TRƯỜNG KỸ THUẬT VÀ CÔNG NGHỆ

**KHOA CÔNG NGHỆ THÔNG TIN**



**THỰC TẬP ĐỒ ÁN CHUYÊN NGÀNH**

**HỌC KỲ I, NĂM HỌC 2025-2026**

**PHÁT TRIỂN ỨNG DỤNG WEB HỖ TRỢ HỌC TIẾNG ANH THEO CHỦ ĐỀ**

*Sinh viên thực hiện:*

Họ tên: Tiêu Anh Huy

MSSV: 110122007

Lớp: DA22TTA

*Giảng viên hướng dẫn:*

ThS. Thạch Kọng Saoane

***Vĩnh Long, tháng 12 năm 2025***

TRƯỜNG KỸ THUẬT VÀ CÔNG NGHỆ

**KHOA CÔNG NGHỆ THÔNG TIN**



**THỰC TẬP ĐỒ ÁN CHUYÊN NGÀNH**

**HỌC KỲ I, NĂM HỌC 2025-2026**

**PHÁT TRIỂN ỨNG DỤNG WEB HỖ TRỢ HỌC TIẾNG ANH THEO CHỦ ĐỀ**

*Giảng viên hướng dẫn:*

ThS. Thạch Kọng Saoane

*Sinh viên thực hiện:*

Họ tên: Tiêu Anh Huy

MSSV: 110122007

Lớp: DA22TTA

***Vĩnh Long, tháng 12 năm 2025***

**NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN**

*Vĩnh Long, ngày ….. tháng …… năm ……*

**Giảng viên hướng dẫn**

*(Ký tên và ghi rõ họ tên)*

**NHẬN XÉT CỦA THÀNH VIÊN HỘI ĐỒNG**

*Vĩnh Long, ngày ….. tháng …… năm ……*

**Thành viên hội đồng**

*(Ký tên và ghi rõ họ tên)*

**LỜI CẢM ƠN**

Em xin chân thành gửi lời cảm ơn đặc biệt và sâu sắc đến thầy Thạch Kọng Saoane đã tận tình hướng dẫn và giúp đỡ em trong suốt khoảng thời gian thực hiện đồ án cơ sở ngành. Trong khoảng thời gian ấy, bản thân em phần nào hiểu được sự ân cần, quan tâm của thầy đến em nói riêng và các bạn sinh viên nói chung. Em xin gửi lời cảm ơn sâu sắc đến các giảng viên bộ môn của khoa Công nghệ thông tin đã tạo cơ hội cho em được tham gia học tập và xây dựng đồ án chuyên ngành để em có thể được tiếp cận các kiến thức mới mẻ và bổ ích. Em xin khắc ghi sự dạy bảo của thầy cô và sẽ luôn cố gắng học tập cũng như trau dồi hơn nữa trong tương lai.

**MỤC LỤC**

[CHƯƠNG 1: TỔNG QUAN 12](#_Toc217878372)

[CHƯƠNG 2: NGHIÊN CỨU LÝ THUYẾT 13](#_Toc217878373)

[2.1 Ngôn ngữ HTML 13](#_Toc217878374)

[2.1.1 Giới thiệu ngôn ngữ HTML 13](#_Toc217878375)

[2.1.2 Lịch sử phát triển ngôn ngữ HTML 14](#_Toc217878376)

[2.2 Ngôn ngữ Tailwind CSS 14](#_Toc217878377)

[2.1.3 Giới thiệu ngôn ngữ Tailwind CSS 14](#_Toc217878378)

[2.1.4 Lịch sử phát triển Tailwind CSS 16](#_Toc217878379)

[2.3 ReactJS 16](#_Toc217878380)

[2.1.5 Giới thiệu về ReactJS 16](#_Toc217878381)

[2.1.6 Lịch sử phát triển ReactJS 17](#_Toc217878382)

[2.4 NodeJS 18](#_Toc217878383)

[2.4.1 Giới thiệu về NodeJS 18](#_Toc217878384)

[2.4.2 Lịch sử phát triển NodeJS 19](#_Toc217878385)

[2.5 ExpressJS 20](#_Toc217878386)

[2.5.1 Giới thiệu về ExpressJS 20](#_Toc217878387)

[2.5.2 Lịch sử phát triển ExpressJS 21](#_Toc217878388)

[2.6 MySQL 22](#_Toc217878389)

[2.6.1 Giới thiệu về MySQL 22](#_Toc217878390)

[2.6.2 Lịch sử phát triển MySQL 23](#_Toc217878391)

[CHƯƠNG 3: HIỆN THỰC HÓA NGHIÊN CỨU 25](#_Toc217878392)

[3.1. Phân tích yêu cầu hệ thống 25](#_Toc217878393)

[3.1.1. Yêu cầu chức năng 25](#_Toc217878394)

[3.1.2. Yêu cầu phi chức năng 25](#_Toc217878395)

[3.2 Thiết kế hệ thống 26](#_Toc217878396)

[3.2.1 Kiến trúc hệ thống 26](#_Toc217878397)

[3.2.2. Thiết kế Cơ sở dữ liệu 26](#_Toc217878398)

[3.2.3 Thiết kế giao diện 27](#_Toc217878399)

[3.3 Cài đặt và hiện thực chương trình 29](#_Toc217878400)

[3.3.1. Môi trường phát triển 29](#_Toc217878401)

[3.3.2 Tổ chức mã nguồn 29](#_Toc217878402)

[3.3.3 Hiện thực các chức năng chính 30](#_Toc217878403)

[3.4 Kết quả kiểm thử 32](#_Toc217878404)

[CHƯƠNG 4: KẾT QUẢ NGHIÊN CỨU 33](#_Toc217878405)

[4.1 Các giao diện chức năng chính 33](#_Toc217878406)

[4.1.1 Trang chủ và Điều hướng 33](#_Toc217878407)

[4.1.2 Chức năng Đăng ký và Đăng nhập 34](#_Toc217878408)

[4.1.3. Chức năng học từ vựng và giao tiếp 35](#_Toc217878409)

[4.1.4 Chức năng luyện tập 39](#_Toc217878410)

[4.1.5 Trang hồ sơ cá nhân 43](#_Toc217878411)

[4.1.6 Trang quản trị 44](#_Toc217878412)

[4.4 Kết quả nghiên cứu 47](#_Toc217878413)

[CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 48](#_Toc217878414)

[5.1 Kết luận 48](#_Toc217878415)

[5.2 Hướng phát triển 49](#_Toc217878416)

**DANH MỤC HÌNH ẢNH**

[Hình 1.1 Mô hình thực thể 27](#_Toc217879341)

[Hình 1.2 Giao diện ứng dụng 28](#_Toc217879342)

[Hình 1.2 Giao diện trang chủ và thanh điều hướng 33](#_Toc217879343)

[Hình 1.3 Giao diện trang chủ với các thẻ 34](#_Toc217879344)

[Hình 1.4 Giao diện đăng ký 35](#_Toc217879345)

[Hình 1.5 Giao diện đăng nhập 35](#_Toc217879346)

[Hình 1.6 Giao diện chọn chủ đề học từ vựng 36](#_Toc217879347)

[Hình 1.7 Giao diện chọn chủ đề học giao tiếp 36](#_Toc217879348)

[Hình 1.8 Trang học từ vựng bằng Flashcard 37](#_Toc217879349)

[Hình 1.9 Danh sách từ vựng 38](#_Toc217879350)

[Hình 1.10 Trang học giao tiếp 38](#_Toc217879351)

[Hình 2.1 Các mẫu câu giao tiếp 39](#_Toc217879352)

[Hình 2.2 Chọn kỹ năng luyện tập 40](#_Toc217879353)

[Hình 2.3 Chọn chủ đề và bài tập 40](#_Toc217879354)

[Hình 2.4 Làm bài tập 41](#_Toc217879355)

[Hình 2.5 Trang kết quả bài làm 42](#_Toc217879356)

[Hình 2.6 Trang tiến độ học tập 43](#_Toc217879357)

[Hình 2.7 Trang hồ sơ cá nhân 43](#_Toc217879358)

[Hình 2.8 Chỉnh sửa hồ sơ 44](#_Toc217879359)

[Hình 2.9 Trang chọn quản lý chủ đề từ vựng hoặc giao tiếp 45](#_Toc217879360)

[Hình 2.10 Quản lý chủ đề từ vựng 45](#_Toc217879361)

[Hình 3.1 Quản lý từ vựng 46](#_Toc217879362)

[Hình 3.2 Quản lý chủ đề giao tiếp 46](#_Toc217879363)

[Hình 3.3 Quản lý mẫu câu giao tiếp 47](#_Toc217879364)

**DANH MỤC BẢNG BIỂU**

[Bảng 1.1 Các thực thể 26](#_Toc217878590)

**TÓM TẮT ĐỒ ÁN CHUYÊN NGÀNH**

**Vấn đề nghiên cứu**

Trong bối cảnh toàn cầu hóa, tiếng Anh đã trở thành ngôn ngữ giao tiếp quốc tế quan trọng, đặc biệt trong môi trường làm việc và học tập. Tuy nhiên, tại Việt Nam, người học tiếng Anh thường gặp khó khăn trong việc phát triển kỹ năng giao tiếp thực tế do thiếu môi trường luyện tập tự nhiên và các tài liệu học tập không phản ánh đúng ngữ cảnh sử dụng thực tiễn. Các ứng dụng học tiếng Anh hiện nay tuy hiệu quả trong việc rèn luyện từ vựng và phát âm cơ bản, nhưng vẫn tồn tại hạn chế khi không tận dụng được nguồn video thực tế từ người bản ngữ, dẫn đến việc người học khó hình dung và áp dụng ngôn ngữ vào tình huống thực tế. Do đó, cần có một ứng dụng học tiếng Anh có thể kết hợp giữa ngữ nghĩa, âm thanh, video, giúp cho việc học trở nên dễ dàng tiếp thu và hiệu quả hơn thay vì chỉ học một cách kho khan chỉ toàn nội dung là chữ.

**Các hướng tiếp cận**

Để giải quyết vấn đề trên, sẽ tiến hành nghiên cứu và xây dựng một ứng dụng Web học tiếng Anh theo chủ đề, gồm có các bài học về từ vựng và bài học về giao tiếp. Từ vựng và các mẫu câu giao tiếp sẽ được phân bố theo nhiều chủ đề, người dùng có thể chọn chủ đề mình muốn học. Bên cạnh đó, ứng dụng có phần thực hành để củng cố kiến thức đã học, thực hiện các bài kiểm tra nhỏ, xem kết quả bài làm của mình để biết được các lỗi sai, đồng thời có thể theo dõi tiến độ học tập của người dùng khi sử dụng ứng dụng.

**Cách giải quyết vấn đề**

Hệ thống được thiết kế theo mô hình client – server với kiến trúc rõ ràng và phân lớp. Ở phía người dùng, frontend React cung cấp giao diện trực quan với các module chính: học từ vựng, giao tiếp từ video (Conversation), thực hành (Practice), và quản lý tài khoản (Auth). Phần kiểm tra từ vựng được triển khai dưới dạng trắc nghiệm 10 câu với chấm điểm tức thì. Người dùng đăng nhập tài khoản để học giúp lưu thông tin có thể xem tiến độ học tập của họ.

**Một số kết quả đạt được**

Sau khi thực hiện xong dự án, sẽ hoàn thành được một ứng dụng Web học tiếng Anh theo chủ đề với giao diện thân thiện, dễ sử dụng. Có các chức năng cơ bản liên quan đến việc học tiếng Anh theo chủ đề. Ứng dụng có thể lưu trữ dữ liệu tài khoản người dùng thông qua quá trình học tập và thực hành.

**MỞ ĐẦU**

**Lý do chọn đề tài**

Trong bối cảnh toàn cầu hóa ngày nay, tiếng Anh đã trở thành ngôn ngữ giao tiếp quốc tế thiết yếu, đặc biệt trong giáo dục và công việc, nhưng tại Việt Nam, nhiều người học vẫn gặp khó khăn trong việc rèn luyện kỹ năng thực tế do thiếu tài liệu tương tác và môi trường luyện tập tự nhiên. Là một sinh viên Công nghệ Thông tin, em chọn đề tài "Phát triển ứng dụng web hỗ trợ học tiếng Anh theo chủ đề" vì mong muốn kết hợp kiến thức lập trình với nhu cầu học ngoại ngữ thực tiễn, giúp khắc phục hạn chế của các ứng dụng hiện tại chỉ tập trung vào từ vựng cơ bản mà thiếu yếu tố giao tiếp và video thực tế từ người bản ngữ.

**Mục đích**

Mục đích chính của đề tài là xây dựng một ứng dụng web thân thiện, hỗ trợ người học tiếng Anh cơ bản theo chủ đề qua các module học từ vựng, giao tiếp, thực hành và theo dõi tiến độ cá nhân, sử dụng công nghệ ReactJS cho frontend, Node.js/ExpressJS cho backend và MySQL cho cơ sở dữ liệu, nhằm nâng cao hiệu quả học tập và khuyến khích sự tương tác.

**Đối tượng nghiên cứu**

Đối tượng nghiên cứu hướng đến học sinh, sinh viên và người đi làm tại Việt Nam có trình độ tiếng Anh sơ cấp đến trung cấp, những người cần công cụ hỗ trợ học tập linh hoạt mà không yêu cầu môi trường học truyền thống.

**Phạm vi nghiên cứu**

Phạm vi nghiên cứu tập trung vào việc phát triển các chức năng cốt lõi như đăng nhập tài khoản, học bài theo chủ đề (từ vựng và conversation với video), bài kiểm tra trắc nghiệm 10 câu, và lưu trữ tiến độ học tập; dự án giới hạn ở kiến trúc web client-server cơ bản, không bao gồm tích hợp AI nâng cao hoặc ứng dụng di động, với thời gian thực hiện từ tháng 11 đến tháng 12 năm 2025.

# CHƯƠNG 1: TỔNG QUAN

Trong bối cảnh hội nhập quốc tế, tiếng Anh đóng vai trò quan trọng như ngôn ngữ giao tiếp toàn cầu, đặc biệt trong giáo dục và kinh doanh, nhưng tại Việt Nam, nhiều người học gặp thách thức lớn do thiếu tài nguyên học tập tương tác và môi trường thực hành thực tế, dẫn đến kỹ năng giao tiếp kém phát triển. Đề tài tập trung nghiên cứu vấn đề này bằng cách phát triển ứng dụng web hỗ trợ học tiếng Anh theo chủ đề, nhằm cung cấp công cụ trực tuyến giúp người dùng rèn luyện từ vựng, giao tiếp qua video và thực hành, khắc phục hạn chế của các ứng dụng hiện tại chỉ dừng ở mức học thuộc lòng cơ bản mà thiếu yếu tố ngữ cảnh thực tế.

Các ứng dụng học tiếng Anh phổ biến như Duolingo hay Elsa Speak đã mang lại lợi ích về từ vựng và phát âm, nhưng chúng thường không tùy chỉnh theo chủ đề hoặc tích hợp video bản ngữ để mô phỏng tình huống giao tiếp hàng ngày, khiến người học khó áp dụng kiến thức vào thực tế. Hơn nữa, theo khảo sát từ các nguồn giáo dục, hơn 70% sinh viên Việt Nam gặp khó khăn trong kỹ năng nghe-nói do thiếu nền tảng thực hành, đòi hỏi một giải pháp công nghệ đơn giản, dễ tiếp cận qua web để theo dõi tiến độ cá nhân hóa.

Giải pháp đề xuất là xây dựng ứng dụng web theo mô hình client-server, với frontend ReactJS tạo giao diện thân thiện (các module: đăng nhập/đăng ký, học từ vựng theo chủ đề như "Gia đình" hoặc "Du lịch",..., phần conversation với video, và test củng cố 10 câu), backend Node.js/ExpressJS xử lý API RESTful, kết nối với MySQL để lưu dữ liệu người dùng, bài học và kết quả học tập. Ứng dụng nhắm đến việc nâng cao hiệu quả học tập bằng cách kết hợp đa phương tiện (audio, video) và cá nhân hóa tiến độ, phù hợp với đối tượng học sinh/sinh viên cơ bản, đồng thời dễ mở rộng cho admin quản lý nội dung.

# CHƯƠNG 2: NGHIÊN CỨU LÝ THUYẾT

2.1 Ngôn ngữ HTML

* + 1. Giới thiệu ngôn ngữ HTML

HTML có tên đầy đủ là Hypertext Markup Language nghĩa là ngôn ngữ đánh dấu siêu văn bản. HTML thường được sử dụng để tạo và cấu trúc các phần trong trang web và ứng dụng, phân chia các đoạn văn, heading, link, blockquotes,…

HTML không phải là một ngôn ngữ lập trình mà chỉ là một ngôn ngữ đánh dấu. Điều này đồng nghĩa với việc HTML không thể thực hiện các chức năng “động”. Nói cách khác, HTML tương tự như phần mềm Microsoft Word, chỉ có tác dụng định dạng các thành phần có trong website. [1]

HTML được dùng với 3 mục đích chính:

- Tạo nội dung cho trang web.

- Thiết kế giao diện cho trang web.

- Lập trình tương tác cho trang web.

HTML là một công cụ hữu hiệu giúp người dùng thiết kế và tạo nội dung nhanh chóng cho trang web. Nguyên nhân là bởi HTML sở hữu nhiều ưu điểm nổi bật như sau:

- Kho tài nguyên khổng lồ với cộng đồng người dùng rộng lớn.

- Sử dụng mã nguồn mở nên người dùng có thể sử dụng miễn phí.

- Hoạt động mượt mà trên hầu hết các trình duyệt.

- Cách thức hoạt động đơn giản nên người học có thể dễ dàng nắm bắt được kiến thức để triển khai cho website của mình.

- Có thể tích hợp với nhiều ngôn ngữ khác nhau như PHP, Java, NodeJs, Ruby,… giúp người dùng xây dựng được một trang web với nhiều tính năng hấp dẫn.

Bên cạnh đó HTML cũng có một số nhược điểm:

- Một số trình duyệt còn cập nhật chậm để hỗ trợ các phiên bản mới của HTML, đặc biệt là HTML5.

- Việc kiểm soát cách đọc và hiển thị file HTML của trình duyệt khó thực hiện.

- Chỉ áp dụng với cấu trúc nhất định, không có khả năng sáng tạo.

* + 1. Lịch sử phát triển ngôn ngữ HTML

Hiện nay, có 3 loại HTML được ứng dụng nhiều nhất đó là HTML4, XHTML, HTML5. Sau đây là một vài đặc điểm của các loại HTML này:

- HTML 4: HTML 4 ra đời vào năm 1997 là phiên bản thứ 4 của ngôn ngữ đánh dấu siêu văn bản. Phiên bản HTML này được xuất bản dưới dạng như một W3C Recommendation. HTML 4 áp dụng cho nhiều phần tử và thuộc tính khác nhau cho trình duyệt web.

- XHTML: XHTML có tên đầy đủ là Extensible HyperText Markup Language, nghĩa là ngôn ngữ đánh dấu siêu văn bản mở rộng. Đây là một ngôn ngữ thay thế của HTML với cú pháp chặt chẽ hơn. Cụ thể, XHTML yêu cầu mọi phần tử được đóng bằng thẻ đóng hoặc cú pháp tự đóng riêng và phân biệt được chữ in hoa hoặc chữ in thường, trong khi đó HTML không có điều này.

- HTML5: HTML5 phiên bản thứ 5 của HTML được công bố bởi World Wide Web Consortium (W3C). HTML5 là sự kết hợp giữa HTML4, XHTML, DOM cấp 2 và JavaScript. HTML5 hỗ trợ chạy trên mọi trình duyệt.

2.2 Ngôn ngữ Tailwind CSS

* + 1. Giới thiệu ngôn ngữ Tailwind CSS

Tailwind CSS là một framework CSS theo hướng utility-first (ưu tiên lớp tiện ích), được thiết kế để giúp nhà phát triển xây dựng giao diện người dùng nhanh chóng mà không cần rời khỏi file HTML hoặc JSX. Khác với các framework truyền thống như Bootstrap hay Material-UI (cung cấp các component có sẵn và class ngữ nghĩa), Tailwind CSS cung cấp hàng nghìn class nhỏ (atomic classes) tương ứng trực tiếp với các thuộc tính CSS riêng lẻ.

Component: Trong React, component là một khối xây dựng cơ bản của UI. Mỗi component có thể đại diện cho một phần giao diện (như nút bấm, form, hoặc menu) và có thể dùng lại nhiều lần trong ứng dụng. Có hai loại component chính trong React: Class components (sử dụng khi cần quản lý trạng thái và các lifecycle methods) và Functional components (dễ dàng sử dụng và thường được ưa chuộng, đặc biệt khi kết hợp với Hooks).

JSX (JavaScript XML): JSX là một cú pháp mở rộng của JavaScript cho phép viết mã HTML trong JavaScript. JSX trông giống như HTML nhưng thực chất là một sự kết hợp giữa HTML và JavaScript. Khi sử dụng JSX, React sẽ tự động biên dịch nó thành JavaScript thuần túy. JSX giúp việc xây dựng giao diện trở nên dễ dàng và dễ hiểu hơn.

Các đặc điểm nổi bật của Tailwind CSS:

- Utility-first: mọi style đều được biểu diễn bằng các class nhỏ.

- Tùy biến cao: cấu hình thông qua file tailwind.config.js.

- Hiệu suất tốt: Kết hợp với chế độ JIT (Just-In-Time) để chỉ tạo ra các lớp CSS cần thiết, giảm kích thước file và tăng tốc độ tải trang.

- Hệ sinh thái phong phú: Hỗ trợ nhiều plugin và tích hợp tốt với các framework như React, Vue, Next.js. Hỗ trợ mở rộng các chức năng như tối ưu hóa giao diện form, cải thiện hiển thị văn bản dài, duy trì tỷ lệ khung hình cho phần tử, giới hạn số dòng hiển thị của văn bản. [2]

- Hỗ trợ responsive, dark mode, hover, focus, group-hover… ngay trong class.

- Khi đã quen, tốc độ thiết kế giao diện tăng 3–5 lần so với CSS truyền thống.

Nhờ những đặc tính trên, Tailwind CSS đã trở thành lựa chọn hàng đầu trong các dự án hiện đại sử dụng React, Next.js, Vue, Laravel, và được cộng đồng đánh giá là framework CSS phổ biến thứ hai thế giới (sau Bootstrap) vào năm 2025.

* + 1. Lịch sử phát triển Tailwind CSS

Tailwind CSS được phát triển bởi Adam Wathan và cộng sự, ra mắt lần đầu vào năm 2017. Tính đến tháng 5 năm 2025, Tailwind CSS đã đạt hơn 88.000 sao trên GitHub, trở thành một trong những framework CSS phổ biến nhất hiện nay.

Nhờ sự đột phá của Just-In-Time compiler (2020) và engine viết bằng Rust (2022), Tailwind CSS đã vượt qua hầu hết các đối thủ về tốc độ build và trải nghiệm phát triển, trở thành tiêu chuẩn thực tế (de-facto standard) cho các dự án web hiện đại năm 2025.

2.3 ReactJS

* + 1. Giới thiệu về ReactJS

ReactJS là một thư viện JavaScript mã nguồn mở được tạo ra nhằm hỗ trợ phát triển giao diện người dùng (UI) cho các ứng dụng web và mobile. Công nghệ này do Facebook xây dựng và duy trì, giúp lập trình viên phát triển các ứng dụng đơn trang (SPA) có tốc độ phản hồi tốt, cấu trúc rõ ràng và dễ mở rộng. Một trong những ưu điểm nổi bật của React là khả năng tái sử dụng các thành phần giao diện (components), từ đó đơn giản hóa việc bảo trì và phát triển hệ thống về sau. [3]

Mục tiêu cốt lõi của React là tối ưu hóa việc cập nhật giao diện khi dữ liệu thay đổi. Thay vì phải render lại toàn bộ trang, React chỉ xử lý những phần thực sự cần cập nhật, giúp giảm thời gian xử lý và nâng cao hiệu năng tổng thể của ứng dụng. Cơ chế này hoạt động dựa trên Virtual DOM — một mô hình DOM ảo cho phép React theo dõi, so sánh và áp dụng sự thay đổi một cách hiệu quả mà không làm mới toàn bộ UI.

Những điểm nổi bật làm nên sức mạnh của React có thể kể đến như sau:

- Virtual DOM: React sử dụng một bản DOM ảo để so sánh và xác định phần giao diện cần thay đổi, từ đó chỉ cập nhật các khu vực thật sự cần thiết nhằm nâng cao hiệu suất.

- Kiến trúc hướng thành phần (Component-Based): Giao diện được chia nhỏ thành các component độc lập, có thể tái sử dụng ở nhiều nơi, giúp cấu trúc dự án rõ ràng và dễ mở rộng.

- JSX (JavaScript XML): JSX cho phép kết hợp cú pháp HTML trực tiếp trong mã JavaScript, hỗ trợ lập trình viên viết code trực quan, dễ đọc và dễ bảo trì hơn.

- Luồng dữ liệu một chiều (One-way Data Flow): Dữ liệu trong React di chuyển theo một hướng — từ component cha xuống component con — giúp hạn chế xung đột dữ liệu và đơn giản hóa quá trình kiểm lỗi.

- Hooks: Bộ API mạnh mẽ giúp xử lý trạng thái và các tác vụ phụ trong component dạng hàm, cho phép sử dụng các tính năng trước đây chỉ có trong class component mà không cần định nghĩa lớp. [4]

* + 1. Lịch sử phát triển ReactJS

ReactJS xuất hiện vào năm 2011, được phát triển bởi Jordan Walke – kỹ sư phần mềm của Facebook. Trong giai đoạn này, Facebook phải đối mặt với nhiều khó khăn trong việc mở rộng và bảo trì giao diện người dùng cho các ứng dụng có quy mô lớn. Để giải quyết vấn đề đó, React được xây dựng nhằm mang lại một cách tiếp cận mới cho việc phát triển UI, dựa trên mô hình component tách biệt, có khả năng tái sử dụng và quản lý trạng thái rõ ràng. Nhờ vậy, lập trình viên có thể giảm đáng kể độ phức tạp trong việc phát triển và bảo trì giao diện web.

Ban đầu React được áp dụng nội bộ trong các dự án như Facebook Ads và Instagram. Sau những kết quả khả quan, Facebook chính thức công bố React dưới dạng mã nguồn mở vào năm 2013. Công nghệ này nhanh chóng thu hút sự chú ý rộng rãi nhờ hiệu năng cao, tính linh hoạt và thiết kế thân thiện với lập trình viên.

Giai đoạn 2015 đánh dấu bước phát triển vượt bậc của React. Hệ sinh thái của React mở rộng với sự xuất hiện của React Router – công cụ điều hướng cho SPA, và Redux – thư viện quản lý trạng thái trở thành tiêu chuẩn phổ biến trong nhiều dự án. Cũng trong năm này, Facebook giới thiệu React Native, cho phép phát triển ứng dụng iOS và Android bằng JavaScript, mở ra cơ hội lớn cho các lập trình viên web dễ dàng chuyển sang phát triển app di động mà không cần học thêm ngôn ngữ mới.

Cùng với thời gian, cộng đồng React phát triển mạnh mẽ trên toàn cầu. Các hội nghị như React Conf và React Europe ngày càng thu hút đông đảo lập trình viên, góp phần thúc đẩy sự đổi mới liên tục và mở rộng hệ sinh thái React thông qua hàng loạt thư viện và công cụ hỗ trợ.

Một cột mốc nổi bật khác là năm 2018 – thời điểm React giới thiệu Hooks. Đây là thay đổi quan trọng giúp component dạng hàm có thể xử lý state và side effects mà không dùng đến class-based components, qua đó cải thiện tính đơn giản và khả năng tái sử dụng mã nguồn. Trong những năm tiếp theo, React tiếp tục cải tiến với các tính năng như Concurrent Mode và Suspense, giúp xử lý dữ liệu bất đồng bộ mượt mà hơn và nâng cao trải nghiệm người dùng.

Ngày nay, ReactJS đã trở thành một trong những công nghệ cốt lõi trong phát triển giao diện web hiện đại, được đánh giá cao nhờ hiệu năng, khả năng mở rộng và cộng đồng mạnh mẽ, giữ vững vị thế dẫn đầu trong lĩnh vực UI development.

2.4 NodeJS

2.4.1 Giới thiệu về NodeJS

Node.js là một môi trường runtime JavaScript mã nguồn mở và đa nền tảng, cho phép chạy JavaScript bên ngoài trình duyệt, chẳng hạn trên máy chủ. Node.js được xây dựng dựa trên V8 JavaScript Engine của Chrome — một engine hiệu suất cao viết bằng C++ và JavaScript — giúp thực thi code nhanh và hiệu quả.

Node.js được phát triển bởi Ryan Dahl vào năm 2009, mở ra khả năng sử dụng JavaScript cho các ứng dụng server-side.

Một số đặc điểm nổi bật của Node.js gồm:

- Mã nguồn mở (Open-source): Node.js có mã nguồn mở, cho phép cộng đồng toàn cầu truy cập, sử dụng và đóng góp vào quá trình phát triển.

- Đa nền tảng (Cross-platform): Node.js chạy trên nhiều hệ điều hành phổ biến như Linux, macOS và Windows, giúp ứng dụng dễ dàng triển khai trên nhiều môi trường khác nhau mà không cần chỉnh sửa code.

- Môi trường thực thi JavaScript: Node.js mở rộng khả năng chạy JavaScript ra ngoài trình duyệt, cho phép sử dụng JavaScript để xử lý logic server-side, tương tác cơ sở dữ liệu, xử lý file, và nhiều tác vụ khác.

- Dựa trên V8 JavaScript Engine: Nhờ sử dụng V8, Node.js thực thi JavaScript với hiệu suất cao, đồng thời hỗ trợ các tính năng mới nhất của ngôn ngữ này. [4]

2.4.2 Lịch sử phát triển NodeJS

Node.js được phát triển bởi Ryan Dahl vào năm 2009. Mục tiêu ban đầu là mở rộng khả năng của JavaScript — vốn chỉ chạy trong trình duyệt — để có thể chạy trên máy chủ. Điều này cho phép lập trình viên sử dụng JavaScript cho cả frontend và backend, đánh dấu bước ngoặt quan trọng trong việc phát triển ứng dụng web full-stack bằng một ngôn ngữ duy nhất.

Dự án ban đầu được đặt tên là “web.js”, với mục tiêu đơn giản là thay thế các web server truyền thống như Apache và Nginx. Tuy nhiên, tiềm năng của dự án vượt xa vai trò của một web server đơn thuần. Khi nhận thấy Node.js có thể trở thành một nền tảng đa năng và mạnh mẽ hơn, tác giả đã đổi tên thành Node.js, thể hiện khả năng tạo ra các “node” trong mạng và nhấn mạnh tính modular, linh hoạt của nền tảng.

Trong những năm đầu (năm 2009 – 2010), Node.js tập trung vào các ứng dụng mạng I/O phi đồng bộ (asynchronous), nhờ vào cơ chế event-driven, non-blocking I/O. Thiết kế này giúp Node.js xử lý hàng nghìn kết nối đồng thời mà không cần nhiều tài nguyên, mở ra khả năng phát triển các ứng dụng real-time như chat, streaming, hay API server hiệu suất cao.

Giai đoạn năm 2011 – 2014, Node.js nhanh chóng được cộng đồng lập trình viên toàn cầu đón nhận. Các thư viện, công cụ và package manager npm (Node Package Manager) được phát triển, tạo ra một hệ sinh thái phong phú cho việc phát triển ứng dụng. Node.js trở thành lựa chọn phổ biến cho các startup và các dự án quy mô lớn nhờ hiệu suất cao và khả năng mở rộng linh hoạt.

Từ năm 2015 đến nay, Node.js hiện thuộc OpenJS Foundation, một tổ chức phi lợi nhuận đảm bảo sự phát triển bền vững và quản lý ổn định. Trong những năm gần đây, Node.js liên tục được cập nhật để hỗ trợ các tính năng mới của JavaScript, cải thiện hiệu suất, bảo mật và khả năng tương thích với các môi trường hiện đại.

Node.js không chỉ thay đổi cách phát triển backend bằng JavaScript mà còn mở ra khái niệm full-stack JavaScript, giúp lập trình viên dùng một ngôn ngữ duy nhất cho cả frontend và backend. Node.js là nền tảng cho nhiều framework nổi tiếng khác như Express.js, NestJS, Meteor, và cũng là thành phần quan trọng trong các stack phổ biến như MEAN, MERN, MEVN.

2.5 ExpressJS

2.5.1 Giới thiệu về ExpressJS

ExpressJS là một framework mã nguồn mở, miễn phí, được xây dựng để hỗ trợ phát triển các ứng dụng web trên nền tảng Node.js. Nhờ cấu trúc đơn giản và dễ tiếp cận, Express (cách gọi khác cho ExpressJS) cho phép lập trình viên nhanh chóng tạo ra các ứng dụng web cũng như API một cách hiệu quả, đặc biệt thuận lợi cho những người đã quen làm việc với JavaScript.

Vì được phát triển dựa trên Node.js, ExpressJS cung cấp sẵn nhiều công cụ và chức năng hỗ trợ, giúp lập trình viên xây dựng các ứng dụng đơn trang (SPA), đa trang hoặc kết hợp nhiều kiểu giao diện khác nhau. Framework này cũng hỗ trợ mô hình kiến trúc MVC (Model–View–Controller), giúp tổ chức mã nguồn rõ ràng và dễ quản lý trong các ứng dụng server-side.

Một trong những lợi thế lớn của ExpressJS là khả năng đơn giản hóa quá trình xử lý request và xây dựng API. Nếu chỉ dùng Node.js nguyên bản, lập trình viên phải tự viết nhiều đoạn mã để xử lý routing, middleware và các tác vụ liên quan. ExpressJS giúp rút ngắn khối lượng công việc này thông qua hệ thống middleware linh hoạt, dễ mở rộng.

Bên cạnh đó, hệ sinh thái mạnh mẽ từ cộng đồng và vô số thư viện mở rộng cũng khiến Express trở thành lựa chọn phổ biến trong phát triển web backend và API trên nền tảng Node.js. [5]

Đặc điểm nổi bật của ExpressJS:

- Nhẹ và tối giản: Express có kiến trúc nhỏ gọn, chỉ cung cấp những tính năng cơ bản, cho phép lập trình viên tự do mở rộng theo nhu cầu dự án.

- Hỗ trợ middleware linh hoạt: Middleware là các hàm xử lý request và response. Express cho phép chuỗi các middleware một cách linh hoạt, giúp quản lý xác thực, logging, xử lý lỗi… dễ dàng.

- Hệ thống routing mạnh mẽ: Express cung cấp cơ chế routing trực quan, cho phép định nghĩa các đường dẫn URL, phương thức HTTP và xử lý tương ứng một cách rõ ràng và dễ quản lý.

- Tương thích với NodeJS: Chạy trên Node.js, tận dụng cơ chế non-blocking, event-driven, giúp xây dựng ứng dụng web và API hiệu suất cao.

2.5.2 Lịch sử phát triển ExpressJS

Express.js được tạo ra vào năm 2010 bởi TJ Holowaychuk, nhằm cung cấp một framework web tối giản nhưng mạnh mẽ trên nền tảng Node.js. Thiết kế của Express tập trung vào việc đơn giản hóa quá trình xây dựng ứng dụng web và API, đồng thời mang lại khả năng mở rộng linh hoạt. Nhờ những ưu điểm này, Express nhanh chóng trở thành một trong những framework phổ biến nhất trong hệ sinh thái Node.js.

Express theo triết lý “unopinionated” — không áp đặt cách tổ chức ứng dụng, cho phép lập trình viên tự do lựa chọn cấu trúc và công cụ phù hợp. Framework cung cấp các tính năng quan trọng như routing, middleware, và hỗ trợ template engines, giúp giảm đáng kể khối lượng công việc trong quá trình phát triển. Sự phát triển mạnh mẽ của Node.js góp phần củng cố vị thế của Express như một lựa chọn hàng đầu cho các ứng dụng server-side.

Năm 2014, Express được chuyển giao quyền sở hữu cho StrongLoop, công ty chuyên về các công cụ Node.js, và sau đó StrongLoop được IBM mua lại. Express.js trở thành một phần của OpenJS Foundation, đảm bảo sự phát triển bền vững, quản lý ổn định và cộng đồng hỗ trợ lâu dài.

Ngày nay, Express.js vẫn là framework phổ biến nhất cho Node.js, được áp dụng trong nhiều dự án từ nhỏ đến lớn. Framework liên tục được cập nhật để tương thích với các phiên bản Node.js mới và đáp ứng các yêu cầu phát triển hiện đại. Cộng đồng mạnh mẽ cùng tài liệu phong phú giúp Express duy trì vị trí quan trọng trong hệ sinh thái Node.js, trở thành công cụ không thể thiếu cho lập trình viên trên toàn thế giới.

2.6 MySQL

2.6.1 Giới thiệu về MySQL

MySQL là một hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) mã nguồn mở, được ứng dụng rộng rãi trong việc lưu trữ, truy xuất và quản lý dữ liệu cho website, ứng dụng và hệ thống phần mềm. MySQL hoạt động theo mô hình client–server, trong đó máy khách gửi yêu cầu (query) và máy chủ MySQL xử lý, trả về kết quả tương ứng.

MySQL hoạt động theo mô hình **client–server**, nghĩa là máy khách (client) gửi các lệnh SQL, máy chủ MySQL xử lý và trả kết quả. Điều này giúp phân tách trách nhiệm giữa giao diện ứng dụng và quản lý dữ liệu, tăng hiệu quả và bảo mật trong quá trình vận hành.

Phần mềm sử dụng SQL (Structured Query Language) — ngôn ngữ tiêu chuẩn để thao tác cơ sở dữ liệu quan hệ — cho phép:

- Tạo và quản lý cơ sở dữ liệu, bảng và quan hệ giữa các bảng.

- Thực hiện các truy vấn phức tạp để lọc, sắp xếp, tổng hợp dữ liệu.

- Quản lý người dùng, phân quyền truy cập và bảo mật dữ liệu.

Nhờ mã nguồn mở, MySQL có thể được sử dụng miễn phí, đồng thời cho phép lập trình viên tùy biến hoặc mở rộng chức năng theo nhu cầu riêng của dự án. Hệ quản trị này được phát triển và duy trì bởi Oracle Corporation, và đã trở thành một trong những RDBMS phổ biến nhất trên thế giới, từ các website nhỏ, ứng dụng doanh nghiệp đến các hệ thống dữ liệu lớn. [6]

Các đặc điểm nổi bật của MySQL:

- Miễn phí và mã nguồn mở: Người dùng dễ dàng tiếp cận, đồng thời có phiên bản thương mại được Oracle hỗ trợ.

- Hiệu suất cao: MySQL xử lý hiệu quả các truy vấn đồng thời, phù hợp với các hệ thống có quy mô lớn.

- Tương thích đa nền tảng: Hoạt động tốt với nhiều ngôn ngữ lập trình như PHP, Python, Java và được tích hợp rộng rãi trong các CMS phổ biến như WordPress, Drupal, Magento.

2.6.2 Lịch sử phát triển MySQL

MySQL là một trong những hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) mã nguồn mở phổ biến nhất trên thế giới, có lịch sử phát triển lâu dài:

Năm 1994 – Thành lập MySQL AB: MySQL được phát triển bởi công ty MySQL AB tại Thụy Điển, nhằm cung cấp một hệ quản trị cơ sở dữ liệu nhanh, nhẹ và dễ sử dụng cho các ứng dụng web và doanh nghiệp.

Năm 1995 – Phát hành phiên bản đầu tiên: Phiên bản đầu tiên của MySQL chính thức được phát hành, nhanh chóng được cộng đồng lập trình viên đón nhận nhờ tính hiệu quả, miễn phí và khả năng mở rộng.

Năm 2008 – MySQL AB được mua lại bởi Sun Microsystems: Sun Microsystems mua lại MySQL AB, nhằm tích hợp MySQL vào các giải pháp phần mềm doanh nghiệp và mở rộng hệ sinh thái công nghệ của mình.

Năm 2010 – Sun Microsystems sáp nhập vào Oracle: Oracle Corporation tiếp quản Sun Microsystems, qua đó trở thành đơn vị phát triển và duy trì MySQL. Điều này đánh dấu sự chuyển giao quyền sở hữu sang một tập đoàn lớn, đảm bảo sự phát triển ổn định và hỗ trợ lâu dài cho MySQL.

Sự ra đời của MariaDB: Đội ngũ phát triển MySQL tách ra và tạo ra MariaDB, một nhánh mã nguồn mở tương thích với MySQL, nhằm duy trì tính độc lập và cam kết phát triển cộng đồng.

Phiên bản 5.6 và 5.7:

- Năm 2013, phiên bản MySQL 5.6 được phát hành, bổ sung nhiều tính năng nâng cao về hiệu suất, tối ưu truy vấn và bảo mật.

- Năm 2015, phiên bản MySQL 5.7 ra mắt với các cải tiến về JSON support, optimizer, replication, giúp đáp ứng nhu cầu ứng dụng hiện đại.

Hiện tại, MySQL hiện có hai phiên bản chính:

- MySQL Community Server: Miễn phí, mã nguồn mở, phù hợp với các dự án cá nhân và doanh nghiệp nhỏ.

- MySQL Enterprise Server: Phiên bản thương mại, đi kèm các tính năng cao cấp, hỗ trợ kỹ thuật chuyên nghiệp và bảo mật nâng cao.

Nhờ lịch sử phát triển lâu dài, MySQL đã trở thành nền tảng cơ sở dữ liệu ổn định, phổ biến và đáng tin cậy, được sử dụng rộng rãi trong nhiều hệ thống từ website nhỏ đến các ứng dụng doanh nghiệp lớn.

CHƯƠNG 3: HIỆN THỰC HÓA NGHIÊN CỨU

3.1. Phân tích yêu cầu hệ thống

3.1.1. Yêu cầu chức năng

Dựa trên mục tiêu và phạm vi nghiên cứu đã đề ra, hệ thống "EnglishComm" được xây dựng với các nhóm chức năng chính sau:

- Quản lý tài khoản: Cho phép người dùng đăng ký, đăng nhập để hệ thống định danh và lưu trữ tiến độ học tập.

- Học tập theo chủ đề:

* Học từ vựng: Hiển thị danh sách từ vựng theo các chủ đề (ví dụ: Gia đình, Du lịch) kèm nghĩa và âm thanh minh họa.
* Học giao tiếp: Cung cấp các đoạn hội thoại mẫu tích hợp video thực tế từ người bản ngữ giúp người học hình dung ngữ cảnh.

- Thực hành và Kiểm tra:

* Hệ thống cung cấp bài kiểm tra trắc nghiệm 10 câu sau mỗi bài học để củng cố kiến thức.
* Chấm điểm tức thì và hiển thị kết quả đúng/sai.

- Theo dõi tiến độ: Lưu trữ lịch sử làm bài, điểm số và hiển thị lộ trình học tập của người dùng.

3.1.2. Yêu cầu phi chức năng

Giao diện: Thân thiện, trực quan, sử dụng Tailwind CSS để tối ưu hóa trải nghiệm người dùng (UX/UI) và khả năng tương thích trên nhiều thiết bị (Responsive).

Hiệu năng: Tốc độ tải trang nhanh nhờ cơ chế Single Page Application (SPA) của ReactJS và xử lý bất đồng bộ của Node.js.

Dữ liệu: Cơ sở dữ liệu MySQL đảm bảo tính toàn vẹn và khả năng truy xuất nhanh chóng.

3.2 Thiết kế hệ thống

3.2.1 Kiến trúc hệ thống

Hệ thống được thiết kế theo mô hình Client – Server:

- Client (Frontend): Xây dựng bằng ReactJS, chịu trách nhiệm hiển thị giao diện và tương tác với người dùng.

- Server (Backend): Xây dựng bằng Node.js và ExpressJS, cung cấp các API RESTful để xử lý logic nghiệp vụ và kết nối cơ sở dữ liệu.

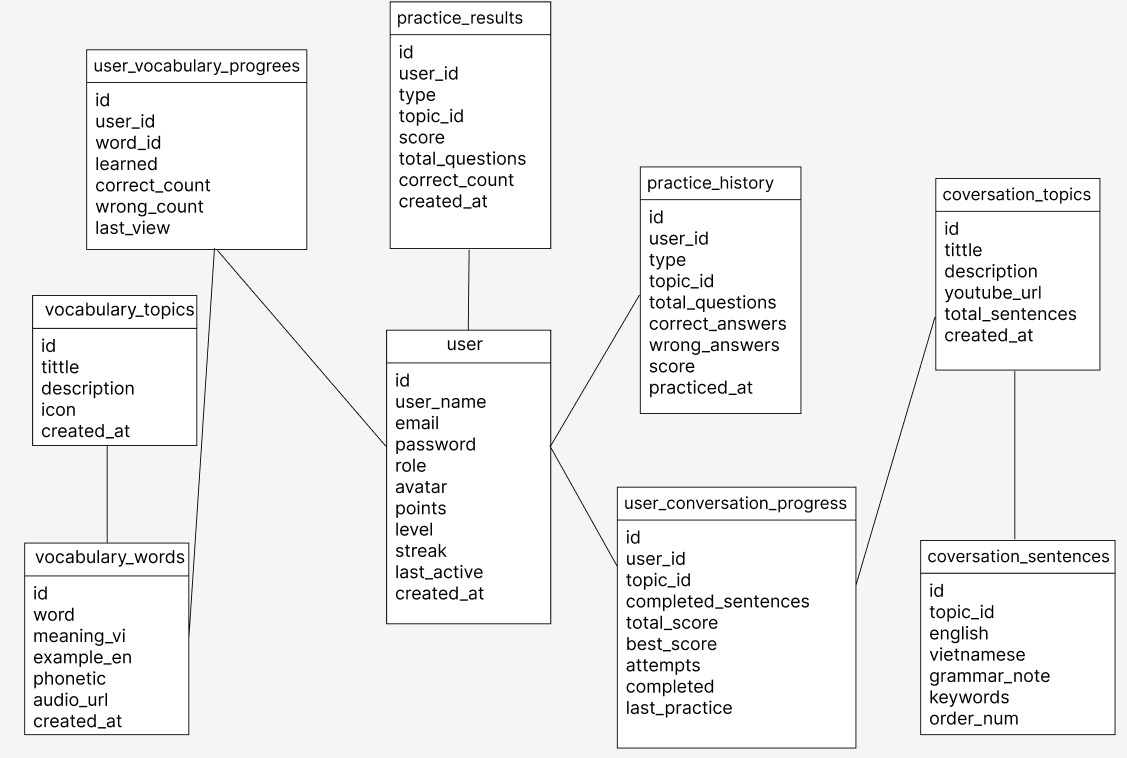
- Database: MySQL dùng để lưu trữ thông tin người dùng, nội dung bài học và kết quả kiểm tra.

3.2.2. Thiết kế Cơ sở dữ liệu

Cơ sở dữ liệu được thiết kế để quản lý các thực thể chính sau:

Bảng 1.1 Các thực thể

|  |  |
| --- | --- |
| **Tên bảng** | **Ý nghĩa** |
| user | Người dùng |
| vocabulary\_topics | Chủ đề từ vựng |
| conversation\_topics | Chủ đề giao tiếp |
| vocabulary\_words | Từ vựng |
| conversation\_sentences | Mẫu câu |
| practice\_results | Kết quả luyện tập |
| practice\_history | Lịch sử luyện tập |
| user\_vocabulary\_progress | Tiến độ kiểm tra từ vựng |
| user\_conversation\_progress | Tiến độ kiểm tra giao tiếp |



Hình 1.1 Mô hình thực thể

3.2.3 Thiết kế giao diện

Giao diện người dùng (UI) của ứng dụng được thiết kế hướng tới sự tối giản, hiện đại và tập trung vào trải nghiệm học tập. Hệ thống sử dụng thư viện ReactJS để xây dựng giao diện dưới dạng các thành phần (components) độc lập kết hợp với Tailwind CSS để tùy biến kiểu dáng linh hoạt và đồng bộ.

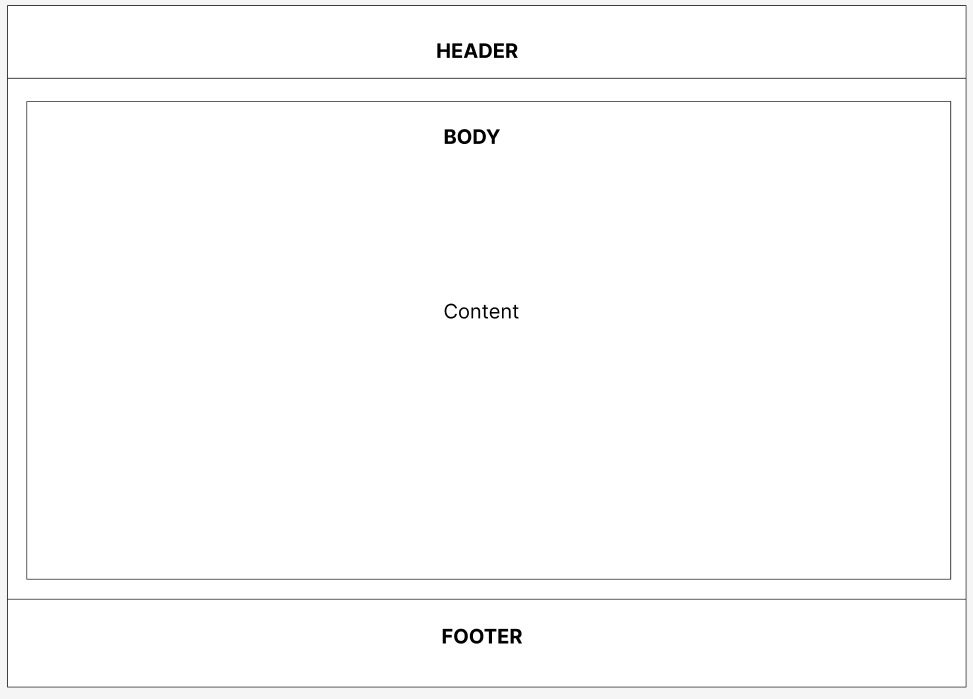
Để đảm bảo tính nhất quán và giúp người dùng dễ dàng điều hướng giữa các chức năng học tập, bố cục tổng thể của mọi trang trong ứng dụng được chuẩn hóa theo cấu trúc tiêu chuẩn gồm 3 vùng chính như mô hình dưới đây:

Cụ thể các thành phần trong bố cục bao gồm:

- Header (Thanh điều hướng): Được cố định ở phía trên cùng của trang, chứa logo thương hiệu, hệ thống menu điều hướng (Trang chủ, Học từ vựng, Giao tiếp, Thực hành, Tiến độ) và khu vực quản lý tài khoản người dùng (Ảnh đại diện, Đăng nhập/Đăng xuất). Phần này giúp người dùng truy cập nhanh vào các chức năng bất kể đang ở trang nào.

- Body (Khu vực nội dung chính): Đây là phần thay đổi linh hoạt tùy theo ngữ cảnh sử dụng (Routing). Ví dụ: khi người dùng chọn học từ vựng, khu vực này sẽ hiển thị danh sách Flashcard; khi vào bài kiểm tra, nó sẽ hiển thị bộ câu hỏi trắc nghiệm.

- Footer (Chân trang): Nằm ở dưới cùng, cung cấp thông tin bản quyền, liên kết nhanh và thông tin liên hệ của đội ngũ phát triển.



Hình 1.2 Giao diện ứng dụng

3.3 Cài đặt và hiện thực chương trình

3.3.1. Môi trường phát triển

Để đảm bảo ứng dụng hoạt động ổn định, tương thích tốt giữa Client (Frontend) và Server (Backend), cũng như thuận tiện cho quá trình gỡ lỗi (debug) và kiểm thử, hệ thống được xây dựng và triển khai trên môi trường phát triển với các thông số cấu hình phần mềm và công cụ cụ thể như sau:

- Hệ điều hành: Windows 10/11.

- Công cụ lập trình (IDE): Visual Studio Code (VS Code).

- Trình duyệt kiểm thử: Google Chrome/ Microsoft Edge.

- Server cục bộ: XAMPP (cho MySQL và Apache).

- Nền tảng Runtime: Node.js (phiên bản LTS).

3.3.2 Tổ chức mã nguồn

Để đảm bảo tính khoa học, dễ bảo trì và thuận tiện cho việc mở rộng các chức năng trong tương lai, mã nguồn của dự án được tổ chức tách biệt rõ ràng thành hai phần chính: Frontend (Giao diện người dùng) và Backend (Xử lý nghiệp vụ phía Server).

Cấu trúc thư mục được sắp xếp theo mô hình module hóa, trong đó các thành phần có chức năng tương tự được gom nhóm vào các thư mục riêng biệt. Dưới đây là mô tả chi tiết cách tổ chức cây thư mục của dự án:

- Cấu trúc phía Frontend (Client): Phần Frontend được xây dựng bằng ReactJS, toàn bộ mã nguồn nằm trong thư mục frontend/src. Các thành phần giao diện và trang chức năng được phân chia cụ thể:

* src/components/: Chứa các thành phần giao diện (UI components) có khả năng tái sử dụng cao, được dùng chung cho nhiều trang khác nhau. Ví dụ: Navbar.tsx (Thanh điều hướng), Footer.tsx (Chân trang).
* src/pages/: Chứa các trang hiển thị chính của ứng dụng, tương ứng với các tuyến đường (routes) mà người dùng truy cập. Ví dụ: Home.tsx (Trang chủ), Login.tsx (Đăng nhập), Practice.tsx (Trang luyện tập).
* Đặc biệt, các trang quản trị được gom vào thư mục con src/pages/admin/ (như AdminDashboard.tsx) để dễ dàng phân quyền và quản lý.

- Cấu trúc phía Backend (Server) Phần Backend sử dụng Node.js và ExpressJS, được tổ chức theo mô hình MVC (Model-View-Controller) để xử lý các yêu cầu từ phía Client:

* src/config/: Chứa các file cấu hình hệ thống, file cơ sở dữ liệu, quan trọng nhất là db.js để thiết lập kết nối tới cơ sở dữ liệu MySQL.
* src/controllers/: Chứa các file xử lý logic nghiệp vụ chính (Controller). Tại đây, các hàm sẽ tiếp nhận yêu cầu (Request), xử lý dữ liệu và trả về phản hồi. Ví dụ: auth.controllers.js (xử lý đăng nhập/đăng ký), vocabulary.controller.js (xử lý dữ liệu từ vựng).
* src/routes/: Định nghĩa các điểm cuối (API Endpoints) và điều hướng yêu cầu đến Controller tương ứng. Ví dụ: auth.routes.js, vocabulary.routes.js.
* src/app.js & server.js: Các file khởi tạo ứng dụng, cấu hình middleware (như CORS, Body Parser) và khởi động máy chủ.

3.3.3 Hiện thực các chức năng chính

**Chức năng đăng nhập/đăng ký**: Đây là chức năng đầu tiên người dùng tiếp cận để định danh và lưu trữ tiến độ học tập.

- Giao diện (Frontend): Được xây dựng trong component Auth.tsx.

- Sử dụng React Hooks (useState) để quản lý trạng thái của các trường dữ liệu nhập vào (như regName, regEmail, loginEmail...) và trạng thái chuyển đổi giữa hai tab "Register" và "Login".

- Hỗ trợ tải lên ảnh đại diện (avatar) thông qua FormData để xử lý dữ liệu đa phương tiện.

Xử lý Logic:

- Sử dụng hàm fetch để gửi yêu cầu HTTP phương thức POST đến các API backend (/api/auth/register và /api/auth/login).

- Sau khi đăng nhập thành công, hệ thống nhận về token (JWT) và thông tin user. Dữ liệu này được lưu trữ vào localStorage để duy trì phiên làm việc và xác thực cho các yêu cầu sau này.

**Chức năng học từ vựng và giao tiếp:**

- Học từ vựng: Hệ thống gọi API GET /api/vocabulary/:topicId để lấy danh sách từ vựng thuộc chủ đề đã chọn. Dữ liệu trả về (gồm từ, nghĩa, phiên âm, ảnh minh họa) được hiển thị dưới dạng thẻ Flashcard giúp người học dễ ghi nhớ.

- Học giao tiếp: Sử dụng thẻ <video> của HTML5 để nhúng trực tiếp các video bài học. Nội dung video được lấy từ các kênh dạy học giao tiếp tiếng Anh trực tuyến trên nền tảng Youtube. Người dùng có thể quan sát từ ngữ, mẫu câu và cách nhấn giọng qua lời thoại.

**Chức năng thực hành:** Chức năng này giúp người dùng củng cố kiến thức sau khi học.

- Hệ thống thực hiện thuật toán ngẫu nhiên (random) để lấy ra 10 câu hỏi từ ngân hàng dữ liệu trong MySQL, đảm bảo mỗi lần kiểm tra là một bộ câu hỏi khác nhau.

- Cơ chế chấm điểm:

* Khi người dùng nộp bài, hệ thống so sánh đáp án người dùng chọn với đáp án đúng trong cơ sở dữ liệu.
* Điểm số được tính toán trên thang điểm 10 và lưu trữ trực tiếp vào bảng practice\_results kèm theo thời gian thực hiện.

Phản hồi: Kết quả (số câu đúng/sai, điểm số) và đáp án chi tiết được hiển thị ngay lập tức (Real-time feedback) giúp người học tự đánh giá năng lực.

**Chức năng Quản trị (Admin Dashboard):** Là chức năng dành riêng cho đội ngũ phát triển ứng dụng.

**-** Giao diện dành riêng cho tài khoản có quyền role = 'admin'.

**-** Tại đây, người quản trị có thể xem danh sách các chủ đề hiện có.

**-** Các thao tác CRUD (Thêm, Xem, Sửa, Xóa) được thực hiện thông qua các API quản trị, cho phép cập nhật nội dung bài học một cách nhanh chóng mà không cần can thiệp vào mã nguồn hay cơ sở dữ liệu thủ công.

3.4 Kết quả kiểm thử

Sau quá trình cài đặt và triển khai thử nghiệm trên môi trường cục bộ (Localhost), hệ thống đã hoạt động ổn định và đạt được các tiêu chí kỹ thuật đề ra:

- Kết nối dữ liệu: Kết nối giữa Backend (Node.js) và Database (MySQL) hoạt động thông suốt. Các thao tác truy vấn (Query) diễn ra chính xác, không xảy ra lỗi mất mát dữ liệu.

- Hiệu năng API: Các API phản hồi ổn, dữ liệu JSON trả về đúng cấu trúc yêu cầu của Frontend.

- Giao diện người dùng: Giao diện hiển thị tốt, bố cục không bị vỡ trên trình duyệt ở các kích thước màn hình máy tính phổ biến. Các hiệu ứng chuyển trang, hover và modal hoạt động mượt mà.

Chức năng nghiệp vụ:

- Quy trình Đăng ký -> Đăng nhập -> Lưu Token hoạt động chính xác.

- Chức năng "Lưu điểm số" vào bảng practice\_results đã được kiểm chứng: dữ liệu điểm số xuất hiện trong Database ngay sau khi người dùng nhấn nút "Nộp bài".

CHƯƠNG 4: KẾT QUẢ NGHIÊN CỨU

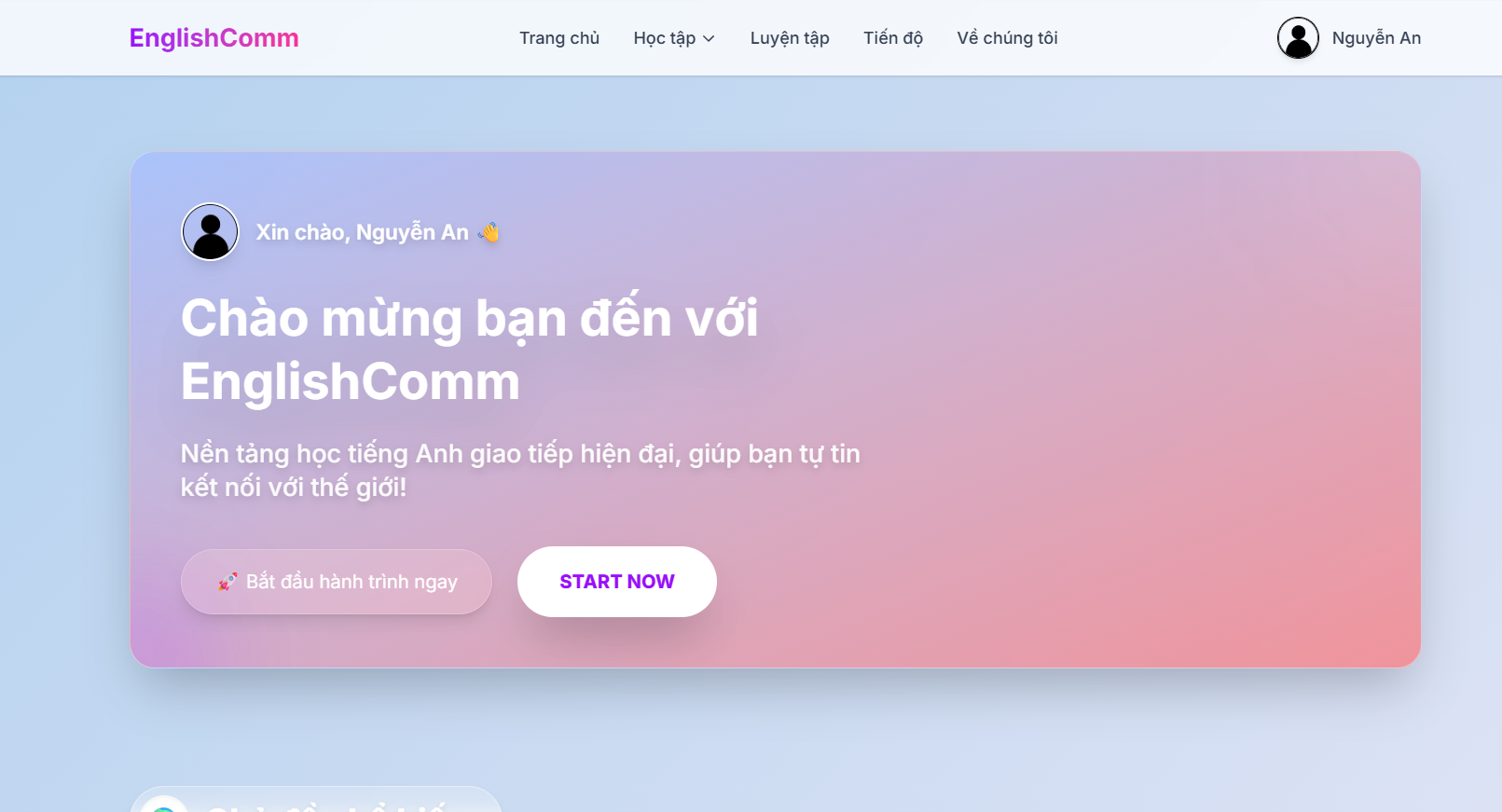
4.1 Các giao diện chức năng chính

Sau quá trình thiết kế và cài đặt, ứng dụng EnglishComm đã hoàn thiện với đầy đủ các phân hệ dành cho Người dùng (User) và Quản trị viên (Admin). Dưới đây là các giao diện chức năng tiêu biểu:

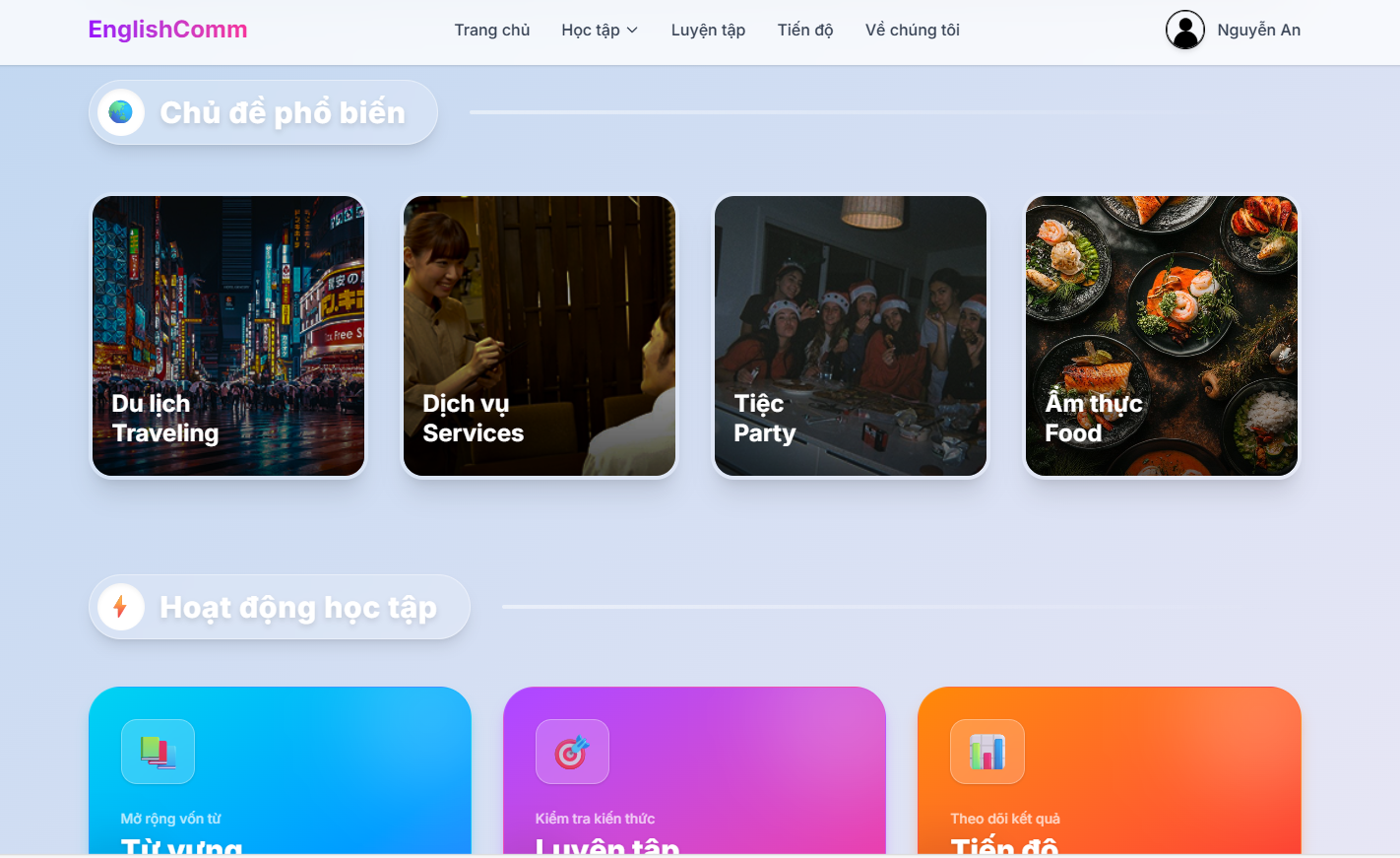
4.1.1 Trang chủ và Điều hướng

Giao diện trang chủ được thiết kế theo phong cách hiện đại với thanh điều hướng (Navbar) cố định và các thẻ chức năng (Card) nổi bật.

Mô tả: Hiển thị lời chào cá nhân hóa (dựa trên tài khoản đăng nhập), danh sách các chủ đề từ vựng phổ biến và lối tắt truy cập nhanh vào phần Luyện tập hoặc Xem tiến độ.



Hình 1.2 Giao diện trang chủ và thanh điều hướng



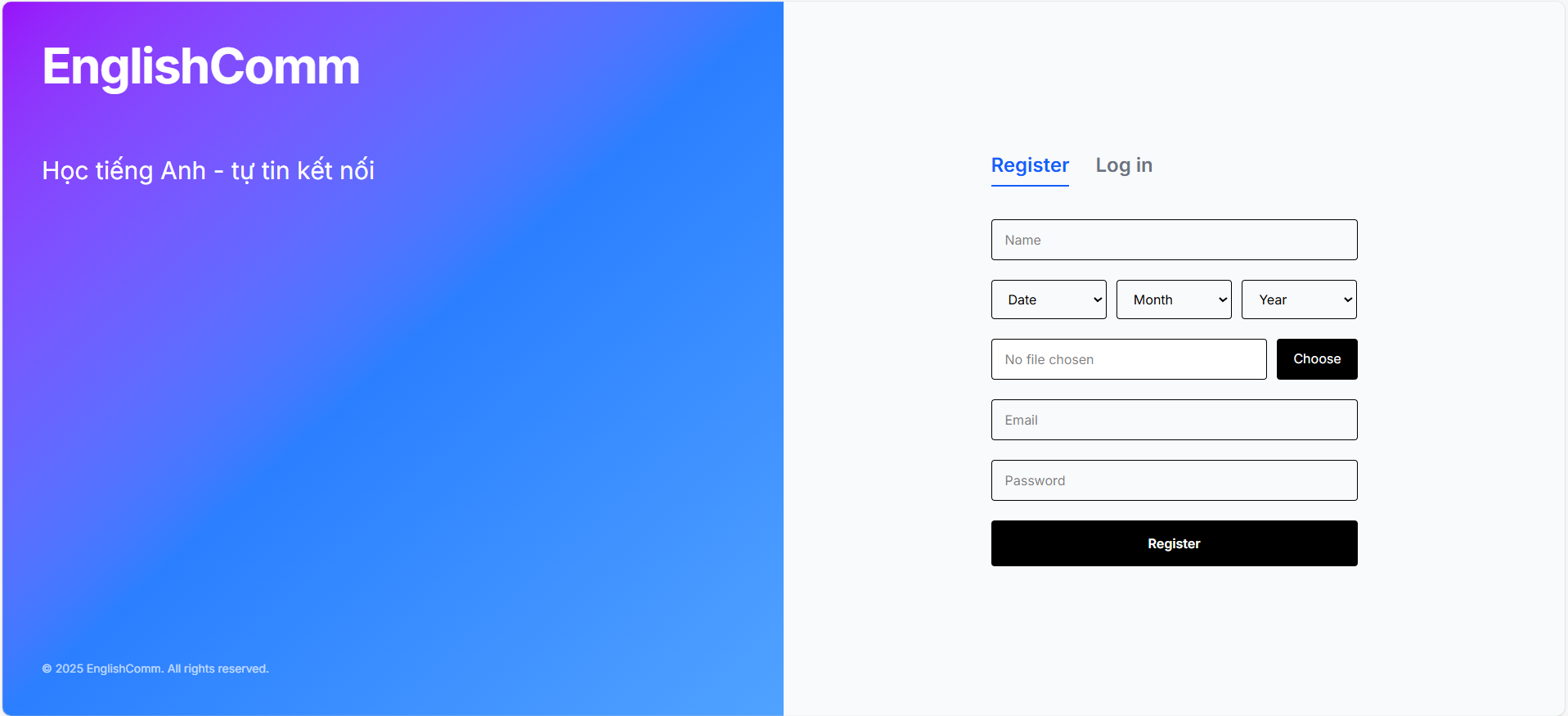
Hình 1.3 Giao diện trang chủ với các thẻ

4.1.2 Chức năng Đăng ký và Đăng nhập

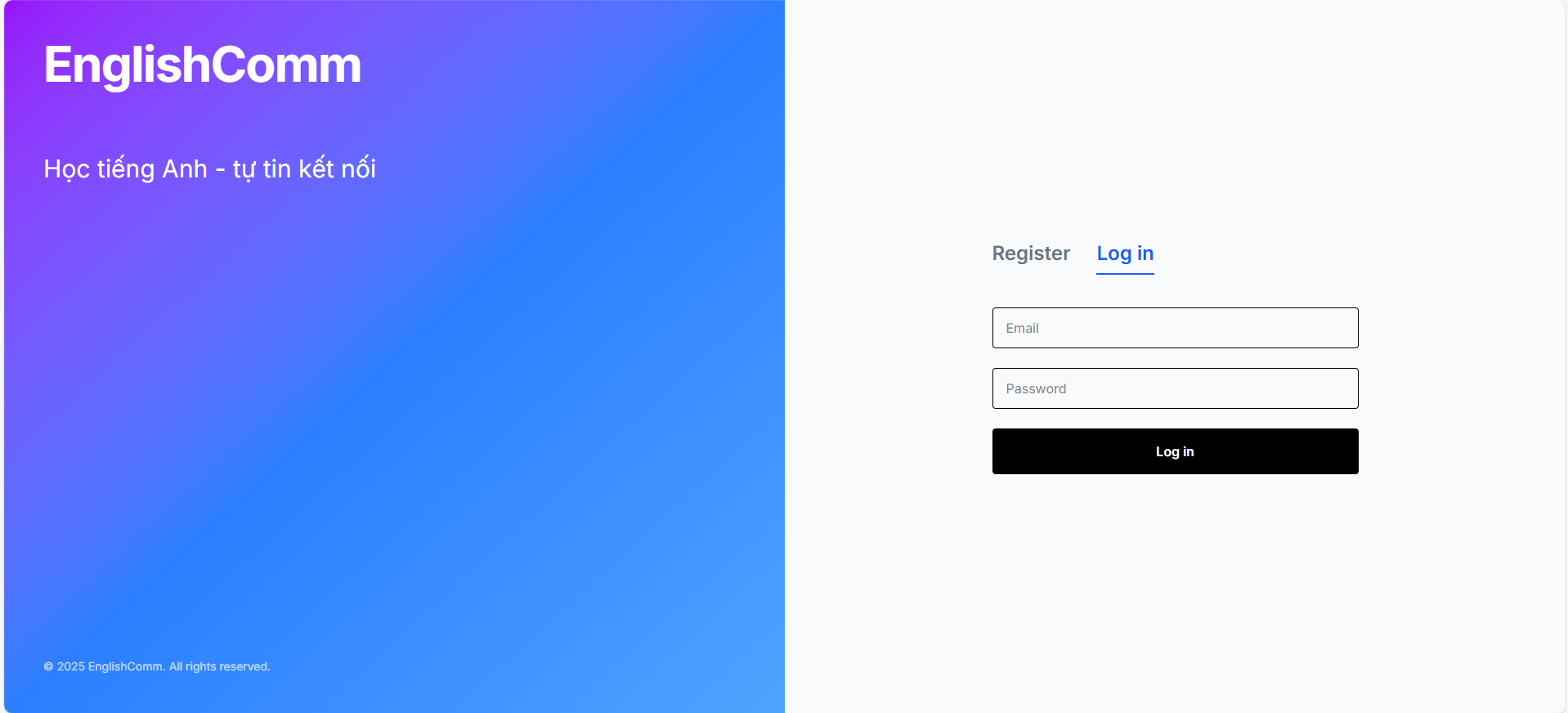
Hệ thống xác thực người dùng thông qua Token (JWT) đảm bảo tính bảo mật.

Mô tả: Form đăng ký yêu cầu các thông tin cơ bản (Tên, Email, Mật khẩu) và hỗ trợ tải lên ảnh đại diện (Avatar). Giao diện chuyển đổi linh hoạt giữa Đăng ký và Đăng nhập mà không cần tải lại trang.

Kết quả: Dữ liệu người dùng được lưu trữ an toàn trong MySQL, mật khẩu được mã hóa.



Hình 1.4 Giao diện đăng ký



Hình 1.5 Giao diện đăng nhập

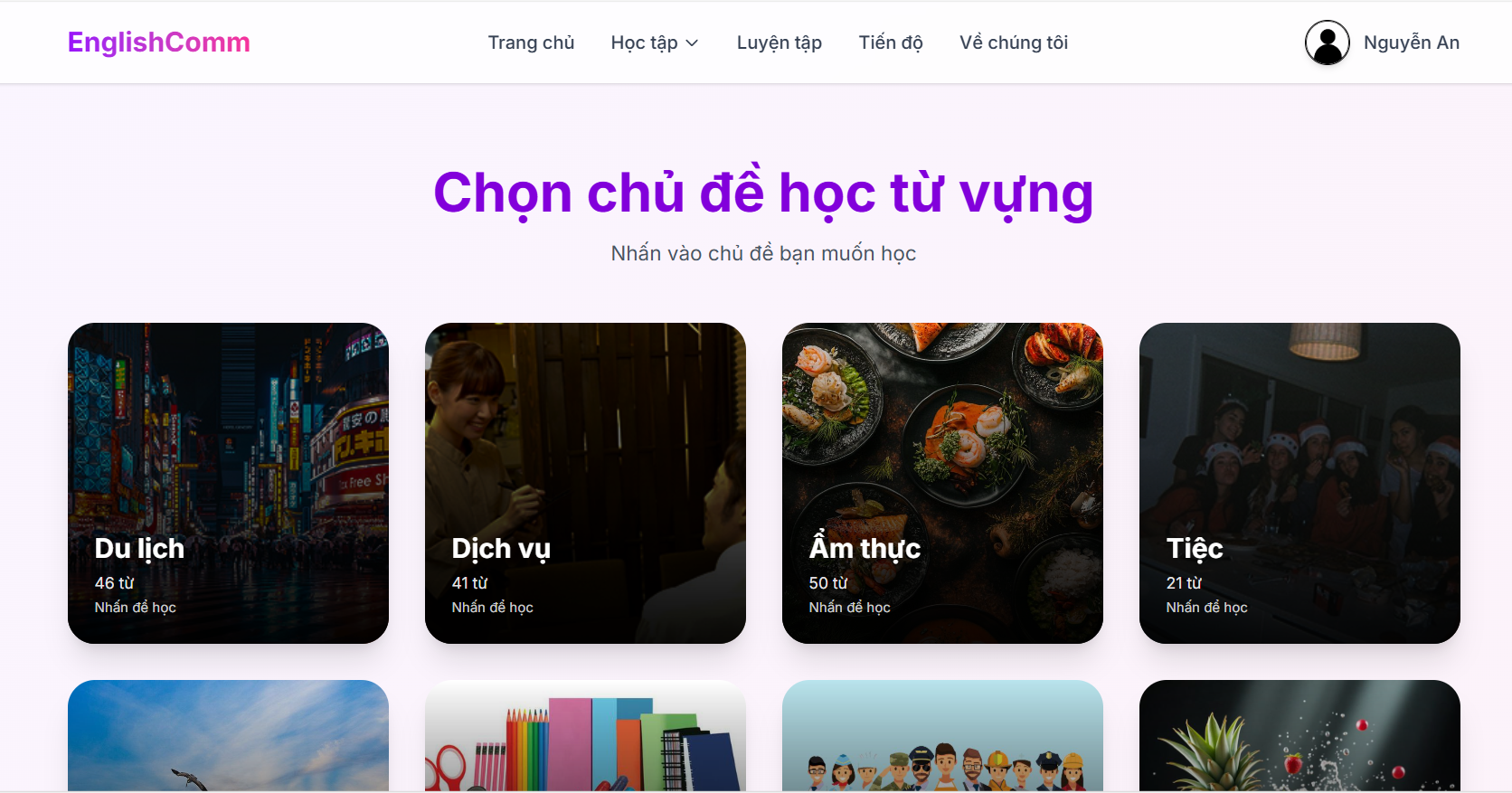
4.1.3. Chức năng học từ vựng và giao tiếp

Đây là chức năng cốt lõi của hệ thống, giúp người dùng tiếp thu kiến thức mới.

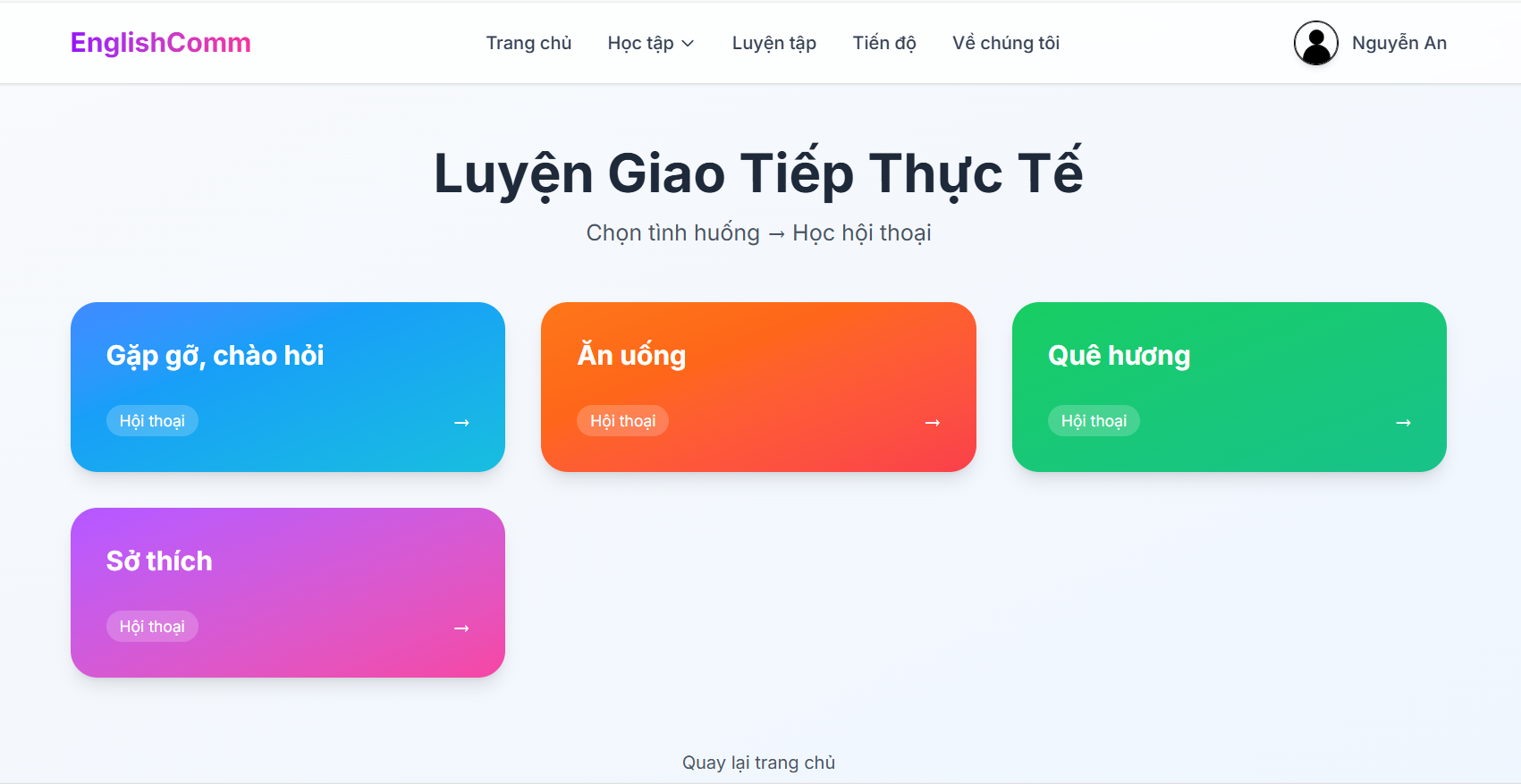
Học từ vựng: Hiển thị danh sách từ vựng theo chủ đề dưới dạng Flashcard gồm từ tiếng Anh, nghĩa tiếng Việt, ví dụ câu liên quan đến từ vựng, phiên âm và phát âm.

Học giao tiếp: Cung cấp danh sách các mẫu câu hội thoại thông dụng kèm theo cấu trúc ngữ pháp và video hội thoại.

Khi chọn học từ vựng hoặc chủ đề, người học cần chọn chủ đề mà mình muốn học và trao dồi từ vựng hoặc giao tiếp.

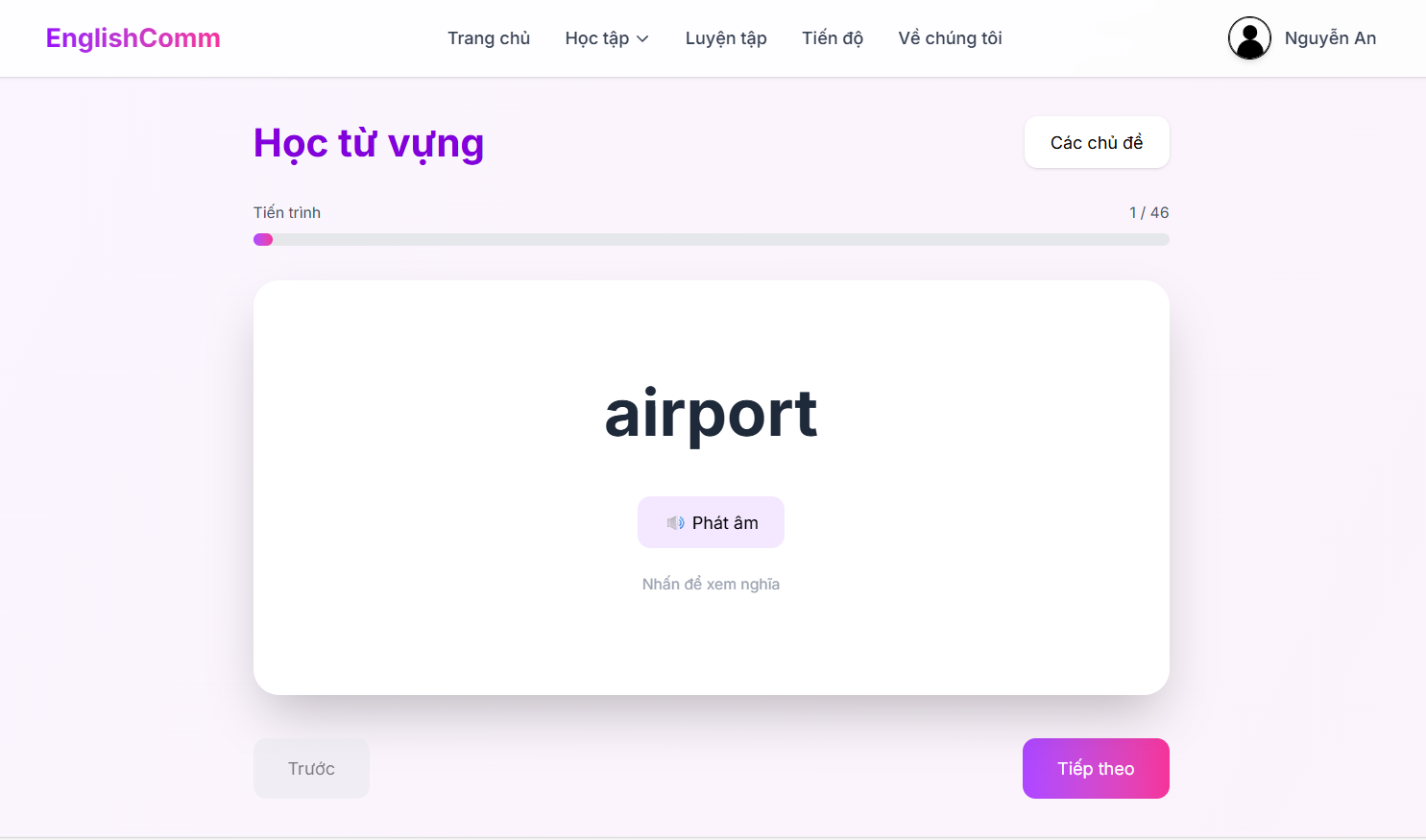


Hình 1.6 Giao diện chọn chủ đề học từ vựng



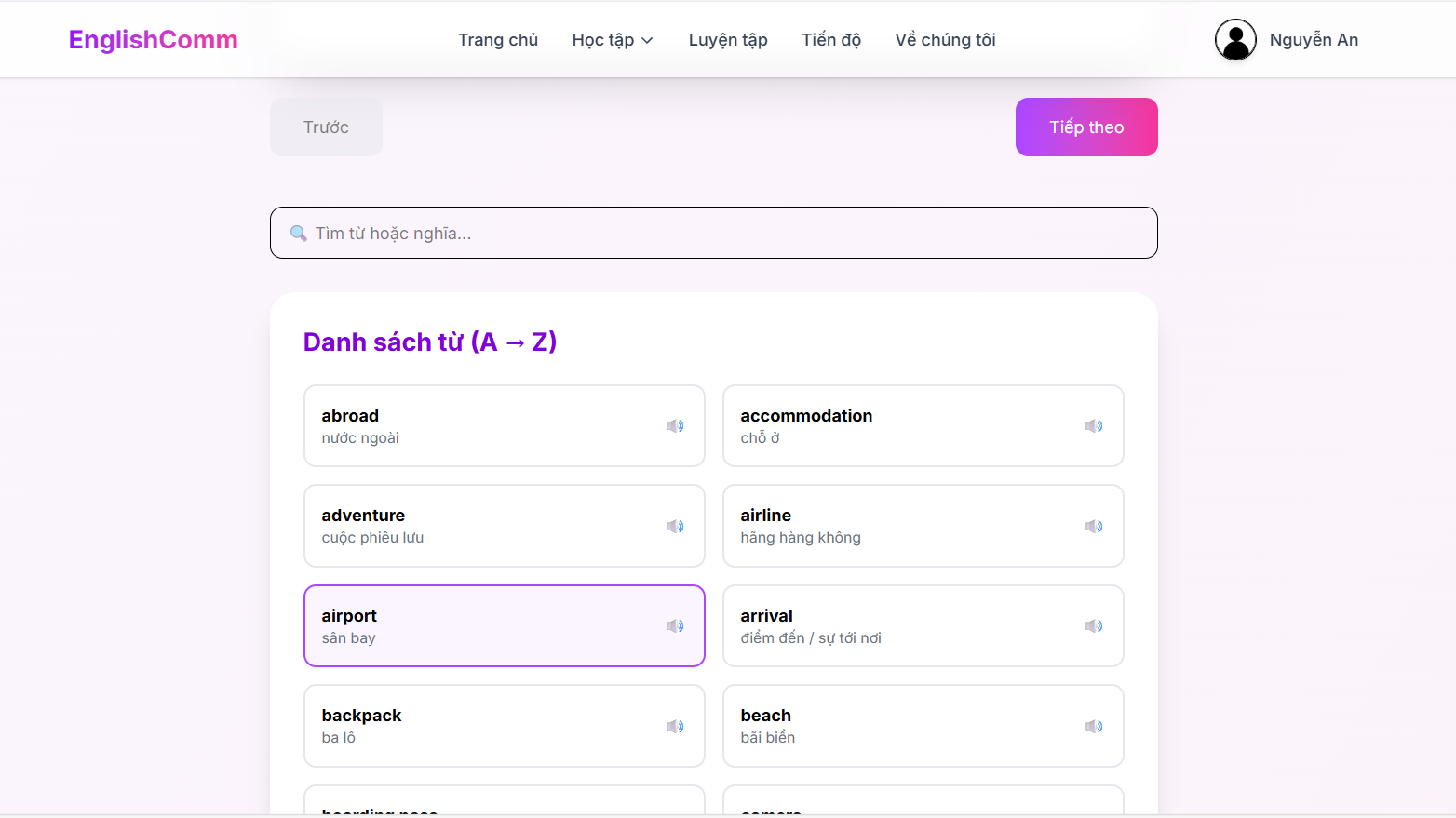
Hình 1.7 Giao diện chọn chủ đề học giao tiếp

Sau khi chọn chủ đề, người học sẽ học các từ vựng hoặc mẫu giao giao tiếp liên quan đến chủ đề đó.



Hình 1.8 Trang học từ vựng bằng Flashcard

Bên dưới Flashcard, người học có thể xem danh sách tất cả từ vựng trong chủ đề này được sắp xếp theo tứ tự thừ A đến Z theo bảng chữ cái, người học có thể tìm kiếm từ mình muốn học bằng tiếng Anh hoặc nghĩa tiếng Việt trên thanh tìm kiếm. Người học có thể nhấn nghe phát âm từ vựng.



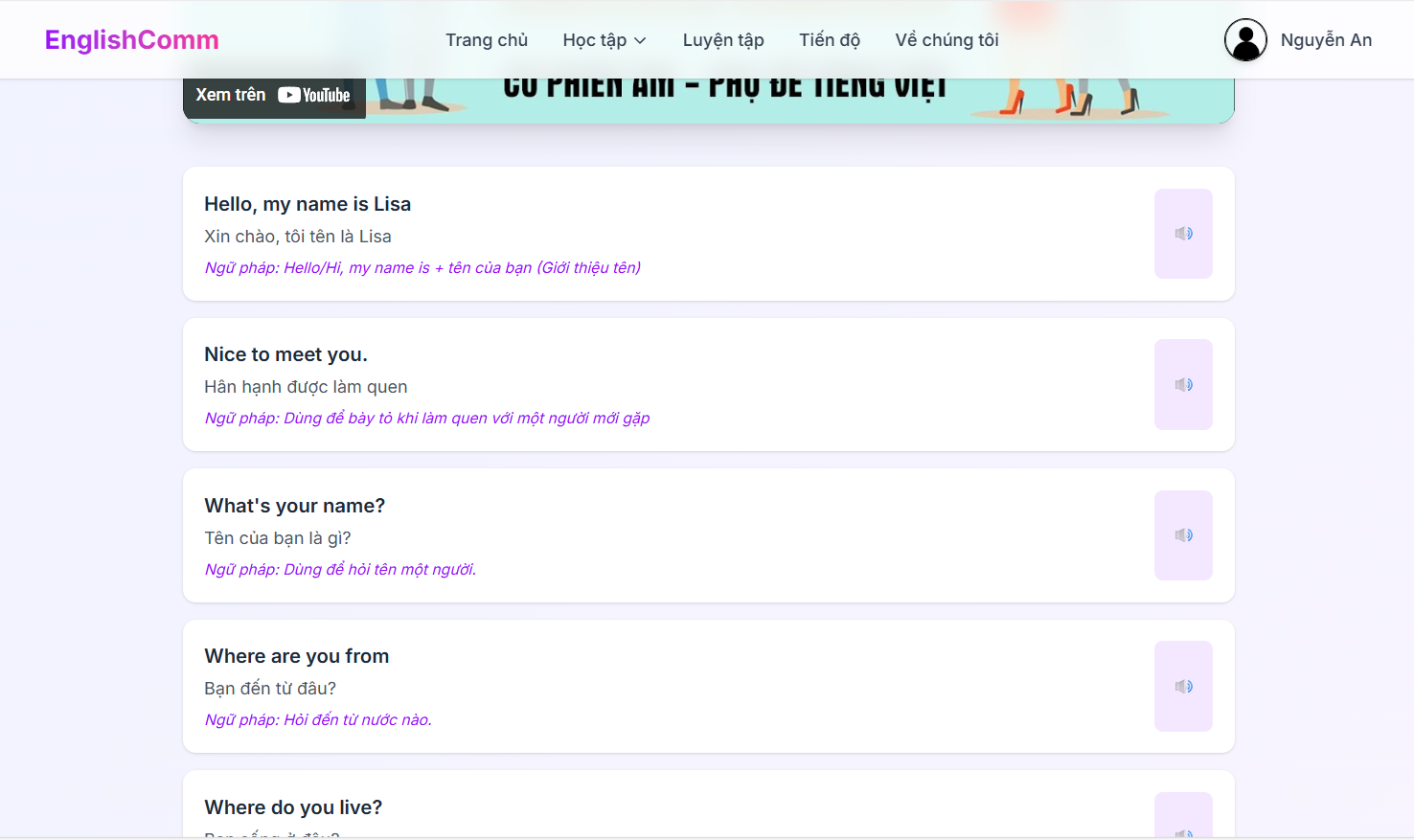
Hình 1.9 Danh sách từ vựng

Người học xem video để học giao tiếp và học các mẫu câu trọng tâm của chủ đề ngay bên dưới.



Hình 1.10 Trang học giao tiếp

Người dùng có thể nghe phát âm các mẫu câu.



Hình 2.1 Các mẫu câu giao tiếp

4.1.4 Chức năng luyện tập

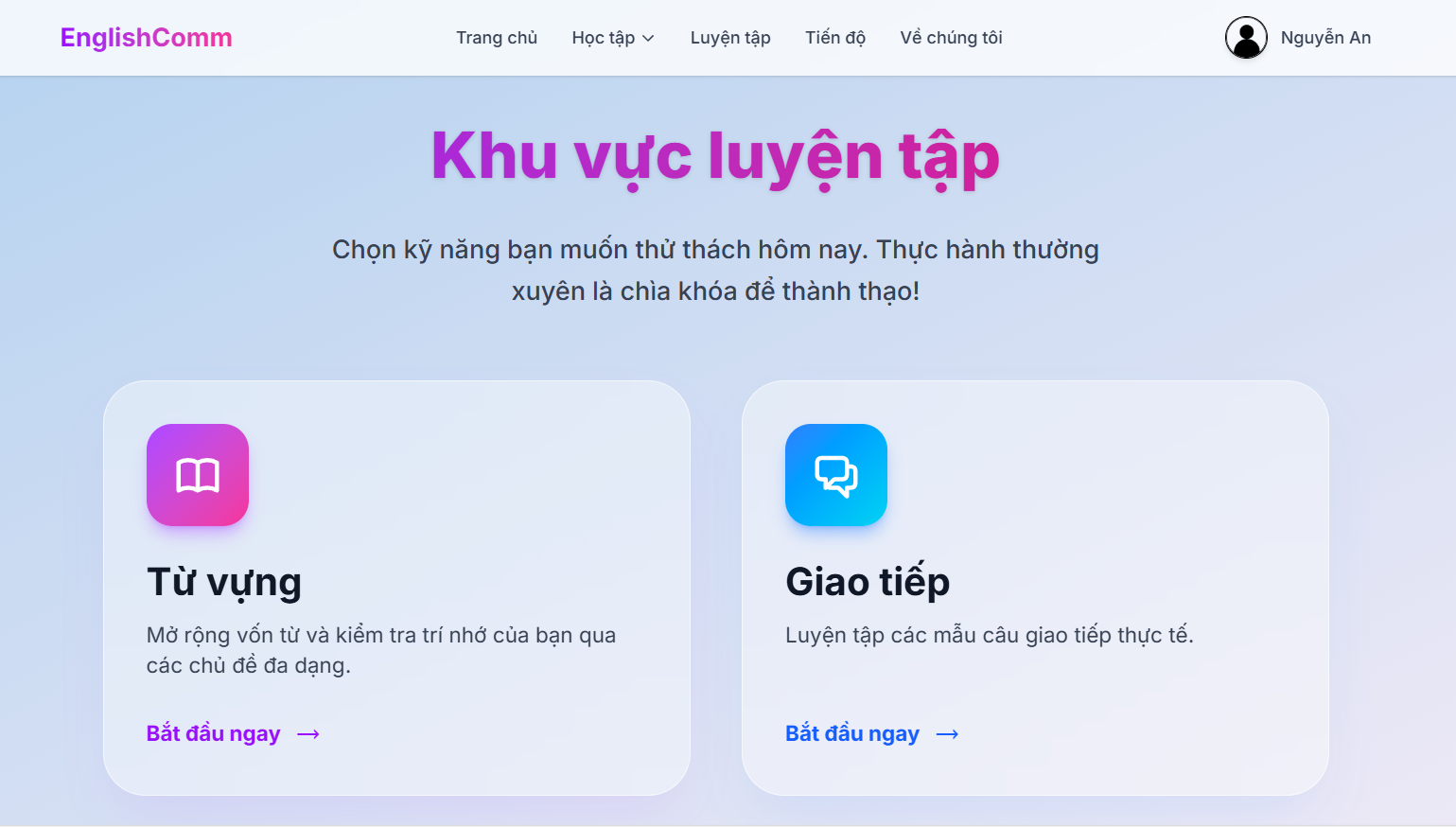
Hệ thống cung cấp bài kiểm tra trắc nghiệm để đánh giá kiến thức.

Cơ chế: Hệ thống lấy ngẫu nhiên 10 câu hỏi từ cơ sở dữ liệu.

Giao diện: Hiển thị câu hỏi, 4 đáp án lựa chọn, và thanh điều hướng câu hỏi. Người dùng có thể đánh dấu (Flag) các câu chưa chắc chắn.

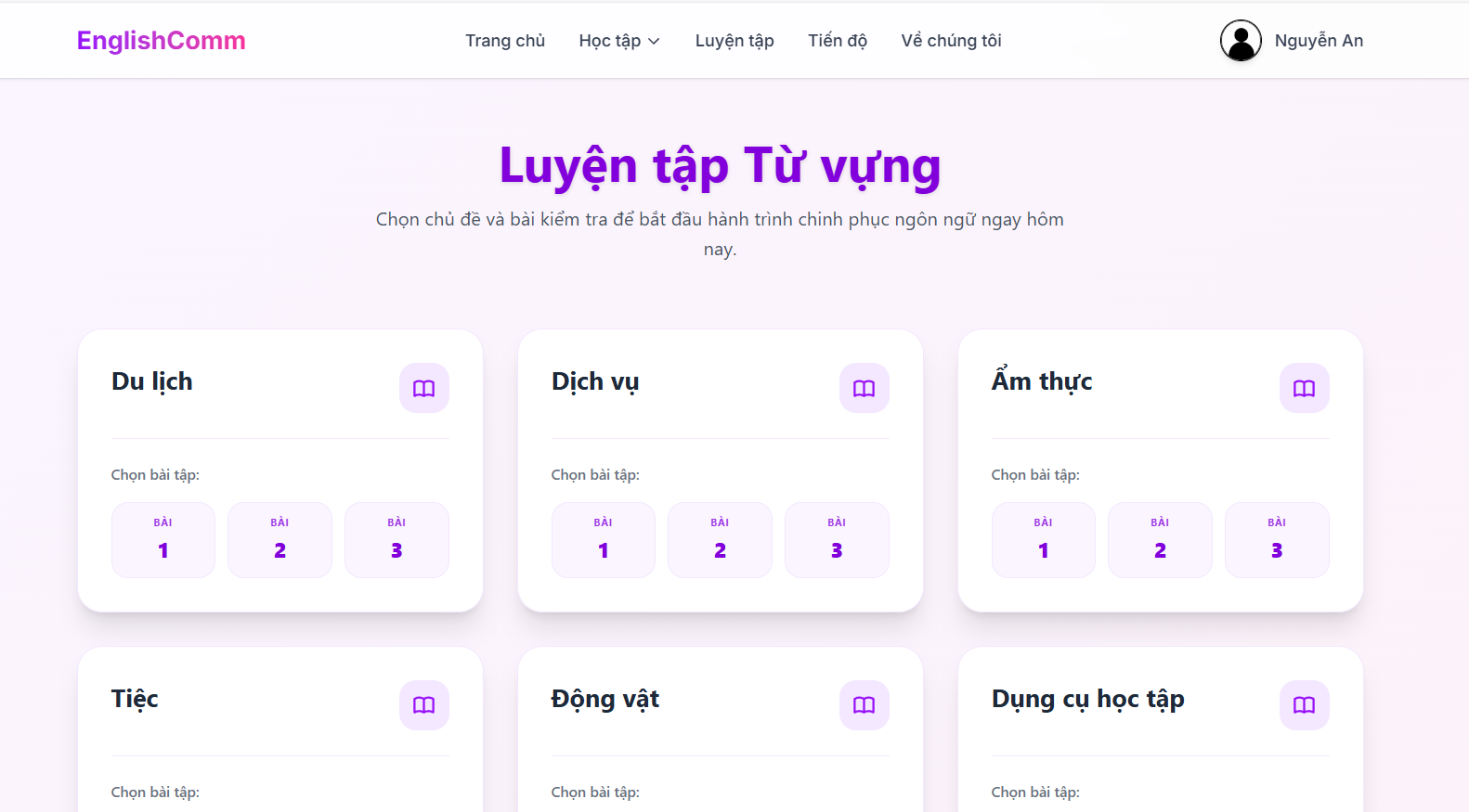
Kết quả: Sau khi nộp bài, hệ thống hiển thị ngay điểm số, số câu đúng/sai và đáp án chi tiết cho từng câu.

Tương tự, người học chọn kỹ năng muốn luyện tập sau khi học.

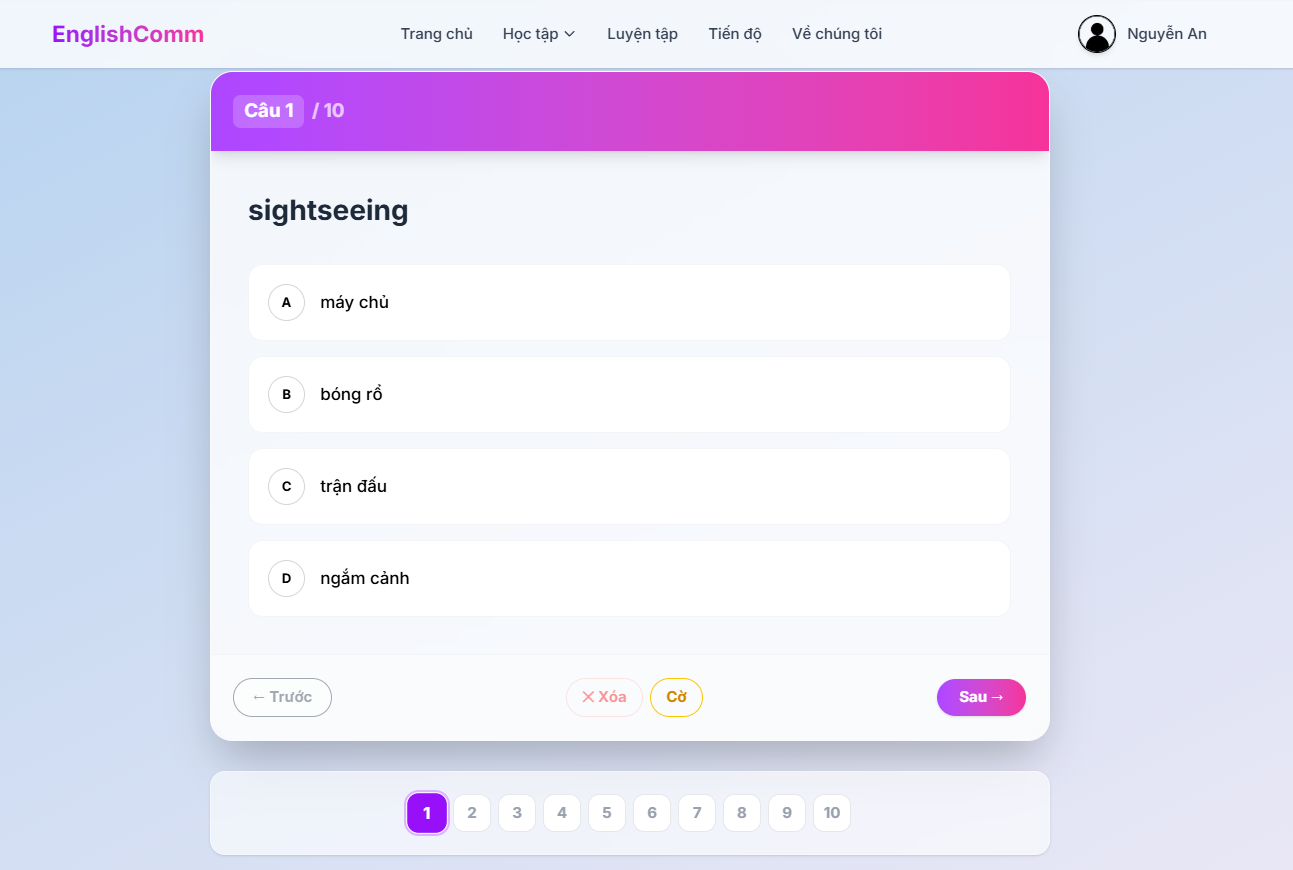


Hình 2.2 Chọn kỹ năng luyện tập

Người học chọn bài tập và chủ đề mà mình muốn.

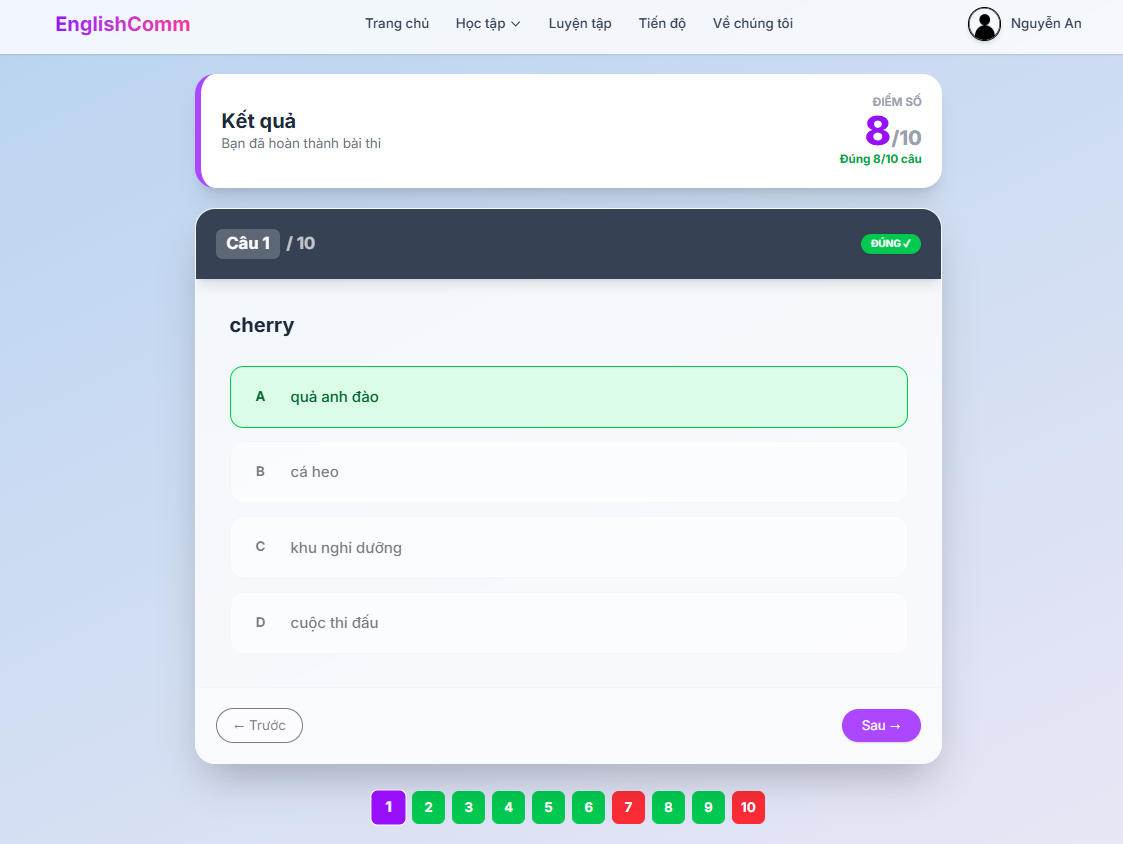


Hình 2.3 Chọn chủ đề và bài tập



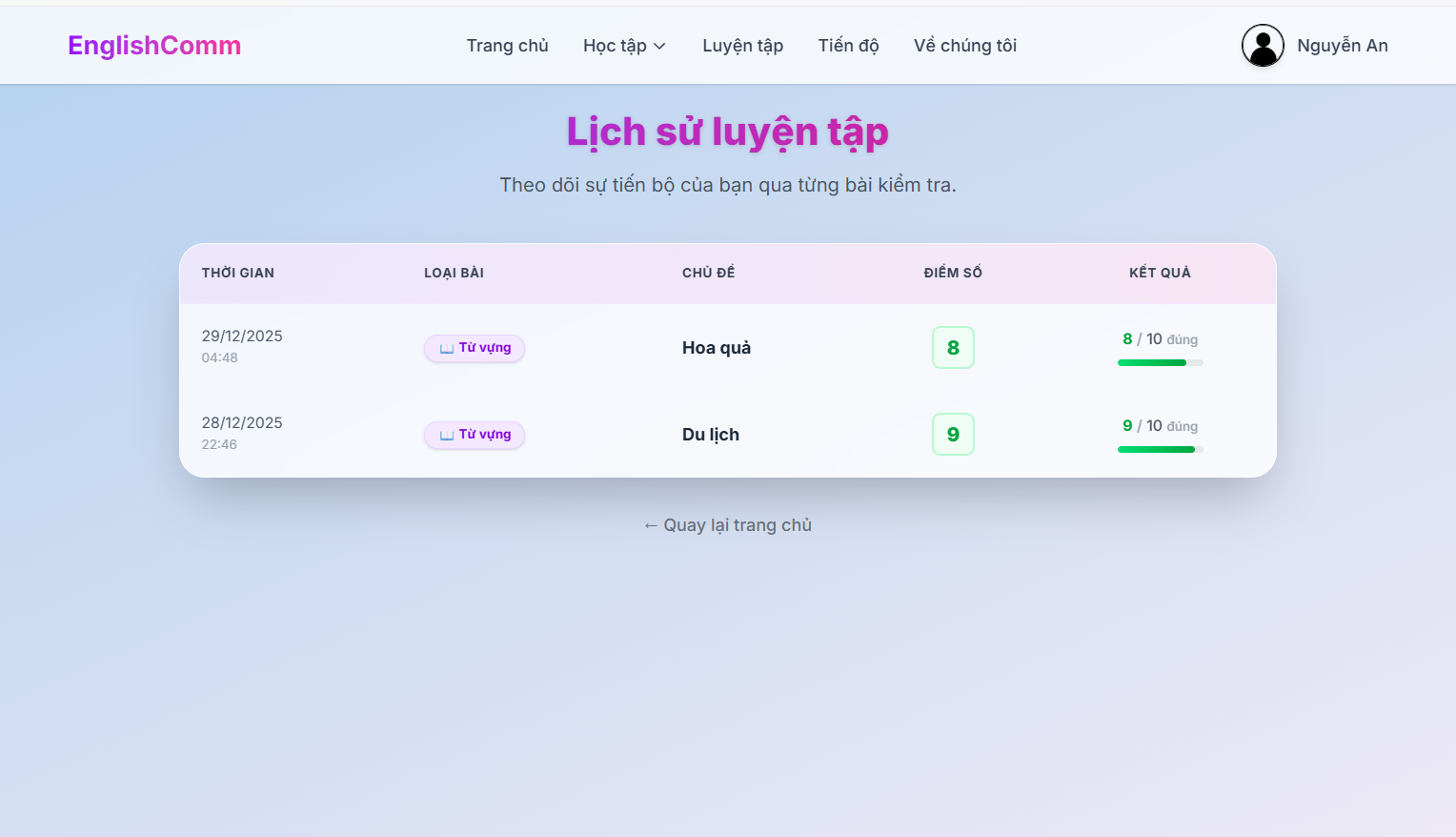
Hình 2.4 Làm bài tập

Sau khi làm bài xong, hệ thống hiển thị kết quả bài làm.



Hình 2.5 Trang kết quả bài làm

Người học có thể xem lịch sử làm bài của mình ở trang tiến độ.



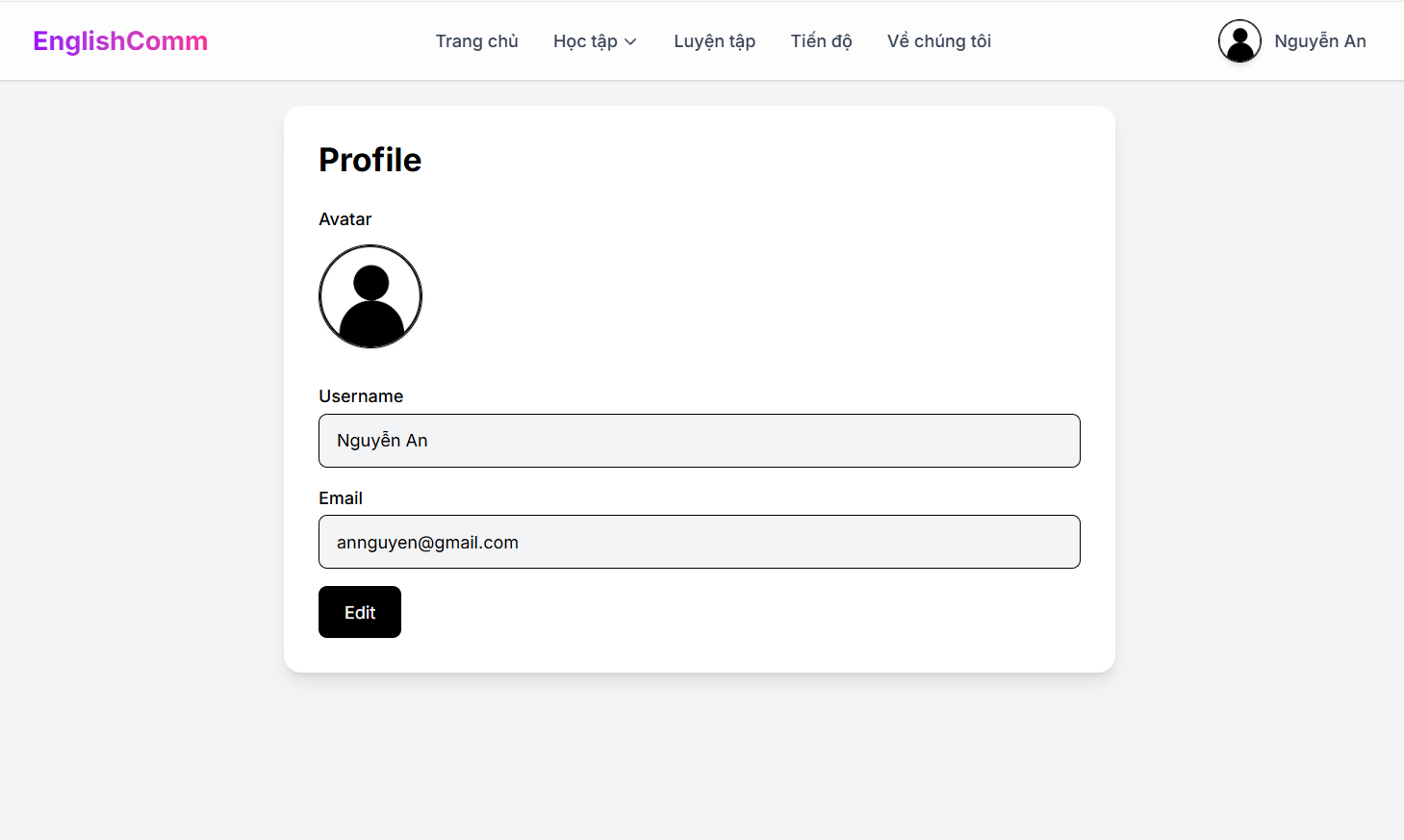
Hình 2.6 Trang tiến độ học tập

4.1.5 Trang hồ sơ cá nhân

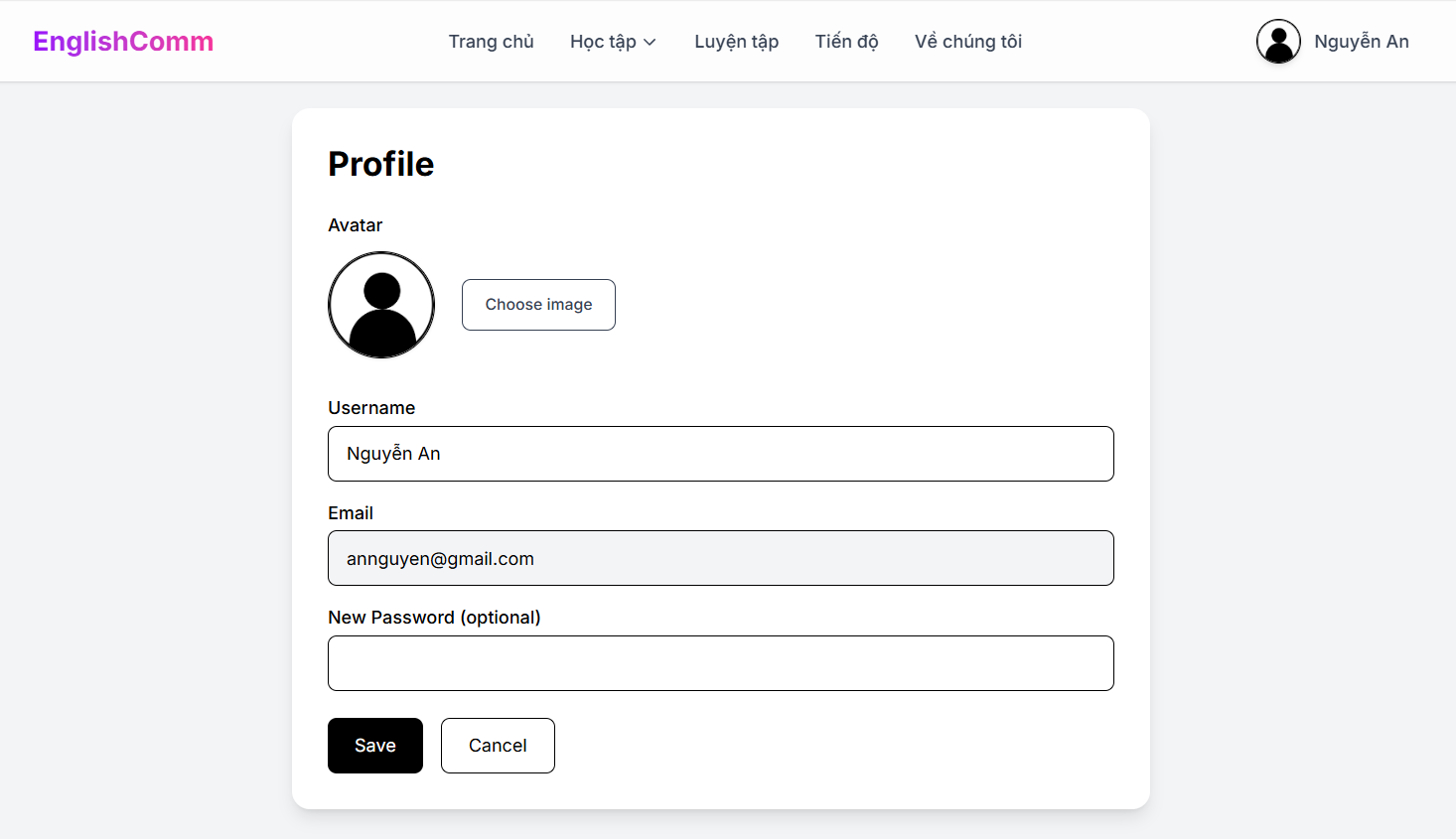
Trang Hồ sơ cá nhân là nơi người dùng có thể xem và quản lý thông tin tài khoản của mình. Giao diện được thiết kế tối giản, tập trung vào các thông tin quan trọng nhất để người dùng dễ dàng theo dõi.

Hiển thị thông tin: Hệ thống tự động tải thông tin người dùng từ cơ sở dữ liệu thông qua API (bao gồm Tên hiển thị - Username, Email đăng ký và Ảnh đại diện - Avatar) và điền sẵn vào các trường tương ứng.

Cập nhật thông tin: Nút "Edit" cho phép người dùng chuyển sang chế độ chỉnh sửa để thay đổi tên hiển thị hoặc cập nhật các thông tin cá nhân khác.



Hình 2.7 Trang hồ sơ cá nhân



Hình 2.8 Chỉnh sửa hồ sơ

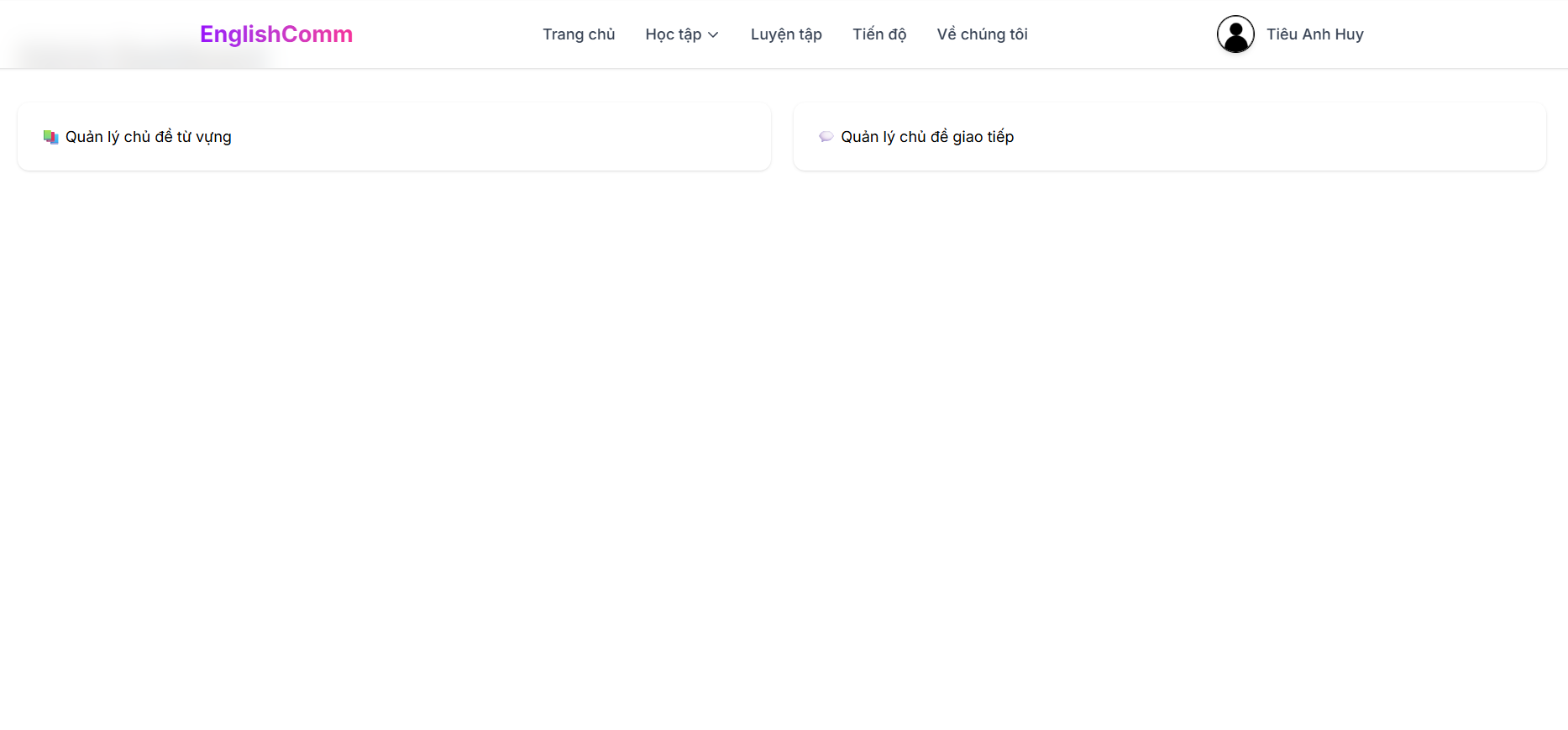
4.1.6 Trang quản trị

Dành riêng cho tài khoản có quyền Admin để quản lý nội dung hệ thống.

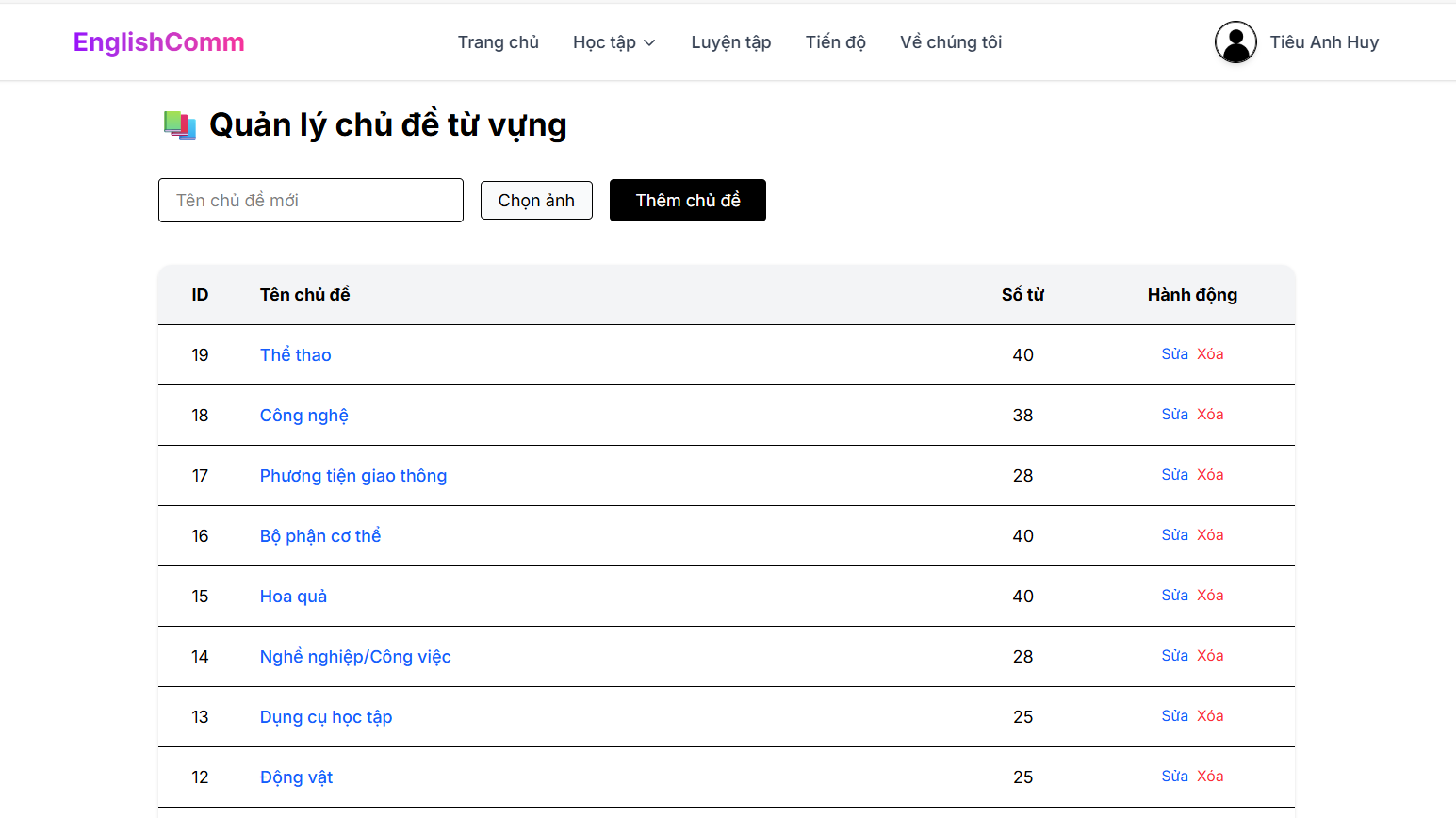
Quản lý chủ đề: Cho phép Thêm/Sửa/Xóa các chủ đề từ vựng và giao tiếp.

Quản lý nội dung chi tiết: Giao diện cho phép thêm từng từ vựng hoặc câu hội thoại vào chủ đề tương ứng.

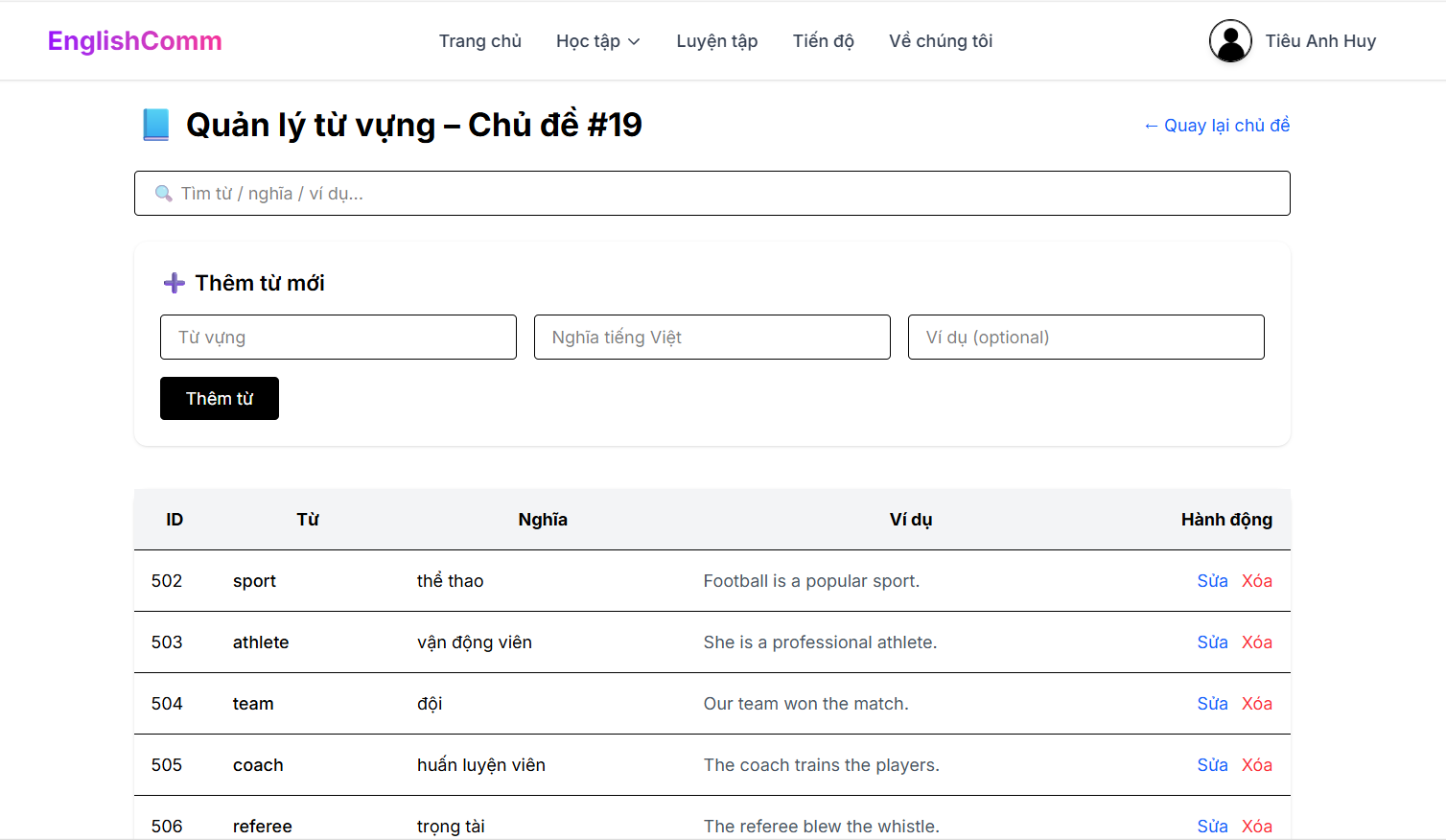
Đặc điểm: Sử dụng bảng (Table) trực quan, thao tác cập nhật dữ liệu diễn ra tức thời (Real-time update) mà không cần reload trang.



Hình 2.9 Trang chọn quản lý chủ đề từ vựng hoặc giao tiếp



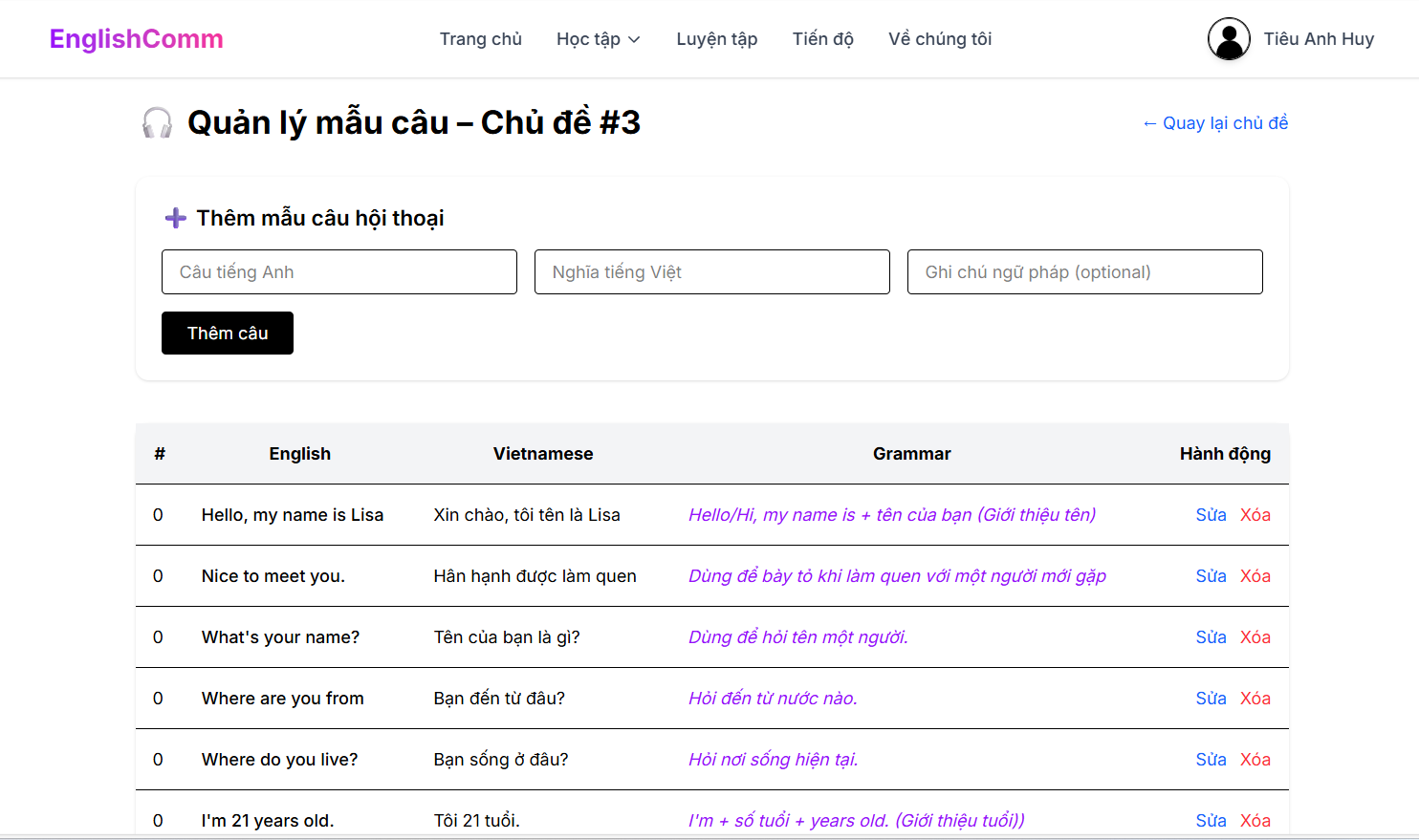
Hình 2.10 Quản lý chủ đề từ vựng



Hình 3.1 Quản lý từ vựng



Hình 3.2 Quản lý chủ đề giao tiếp



Hình 3.3 Quản lý mẫu câu giao tiếp

4.4 Kết quả nghiên cứu

Kết quả nghiên cứu cho thấy ứng dụng web EnglishComm đã đáp ứng được các yêu cầu đặt ra trong đồ án:

- Xây dựng thành công quy trình trọn vẹn từ Đăng ký -> Học tập -> Kiểm tra -> Lưu kết quả.

- Giao diện hiện đại, thân thiện và dễ sử dụng.

- Hệ thống quản trị (Admin) hoạt động hiệu quả, giúp dễ dàng mở rộng nội dung bài học trong tương lai.

Tuy nhiên, hệ thống vẫn còn một số hạn chế như chưa có tính năng kiểm tra phát âm bằng AI hay bảng xếp hạng người dùng (Leaderboard). Đây sẽ là tiền đề cho các hướng phát triển tiếp theo.

CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1 Kết luận

Sau quá trình nghiên cứu, phân tích, thiết kế và cài đặt, đồ án xây dựng ứng dụng web học tiếng Anh giao tiếp và từ vựng (EnglishComm) đã hoàn thành và đạt được những kết quả cụ thể sau:

**Về mặt sản phẩm:**

Xây dựng thành công hệ thống ứng dụng web hoàn chỉnh theo kiến trúc Client-Server.

Hiện thực hóa đầy đủ các chức năng quan trọng cho người dùng cuối: Đăng ký/Đăng nhập bảo mật, Học từ vựng theo chủ đề (Flashcard), Học mẫu câu giao tiếp, Làm bài kiểm tra trắc nghiệm (Random Test) và Xem lịch sử tiến độ.

Xây dựng thành công phân hệ Quản trị (Admin) cho phép quản lý, cập nhật dữ liệu (Thêm/Sửa/Xóa chủ đề, từ vựng, câu hỏi) một cách linh hoạt mà không cần can thiệp vào mã nguồn.

**Về mặt kỹ thuật:**

Ứng dụng thành công các công nghệ lập trình web hiện đại: ReactJS (Frontend), Node.js/Express (Backend) và MySQL (Cơ sở dữ liệu).

Xây dựng được giao diện người dùng (UI/UX) thân thiện, có khả năng tương thích trên nhiều thiết bị nhờ Tailwind CSS.

Áp dụng quy trình xác thực người dùng an toàn thông qua JWT (JSON Web Token).

**Ý nghĩa thực tiễn:**

Tạo ra một môi trường tự học tiếng Anh miễn phí, trực quan và dễ tiếp cận cho người mới bắt đầu.

Cung cấp công cụ tự đánh giá năng lực thông qua hệ thống bài tập trắc nghiệm khách quan, giúp người học theo dõi sát sao sự tiến bộ của bản thân.

5.2 Hướng phát triển

Mặc dù hệ thống đã đáp ứng được các yêu cầu cơ bản đặt ra ban đầu, nhưng để ứng dụng trở nên hoàn thiện hơn và có khả năng cạnh tranh cao trong thực tế, đồ án đề xuất các hướng nghiên cứu và phát triển tiếp theo như sau:

- Tích hợp Trí tuệ nhân tạo (AI) vào việc học:

* Nghiên cứu tích hợp các API nhận diện giọng nói (Speech-to-Text) để xây dựng tính năng kiểm tra phát âm của người dùng, giúp cải thiện kỹ năng nói thực tế thay vì chỉ học qua văn bản.
* Sử dụng AI để gợi ý lộ trình học cá nhân hóa dựa trên kết quả các bài kiểm tra trước đó của người dùng.
* Phát triển tính năng Gamification (Trò chơi hóa):
* Xây dựng hệ thống bảng xếp hạng (Leaderboard) để kích thích sự thi đua giữa các người dùng.

- Mở rộng nền tảng:

* Phát triển phiên bản ứng dụng di động sử dụng React Native để tận dụng lại mã nguồn ReactJS hiện có, giúp người dùng học tập mọi lúc mọi nơi thuận tiện hơn.

- Tăng cường tính tương tác cộng đồng:

* Phát triển thêm tính năng diễn đàn thảo luận hoặc phần bình luận dưới mỗi bài học để người dùng có thể trao đổi kiến thức và hỗ trợ lẫn nhau.
* Xây dựng tính năng "Thách đấu" (Battle) từ vựng trực tuyến giữa hai người dùng theo thời gian thực (Real-time) sử dụng công nghệ Socket.io.

**DANH MỤC TÀI LIỆU THAM KHẢO**

[1] admin@, “HTML là gì? Tìm hiểu về ngôn ngữ HTML từ A đến Z,” FPT Smart Cloud, Dec. 24, 2024. <https://fptcloud.com/html-la-gi/>

[2] TakiDev, “Tailwind CSS là gì? Tổng hợp A–Z thông tin chi tiết,” TakiDev, truy cập: Nov. 30, 2025. Available: <https://takidev.com/tailwind-css-la-gi-tong-hop-a-z-thong-tin-chi-tiet/>

[3] Ecxo.vn, “ReactJS là gì? Lý do trở thành lựa chọn hàng đầu cho phát triển Web,” Ecxo.vn. [Online]. Available: https://ecxo.vn/reactjs-la-gi/

[4] Vietnix, “NodeJS là gì? Tổng quan kiến thức về Node.JS chi tiết từ A–Z,” Vietnix, Jun. 16, 2025. [Online]. Available: https://vietnix.vn/nodejs-la-gi/

[5] CodeGym, “ExpressJS là gì? Hướng dẫn cách học ExpressJS chi tiết,” CodeGym, [Online]. Available: https://codegym.vn/blog/expressjs-la-gi-huong-dan-cach-hoc-expressjs-chi-tiet/

[6] Vietnix, “MySQL là gì? Tổng quan về MySQL chi tiết từ A–Z,” Vietnix, Jun. 16, 2025. [Online]. Available: https://vietnix.vn/mysql-la-gi/