoản thành công	Đúng
TC-AUTH-02	Đăng ký gia sư có MSSV	Email, password, MSSV, lớp	Tạo tài khoản, chờ xác thực	Đúng
TC-AUTH-03	Đăng ký email trùng	Email đã tồn tại	Hiển thị lỗi "Email đã được sử dụng"	Đúng
TC-AUTH-04	Đăng nhập hợp lệ	Email, password	Chuyển đến dashboard theo	Đúng

		đúng	role	
TC-AUTH-05	Đăng nhập sai mật khẩu	Email đúng, password sai	Hiển thị lỗi "Sai mật khẩu"	Đúng
TC-AUTH-06	Phân quyền parent	Login với role parent	Redirect /dashboard/parent	Đúng
TC-AUTH-07	Phân quyền tutor	Login với role tutor	Redirect /dashboard/tutor	Đúng
TC-AUTH-08	Phân quyền admin	Login với role admin	Redirect /dashboard/admin	Đúng

4.2.2. Chức năng Quản lý Hồ sơ Gia sư

Mô tả: Gia sư có thể tạo và cập nhật hồ sơ cá nhân, bao gồm thông tin sinh viên TVU, môn dạy, học phí và lịch rảnh.

Kết quả kiểm thử:

Bảng 4.3 Kết quả test chức năng Quản lý Hồ sơ Gia sư

Test Case	Mô tả	Kết quả
TC-PROF-01	Tạo hồ sơ gia sư mới	Lưu thành công
TC-PROF-02	Cập nhật thông tin cá nhân	Cập nhật thành công
TC-PROF-03	Thêm môn học có thể dạy	Thêm thành công
TC-PROF-04	Xóa môn học	Xóa thành công
TC-PROF-05	Cập nhật học phí	Cập nhật thành công
TC-PROF-06	Hiển thị verified badge	Badge hiển thị khi đã xác thực
TC-PROF-07	Upload avatar	Upload và hiển thị thành công

4.2.3. Chức năng Tìm kiếm và Lọc Gia sư

Mô tả: Phụ huynh có thể tìm kiếm gia sư theo nhiều tiêu chí: môn học, cấp lớp, học phí, khu vực, kinh nghiệm.

Kết quả kiểm thử:

Bảng 4.4 Chức năng Tìm kiếm và Lọc Gia sư

Test Case	Mô tả	Kết quả
TC-SEARCH-01	Lọc theo môn học	Hiển thị đúng gia sư
TC-SEARCH-02	Lọc theo cấp học	Hiển thị đúng gia sư
TC-SEARCH-03	Lọc theo học phí (range)	Hiển thị đúng khoảng giá
TC-SEARCH-04	Lọc theo khu vực	Hiển thị đúng địa điểm
TC-SEARCH-05	Lọc nhiều tiêu chí kết hợp	Hiển thị đúng kết quả
TC-SEARCH-06	Pagination	Chuyển trang đúng
TC-SEARCH-07	Click vào TutorCard	Chuyển đến trang detail

4.2.4. Chức năng Dashboard

Mô tả: Mỗi vai trò có dashboard riêng hiển thị thống kê và quản lý hoạt động.

Kết quả kiểm thử:

Bảng 4.5 Kết quả kiểm thử chức năng Dashboard

Dashboard	Chức năng	Kết quả
Parent	Hiển thị stats cards	Đúng 4 metrics
Parent	Quản lý tin đăng	Hiển thị đầy đủ
Parent	Xem đơn ứng tuyển	Hiển thị đầy đủ
Tutor	Hiển thị profile summary	Đúng thông tin
Tutor	Hiển thị stats cards	Đúng 4 metrics
Tutor	Xem tin phù hợp	Gợi ý đúng
Tutor	Quản lý đơn đã gửi	Hiển thị đầy đủ
Admin	Hiển thị stats với trend	Đúng số liệu
Admin	User management table	Đúng dữ liệu
Admin	Xác thực MSSV	Cập nhật thành công

CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1. Kết luận

5.1.1. Kết quả đạt được

Đồ án " Thiết kế và xây dựng nền tảng tìm kiếm và kết nối gia sư – phụ huynh cho học sinh từ lớp 1 đến lớp 12" đã hoàn thành các mục tiêu đề ra với những kết quả cụ thể sau:

Về chức năng: Hệ thống website được xây dựng hoàn chỉnh với 14 trang giao diện, phục vụ 3 nhóm người dùng gồm Phụ huynh, Gia sư (sinh viên Trường Đại học Trà Vinh) và Quản trị viên. Nền tảng triển khai hơn 30 chức năng nghiệp vụ như đăng ký, đăng nhập có phân quyền, quản lý hồ sơ gia sư, đăng tin tuyển dụng, nộp hồ sơ ứng tuyển và các bảng điều khiển quản lý riêng cho từng nhóm người dùng. Đặc biệt, hệ thống tích hợp cơ chế xác thực mã số sinh viên nhằm đảm bảo chỉ sinh viên chính thức của Trường Đại học Trà Vinh mới được đăng ký làm gia sư, góp phần nâng cao độ tin cậy của nền tảng.

Về kỹ thuật: Đồ án áp dụng các công nghệ phát triển web hiện đại, trong đó giao diện người dùng được xây dựng bằng React kết hợp với TypeScript, phía máy chủ sử dụng Express và hệ quản trị cơ sở dữ liệu MySQL. Hệ thống áp dụng cơ chế xác thực người dùng bằng mã thông báo JWT, mật khẩu được mã hóa bằng thuật toán bcrypt nhằm đảm bảo an toàn thông tin. Cơ sở dữ liệu được thiết kế theo hướng chuẩn hóa với 13 bảng, đảm bảo tính toàn vẹn dữ liệu và hiệu quả truy vấn.

Về giao diện: Giao diện hệ thống được thiết kế theo các nguyên tắc thiết kế giao diện hiện đại, hướng đến trải nghiệm người dùng.

5.1.2. Đóng góp mới

Về mặt ứng dụng: Đề tài đã xây dựng một nền tảng đóng vai trò là cầu nối hiệu quả giữa sinh viên Trường Đại học Trà Vinh và nhu cầu tìm kiếm gia sư tại địa phương. Thông qua hệ thống này, sinh viên có thêm cơ hội tạo thu nhập và tích lũy kinh nghiệm giảng dạy thực tế, trong khi phụ huynh có thể dễ dàng tiếp cận nguồn gia sư phù hợp và đáng tin cậy.

Bên cạnh đó, đề tài đã giải quyết được vấn đề về độ tin cậy trong thị trường gia sư tự do bằng việc áp dụng cơ chế xác thực mã số sinh viên. Cơ chế này giúp đảm bảo chỉ những sinh viên chính thức của nhà trường mới có thể đăng ký làm gia

sư, từ đó tạo lợi thế cạnh tranh so với các nền tảng tương tự như Giasudayhoc.vn và TutorOn.vn.

Ngoài ra, hệ thống còn góp phần số hóa quy trình tìm kiếm và tuyển dụng gia sư, giúp giảm thiểu thời gian, chi phí và công sức cho cả phụ huynh và gia sư so với các phương thức truyền thống.

Về mặt kỹ thuật: Về phương diện kỹ thuật, đề tài đã xây dựng hệ thống với kiến trúc mã nguồn được tổ chức theo hướng mô-đun hóa và phân tách rõ ràng chức năng giữa các thành phần, giúp nâng cao khả năng bảo trì, mở rộng và tái sử dụng trong tương lai.

Việc áp dụng ngôn ngữ TypeScript trong quá trình phát triển giúp phát hiện sớm các lỗi tiềm ẩn, nâng cao độ an toàn của mã nguồn và góp phần giảm thiểu lỗi khi hệ thống được triển khai thực tế.

Bên cạnh đó, hệ thống được tài liệu hóa tương đối đầy đủ thông qua các tệp hướng dẫn kỹ thuật như mô tả kiến trúc, hướng dẫn sử dụng API và tài liệu triển khai, tạo điều kiện thuận lợi cho việc tiếp nhận và phát triển hệ thống bởi các lập trình viên khác.

5.1.3. Hạn chế

Mặc dù đã đạt được các mục tiêu đề ra, hệ thống vẫn còn tồn tại một số hạn chế nhất định:

- Hệ thống cũng chưa tích hợp các cổng thanh toán trực tuyến như VNPay hoặc MoMo để hỗ trợ các nghiệp vụ đặt cọc hoặc thanh toán học phí.
- Ngoài ra, chức năng thông báo hiện mới chỉ hoạt động trong phạm vi ứng dụng, chưa hỗ trợ gửi thông báo qua thư điện tử hoặc tin nhắn SMS.
- Bảng điều khiển dành cho quản trị viên cũng chưa cung cấp đầy đủ các biểu đồ phân tích chuyên sâu như tốc độ tăng trưởng, mức độ duy trì người dùng và doanh thu, làm hạn chế khả năng theo dõi và đánh giá hiệu quả hoạt động của hệ thống.

5.2. Hướng phát triển

Trong tương lai, hệ thống “Thiết kế và xây dựng nền tảng tìm kiếm và kết nối gia sư – phụ huynh” có nhiều tiềm năng được mở rộng và hoàn thiện hơn nữa nhằm nâng cao hiệu quả ứng dụng trong thực tiễn cũng như đáp ứng nhu cầu ngày

càng đa dạng của người dùng. Định hướng phát triển của đề tài được xác định theo ba giai đoạn chính: ngắn hạn, trung hạn và dài hạn.

Trong giai đoạn ngắn hạn, hệ thống tập trung hoàn thiện các chức năng cốt lõi và nâng cao mức độ tự động hóa. Trọng tâm là nghiên cứu tích hợp cơ chế xác thực mã số sinh viên tự động với hệ thống quản lý sinh viên của Trường Đại học Trà Vinh, giúp rút ngắn thời gian xác thực, đồng thời đảm bảo độ chính xác và tính minh bạch. Bên cạnh đó, hệ thống trao đổi thông tin giữa phụ huynh và gia sư sẽ được nâng cấp theo hướng thời gian thực, kết hợp với cơ chế thông báo qua thư điện tử hoặc tin nhắn nhằm giúp người dùng kịp thời nắm bắt các sự kiện quan trọng. Ngoài ra, việc tích hợp công thanh toán trực tuyến cũng được xem xét triển khai để hỗ trợ các nghiệp vụ đặt cọc hoặc thanh toán học phí, qua đó nâng cao tính tiện lợi và độ tin cậy của nền tảng.

Trong giai đoạn trung hạn, hệ thống hướng đến mở rộng nền tảng và tăng cường tính thông minh. Cụ thể, việc phát triển ứng dụng di động giúp người dùng tiếp cận hệ thống linh hoạt hơn, đồng thời hỗ trợ các tính năng như thông báo tức thời và tìm kiếm gia sư theo vị trí. Song song đó, hệ thống sẽ nghiên cứu triển khai thuật toán gợi ý gia sư phù hợp dựa trên nhiều tiêu chí như chuyên môn, kinh nghiệm giảng dạy, đánh giá của phụ huynh, khoảng cách địa lý và mức học phí. Ngoài ra, chức năng dạy học trực tuyến thông qua gọi video cũng được xem xét tích hợp nhằm mở rộng phạm vi giảng dạy và nâng cao hiệu quả học tập. Hệ thống quản trị sẽ được bổ sung các công cụ phân tích dữ liệu để hỗ trợ theo dõi và đánh giá hoạt động của toàn bộ nền tảng.

Trong giai đoạn dài hạn, hệ thống được định hướng phát triển thành một nền tảng giáo dục trực tuyến toàn diện. Các chức năng như chia sẻ và kinh doanh tài liệu học tập, chứng nhận năng lực gia sư dựa trên quá trình giảng dạy và phản hồi của phụ huynh sẽ được nghiên cứu triển khai. Bên cạnh đó, hệ thống có thể mở rộng áp dụng cho các cơ sở giáo dục khác trong khu vực và hỗ trợ đa ngôn ngữ nhằm phục vụ nhiều nhóm người dùng hơn. Xa hơn, nền tảng có thể phát triển thành hệ thống quản lý lớp học trực tuyến, hỗ trợ theo dõi tiến độ học tập và cung cấp báo cáo định kỳ cho phụ huynh.

Nhìn chung, các định hướng phát triển trên cho thấy đề tài không chỉ dừng lại ở phạm vi nghiên cứu và xây dựng hệ thống ban đầu mà còn có khả năng mở

rộng và ứng dụng lâu dài trong thực tiễn, góp phần nâng cao hiệu quả kết nối gia sư – phụ huynh và thúc đẩy quá trình chuyển đổi số trong lĩnh vực giáo dục.

DANH MỤC TÀI LIỆU THAM KHẢO

- [1] Meta Platforms, Inc., “React – A JavaScript library for building user interfaces,” 2024. [Online]. Available: <https://react.dev/>. [Accessed: Dec. 19, 2025].
- [2] Microsoft Corporation, “TypeScript: JavaScript With Syntax for Types,” 2024. [Online]. Available: <https://www.typescriptlang.org/>. [Accessed: Dec. 19, 2025].
- [3] OpenJS Foundation, “Express – Fast, unopinionated, minimalist web framework for Node.js,” 2024. [Online]. Available: <https://expressjs.com/>. [Accessed: Dec. 19, 2025].
- [4] Oracle Corporation, *MySQL 8.0 Reference Manual*, 2024. [Online]. Available: <https://dev.mysql.com/doc/refman/8.0/en/>. [Accessed: Dec. 19, 2025].
- [5] M. Jones, J. Bradley, and N. Sakimura, “JSON Web Token (JWT),” RFC 7519, Internet Engineering Task Force, May 2015. [Online]. Available: <https://www.rfc-editor.org/rfc/rfc7519>. [Accessed: Dec. 19, 2025].
- [6] N. Provos and D. Mazières, “A future-adaptable password scheme,” in *Proceedings of the 1999 USENIX Annual Technical Conference*, Monterey, CA, USA, Jun. 1999, pp. 81–91.
- [7] World Wide Web Consortium, “Web Content Accessibility Guidelines (WCAG) 2.1,” W3C Recommendation, Jun. 2018. [Online]. Available: <https://www.w3.org/TR/WCAG21/>. [Accessed: Dec. 19, 2025].
- [8] S. Krug, *Don’t Make Me Think, Revisited: A Common Sense Approach to Web Usability*, 3rd ed. Berkeley, CA, USA: New Riders, 2014.
- [9] Google LLC, “Web Vitals,” 2024. [Online]. Available: <https://web.dev/vitals/>. [Accessed: Dec. 19, 2025].
- [10] R. Elmasri and S. B. Navathe, *Fundamentals of Database Systems*, 7th ed. Boston, MA, USA: Pearson, 2015.
- [11] I. Sommerville, *Software Engineering*, 10th ed. Boston, MA, USA: Pearson, 2015.
- [12] R. T. Fielding, “Architectural styles and the design of network-based software architectures,” Ph.D. dissertation, Dept. Inf. and Comput. Sci., Univ. of California, Irvine, CA, USA, 2000.
- [13] E. Marcotte, *Responsive Web Design*, 2nd ed. New York, NY, USA: A Book Apart, 2014.
- [14] M. G. Moore and G. Kearsley, *Distance Education: A Systems View of Online Learning*, 3rd ed. Belmont, CA, USA: Wadsworth, 2011.
- [15] OWASP Foundation, “OWASP Top Ten Web Application Security Risks,” 2021. [Online]. Available: <https://owasp.org/www-project-top-ten/>. [Accessed: Dec. 19, 2025].

PHỤ LỤC