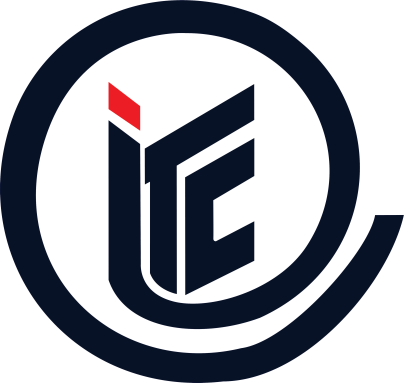
# BỘ LAO ĐỘNG THƯƠNG BINH & XÃ HỘI TRƯỜNG CAO ĐẲNG CÔNG NGHỆ THÔNG TIN TP.HCM KHOA CÔNG NGHỆ THÔNG TIN-ĐIỆN TỬ

****

**ĐỒ ÁN: ỨNG DỤNG WEB  
NGÀNH: CÔNG NGHỆ THÔNG TIN**

**ĐỀ TÀI: XÂY DỰNG WEBSITE ỨNG DỤNG HỌC REACT**

**Giảng viên hướng dẫn: Lưu Giang Nam**

**Sinh viên thực hiện: Bùi Võ Văn Khoa 501230305**

**Phan Lu Vỹ 501230123**

**Nguyễn Tuấn Cường 501230162**

**Dương Trạch Đông 501230121**

**Tên nhóm học phần: 501\_UDW\_HK2.2425\_CD23CT.N1**

**Học kỳ: 2 Năm học: 2024-2025**

**Tp.Hồ Chí Minh, tháng 12 năm 2024**

**LỜI MỞ ĐẦU**

Trong bối cảnh công nghệ thông tin phát triển mạnh mẽ như hiện nay, việc ứng dụng các công nghệ mới vào việc học tập và phát triển kỹ năng lập trình là một xu hướng tất yếu. React, một thư viện JavaScript phổ biến, đang được sử dụng rộng rãi trong việc xây dựng các giao diện người dùng hiện đại và tương tác cao. Việc học và thành thạo React không chỉ mở ra cơ hội nghề nghiệp hấp dẫn mà còn giúp người học nâng cao khả năng tư duy logic và giải quyết vấn đề.

Nhận thức được tầm quan trọng của việc thực hành và áp dụng kiến thức, chúng tôi lựa chọn đề tài **“Xây dựng website ứng dụng học React”** để xây dựng một nền tảng học tập trực tuyến, hiện đại và dễ sử dụng. Website này sẽ tích hợp các bài học, bài tập thực hành, và các công cụ hỗ trợ giúp người dùng có thể tiếp cận và nắm vững kiến thức về React một cách hiệu quả. Dự án này không chỉ mang tính thực tiễn cao mà còn góp phần thúc đẩy việc học tập và ứng dụng công nghệ React trong cộng đồng lập trình.

**LỜI CẢM ƠN**

Để hoàn thành báo cáo đồ án này, chúng tôi xin gửi lời cảm ơn chân thành đến thầy Lưu Giang Nam đã tận tình truyền đạt cho chúng tôi những kiến thức, kinh nghiệm bổ ích, hướng dẫn, góp ý, hỗ trợ hết mình. Những nền tảng kiến thức quý giá ấy giúp ích cho chúng tôi rất nhiều trong quá trình hoàn thành đồ án, đồng thời là hành trang hữu ích cho công việc của chúng tôi. Nhờ sự giúp đỡ hết lòng của thầy mà báo cáo của chúng tôi mới được hoàn thiện như ngày hôm nay.

Tuy nhiên, thời gian ngắn ngủi với kinh nghiệm hạn hẹp của bản thân mình, chúng tôi đã cố gắng hết mình để tìm hiểu, phân tích, xây dựng hệ thống nhưng chắc chắn sẽ không tránh khỏi sự thiếu xót. Chúng tôi rất mong nhận được sự thông cảm cũng như những lời góp ý và chỉ bảo tận tình của thầy điều đó sẽ giúp chúng tôi có thêm sự dày dặn trong kinh nghiệm và kiến thức của mình.

Cuối cùng, chúng tôi xin kính chúc thầy Lưu Giang Nam – Giảng viên Trường Cao đẳng Công nghệ thông tin Thành phố Hồ Chí Minh luôn dồi dào sức khỏe và thành công trong công việc giảng dạy.

Chúng tôi xin chân thành cảm ơn!

**NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN**

Tp. HCM, ngày….tháng…..năm………  
 (chữ ký và ghi rõ họ tên)

**MỤC LỤC**

Trang

[PHẦN I. TỔNG QUAN ĐỀ TÀI 1](#_Toc187002285)

[1. Khảo sát đề tài 1](#_Toc187002286)

[**1.1.** **Tính cấp thiết** 1](#_Toc187002287)

[**1.2.** **Đặt vấn đề** 1](#_Toc187002288)

[**1.3.** **Giải pháp** 2](#_Toc187002289)

[**1.4.** **Mục tiêu đề tài** 2](#_Toc187002290)

[PHẦN II. CÁC CÔNG NGHỆ SỬ DỤNG 3](#_Toc187002291)

[I. CÔNG NGHỆ FRONTEND 3](#_Toc187002292)

[**1.** **ReactJS - Nền tảng cho giao diện người dùng hiện đại** 3](#_Toc187002293)

[**1.1.** **Kiến trúc Component-Based** 3](#_Toc187002294)

[**1.2.** **Virtual DOM và cơ chế Render** 3](#_Toc187002295)

[**1.3.** **Lifecycle và Hooks** 4](#_Toc187002296)

[**2.** **Tailwind CSS và DaisyUI ( frameword hỗ trợ css mạnh mẽ)** 4](#_Toc187002297)

[**2.1.** **Tailwind CSS - Framework Utility-First** 4](#_Toc187002298)

[**2.2.** **DaisyUI - Component Library cho Tailwind** 4](#_Toc187002299)

[**3.** **Các thư viện khác hỗ trợ React** 5](#_Toc187002300)

[**3.1.** **React Hot Toast - Hệ thống thông báo hiện đại** 5](#_Toc187002301)

[**3.1.1.** **Đặc điểm và ưu điểm** 5](#_Toc187002302)

[**3.2.** **Axios - Thư viện HTTP Client** 5](#_Toc187002303)

[**3.2.1.** **Tính năng chính** 5](#_Toc187002304)

[**3.2.2.** **Ưu điểm trong phát triển** 6](#_Toc187002305)

[**3.2.3.** **Tích hợp với React** 6](#_Toc187002306)

[II. CÔNG NGHỆ BACKEND VÀ CƠ SỞ DỮ LIỆU 7](#_Toc187002307)

[**1.** **Node.js và Express - Nền tảng Backend mạnh mẽ** 7](#_Toc187002308)

[**1.1.** **Node.js - Runtime Environment** 7](#_Toc187002309)

[**1.2.** **Express Framework** 7](#_Toc187002310)

[**2.** **MongoDB - Hệ quản trị cơ sở dữ liệu NoSQL** 7](#_Toc187002311)

[**2.1.** **Kiến trúc Document-Oriented** 7](#_Toc187002312)

[**2.2.** **Mongoose ODM (Object Document Mapping)** 8](#_Toc187002313)

[PHẦN III. CÀI ĐẶT THỰC NGHIỆM VÀ KẾT QUẢ 9](#_Toc187002314)

[1. Kiến trúc dự án 9](#_Toc187002315)

[2. Một số giao diện website 11](#_Toc187002316)

[**2.1.** **Phần quản trị** 11](#_Toc187002317)

[**2.1.1.** **Giao diện tài khoản quản trị** 12](#_Toc187002318)

[**2.1.2.** **Giao diện quản lý danh sách khóa học** 12](#_Toc187002320)

[**2.1.3.** **Giao diện quản lý danh sách bài học** 14](#_Toc187002325)

[**2.1.4.** **Giao diện quản lý các bài tập** 16](#_Toc187002330)

[**2.2.** **Phần người dùng** 18](#_Toc187002335)

[**2.2.1.** **Giao diện người dùng ở thiết bị màn hình lớn** 18](#_Toc187002336)

[**2.2.1.1.** **Giao diện đăng nhập** 18](#_Toc187002337)

[**2.2.1.2.** **Giao diện đăng ký** 19](#_Toc187002339)

[**2.2.1.3.** **Giao diện trang chủ** 19](#_Toc187002341)

[**2.2.1.4.** **Giao diện xếp hạng** 21](#_Toc187002345)

[**2.2.1.5.** **Giao diện khóa học** 21](#_Toc187002347)

[**2.2.1.6.** **Giao diện bài học** 22](#_Toc187002349)

[**2.2.1.7.** **Giao diện profile** 26](#_Toc187002355)

[**2.2.2.** **Giao diện người dùng ở thiết bị màn hình nhỏ** 26](#_Toc187002357)

[**2.2.2.1.** **Giao diện trang chủ** 26](#_Toc187002358)

[**2.2.2.2.** **Giao diện xếp hạng** 27](#_Toc187002360)

[**2.2.2.3.** **Giao diện khóa học** 27](#_Toc187002362)

[**2.2.2.4.** **Giao diện bài học** 28](#_Toc187002364)

[PHẦN IV. KẾT LUẬN 29](#_Toc187002366)

[TÀI LIỆU THAM KHẢO 30](#_Toc187002367)

**DANH MỤC CÁC HÌNH, BẢNG BIỂU VÀ SƠ ĐỒ**

Trang

[Hình 1.1 Giao diện tài khoản quản trị. 12](#_Toc187002319)

[Hình 1.2 Giao diện quản lý danh sách khóa học. 12](#_Toc187002321)

[Hình 1.3 Giao diện thêm khóa học. 13](#_Toc187002322)

[Hình 1.4 Giao diện chỉnh sửa thông tin khóa học. 13](#_Toc187002323)

[Hình 1.5 Giao diện thông báo xóa thông tin khóa học. 14](#_Toc187002324)

[Hình 1.6 Giao diện danh sách các bài học. 14](#_Toc187002326)

[Hình 1.7 Giao diện thêm bài học. 15](#_Toc187002327)

[Hình 1.8 Giao diện chỉnh sửa bài học. 15](#_Toc187002328)

[Hình 1.9 Giao diện thông báo xóa thông tin bài học. 16](#_Toc187002329)

[Hình 1.10 Giao diện danh sách bài tập. 16](#_Toc187002331)

[Hình 1.11 Giao diện thêm bài tập. 17](#_Toc187002332)

[Hình 1.12 Giao diện chỉnh sửa bài tập. 17](#_Toc187002333)

[Hình 1.13 Giao diện thông báo xóa bài tập. 18](#_Toc187002334)

[Hình 1.14 Giao diện đăng nhập. 18](#_Toc187002338)

[Hình 1.15 Giao diện đăng ký. 19](#_Toc187002340)

[Hình 1.16 Giao diện phần header. 20](#_Toc187002342)

[Hình 1.17 Giao diện phần feedback. 20](#_Toc187002343)

[Hình 1.18 Giao diện phần footer. 20](#_Toc187002344)

[Hình 1.19 Giao diện xếp hạng. 21](#_Toc187002346)

[Hình 1.20 Giao diện khóa học. 22](#_Toc187002348)

[Hình 1.21 Giao diện bài học. 23](#_Toc187002350)

[Hình 1.22 Giao diện modle bài tập. 23](#_Toc187002351)

[Hình 1.23 Giao diện chatbot. 24](#_Toc187002352)

[Hình 1.24 Giao diện tổng quan. 25](#_Toc187002353)

[Hình 1.25 Giao diện comment. 25](#_Toc187002354)

[Hình 1.26 Giao diện profile. 26](#_Toc187002356)

[Hình 1.27 Giao diện trang chủ màn hình nhỏ. 26](#_Toc187002359)

[Hình 1.28 Giao diện xếp hạng màn hình nhỏ. 27](#_Toc187002361)

[Hình 1.29 Giao diện khóa học màn hình nhỏ. 27](#_Toc187002363)

[Hình 1.30 Giao diện bài học màn hình nhỏ. 28](#_Toc187002365)

**KÝ HIỆU CÁC CỤM TỪ VIẾT TẮT**

DOM: Document Object Model

MVC: Model – View – Controller

CSS: Cascading Style Sheets

HTTP: Client: HyperText Transfer Protocol Client

XSRF: Cross-Site Request Forgery

JSON: JavaScript Object Notation

API: Application Programming Interface

URL: Uniform Resource Locator

ODM: Object Document Mapping

# PHẦN I. TỔNG QUAN ĐỀ TÀI

1. **Khảo sát đề tài**
   1. **Tính cấp thiết**

Trong bối cảnh công nghệ thông tin phát triển không ngừng, nhu cầu về các ứng dụng web hiện đại và tương tác cao ngày càng gia tăng. React, một thư viện JavaScript mạnh mẽ được phát triển bởi Facebook, đã trở thành công cụ thiết yếu cho việc phát triển giao diện người dùng. Nắm vững React không chỉ mở ra cơ hội nghề nghiệp mở rộng mà còn đóng vai trò quan trọng trong việc đáp ứng nhu cầu ngày càng cao của thị trường. Tuy nhiên, việc tiếp cận và học tập React một cách hiệu quả vẫn còn gặp nhiều khó khăn do thiếu hụt các tài liệu học tập chất lượng và bài bản. Chính vì vậy, việc xây dựng một website ứng dụng học React là một yêu cầu cấp thiết.

Website đóng vai trò cung cấp một nền tảng học tập trực tuyến, tập trung và bài bản, giúp người học dễ dàng tiếp cận kiến thức từ cơ bản đến nâng cao, đồng thời thực hành thông qua các bài tập thực tế. Việc này không chỉ giúp người học tiết kiệm thời gian và công sức mà còn góp phần nâng cao chất lượng nguồn nhân lực trong lĩnh vực công nghệ thông tin, đáp ứng nhu cầu phát triển của xã hội. Hơn thế nữa, trong bối cảnh cạnh tranh khốc liệt giữa các nền tảng học tập trực tuyến, một website được thiết kế chuyên nghiệp và cung cấp nội dung chất lượng sẽ là yếu tố then chốt thu hút người học và khẳng định vị thế của nền tảng.

* 1. **Đặt vấn đề**

Thời đại công nghệ số đang bùng nổ, kéo theo nhu cầu về lập trình web ngày càng tăng cao. ReactJS, một thư viện JavaScript mạnh mẽ, đã trở thành công cụ không thể thiếu cho việc xây dựng các giao diện người dùng hiện đại, tương tác và hiệu suất cao. Nắm vững ReactJS mở ra cánh cửa đến vô vàn cơ hội nghề nghiệp hấp dẫn trong lĩnh vực phát triển web. Tuy nhiên, việc học ReactJS hiệu quả lại không hề đơn giản. Người học thường gặp khó khăn trong việc tiếp cận kiến thức một cách có hệ thống, thiếu môi trường thực hành thực tế và khó khăn trong việc tìm kiếm tài liệu học tập chất lượng.

Hiện nay, nhiều người học ReactJS phải tự mày mò qua các tài liệu rời rạc trên mạng, các video hướng dẫn chưa có sự chọn lọc, hoặc tham gia các khóa học offline với chi phí cao và thời gian học tập cố định. Cách học này thường thiếu tính hệ thống, gây khó khăn cho việc nắm bắt kiến thức tổng quan và áp dụng vào thực tế. Hãy thử tưởng tượng, một người mới bắt đầu học ReactJS phải lang thang giữa hàng trăm nghìn kết quả tìm kiếm trên Google, khó khăn phân biệt đâu là nguồn tài liệu đáng tin cậy, đâu là kiến thức cốt lõi cần nắm vững. Hoặc việc phải tham gia một lớp học đông người, khó có thể đặt câu hỏi và nhận được sự hỗ trợ kịp thời từ giảng viên.

Trong bối cảnh đó, việc xây dựng một website ứng dụng học ReactJS trở thành một giải pháp tất yếu, đáp ứng nhu cầu học tập ngày càng cao của cộng đồng lập trình. Website này sẽ cung cấp một lộ trình học tập bài bản, từ cơ bản đến nâng cao, với các bài giảng video chất lượng, bài tập thực hành đa dạng và hệ thống đánh giá tiến độ học tập. Người học có thể học mọi lúc, mọi nơi, theo tốc độ của riêng mình, và nhận được sự hỗ trợ từ cộng đồng học tập trực tuyến.

Xuất phát từ những thực tế trên và nhận thấy tiềm năng to lớn của việc học tập trực tuyến, đề tài “Xây dựng website ứng dụng học ReactJS” được thực hiện nhằm cung cấp một nền tảng học tập hiện đại, hiệu quả, giúp người học dễ dàng tiếp cận và chinh phục công nghệ ReactJS, từ đó mở ra cơ hội nghề nghiệp rộng mở trong lĩnh vực công nghệ thông tin.

* 1. **Giải pháp**

Chúng tôi đã đề xuất giải pháp xây dựng một website ứng dụng học ReactJS, một nền tảng học tập trực tuyến toàn diện, nhằm giải quyết các khó khăn hiện tại trong việc học tập và tiếp cận công nghệ ReactJS. Website này được thiết kế với mục tiêu mang đến cho người học một trải nghiệm học tập hiệu quả, tương tác cao và dễ dàng tiếp cận, bất kể trình độ hiện tại của họ.

* 1. **Mục tiêu đề tài**

Đề tài “Xây dựng website ứng dụng học React” hướng đến mục tiêu tạo ra một nền tảng học tập trực tuyến và dễ dàng tiếp cận, giúp mọi người có thể học tập một cách hiệu quả, tiết kiệm chi phí và thời gian . Cung cấp một lộ trình học tập rõ ràng, cập nhập kiến thức và công nghệ mới nhất.

# PHẦN II. CÁC CÔNG NGHỆ SỬ DỤNG

## I. CÔNG NGHỆ FRONTEND

1. **ReactJS - Nền tảng cho giao diện người dùng hiện đại**

ReactJS đã trở thành một trong những công nghệ frontend được ưa chuộng nhất trong phát triển web hiện đại. Được phát triển và duy trì bởi Facebook (nay là Meta), React không chỉ đơn thuần là một thư viện JavaScript, mà còn là một hệ sinh thái hoàn chỉnh cho phát triển giao diện người dùng.

* 1. **Kiến trúc Component-Based**

ReactJS chia giao diện thành các component, mỗi component đảm nhận một phần cụ thể của giao diện người dùng.:

* Tính module hóa cao: Mỗi component có thể được phát triển, kiểm thử và bảo trì độc lập, giúp giảm thiểu sự phụ thuộc và tăng tính ổn định của hệ thống.
* Tái sử dụng code hiệu quả: Components có thể được tái sử dụng trong nhiều phần khác nhau của ứng dụng, thậm chí across các dự án khác nhau.
* Quản lý State và Props: React cung cấp cơ chế quản lý dữ liệu thông qua state và props, giúp luồng dữ liệu trong ứng dụng trở nên minh bạch và dễ kiểm soát.

**Ưu điểm nổi bật:**

* Tái sử dụng ReaderComponent cho tất cả các truyện, giúp tiết công sức khi mở rộng hệ thống
* Dễ dàng bảo trì và nâng cấp từng phần mà không ảnh hưởng đến toàn bộ ứng dụng
  1. **Virtual DOM và cơ chế Render**

React sử dụng Virtual DOM như một lớp trừu tượng giữa developer và Real DOM. Đây là một trong những đổi mới quan trọng nhất của React. Cơ chế này hoạt động thông qua các bước:

* Tạo Virtual DOM: Khi có thay đổi trong ứng dụng, React tạo ra một bản sao của DOM thật dưới dạng JavaScript object.
* Batch Update: Các thay đổi được gom nhóm và thực hiện cùng một lúc, giảm thiểu số lần cập nhật DOM thật.
* Efficient Rendering: Chỉ những phần thực sự thay đổi mới được cập nhật trong DOM thật.
  1. **Lifecycle và Hooks**

React cung cấp hệ thống lifecycle methods và hooks phong phú, cho phép developers kiểm soát chính xác hành vi của components trong từng giai đoạn:

React Hooks đã cách mạng hóa cách viết React components, cho phép sử dụng state và các tính năng React khác mà không cần viết class components.

1. **Tailwind CSS và DaisyUI ( frameword hỗ trợ css mạnh mẽ)**
   1. **Tailwind CSS - Framework Utility-First**

Tailwind CSS đại diện cho một cách tiếp cận mới trong việc styling web applications. Khác với các framework CSS truyền thống, Tailwind áp dụng phương pháp utility-first, cung cấp các class có mục đích đơn lẻ, có thể kết hợp để tạo ra bất kỳ thiết kế nào.

**Ưu điểm của Tailwind CSS:**

* Tính linh hoạt cao: Cho phép tạo ra bất kỳ thiết kế nào mà không cần viết CSS tùy chỉnh.
* Hiệu suất tối ưu: Hệ thống purge CSS loại bỏ các styles không sử dụng.
* Responsive Design: Hệ thống breakpoints linh hoạt cho phép dễ dàng xây dựng giao diện responsive.
* Customization: Khả năng tùy chỉnh theme, colors, spacing và các giá trị khác.
  1. **DaisyUI - Component Library cho Tailwind**

DaisyUI mở rộng khả năng của Tailwind CSS bằng cách cung cấp một bộ components được xây dựng sẵn. Thư viện này tập trung vào:

* Semantic Component Classes: Cung cấp các class có tính semantic cao như btn, card, modal.
* Theme System: Hệ thống theme phong phú với khả năng tùy chỉnh cao.
* Reduced Complexity: Giảm độ phức tạp của markup bằng cách gom nhóm các utility classes.

1. **Các thư viện khác hỗ trợ React**
   1. **React Hot Toast - Hệ thống thông báo hiện đại**

React Hot Toast là một thư viện chuyên biệt cho việc hiển thị thông báo (notifications/toasts) trong ứng dụng React. Thư viện này nổi bật với thiết kế tối giản nhưng hiệu quả, cùng khả năng tùy biến cao và hiệu suất tốt.

* + 1. **Đặc điểm và ưu điểm**

React Hot Toast mang đến nhiều ưu điểm đáng chú ý trong việc quản lý thông báo:

* **Tính linh hoạt trong hiển thị:**
* Hỗ trợ nhiều kiểu thông báo khác nhau như success, error, loading, và custom notifications.
* Khả năng hiển thị nhiều thông báo cùng lúc với hệ thống queue thông minh.
* Animations mượt mà và tự nhiên, tăng trải nghiệm người dùng.
* **Hiệu suất tối ưu:**
* Kích thước bundle nhỏ gọn, không làm nặng ứng dụng.
* Tự động loại bỏ thông báo sau một khoảng thời gian.
  1. **Axios - Thư viện HTTP Client**

Axios là một HTTP client dựa trên Promise được sử dụng rộng rãi trong cộng đồng JavaScript/React. Thư viện này đóng vai trò quan trọng trong việc thực hiện các request HTTP từ frontend đến backend.

* + 1. **Tính năng chính**
* **Xử lý Request/Response:**
  + Hỗ trợ đầy đủ các HTTP methods (GET, POST, PUT, DELETE, etc.).
  + Tự động transform dữ liệu JSON.
  + Khả năng xử lý response và error một cách nhất quán.
  + Hỗ trợ Promise API, giúp code gọn gàng và dễ đọc.
* **Bảo mật và cấu hình:**
  + Hỗ trợ XSRF protection.
  + Khả năng cấu hình interceptors cho requests và responses.
  + Tích hợp dễ dàng với các hệ thống authentication.
  + Khả năng set default headers và base URL.
    1. **Ưu điểm trong phát triển**
* **Quản lý API hiệu quả:**
  + Tạo instance riêng cho các nhóm API khác nhau.
  + Cấu hình global cho toàn bộ requests.
  + Hỗ trợ cancel requests khi cần thiết.
* **Xử lý lỗi thống nhất:**
  + Cung cấp thông tin chi tiết về lỗi.
  + Khả năng xử lý lỗi tập trung thông qua interceptors.
  + Hỗ trợ custom error handling cho từng loại request.
    1. **Tích hợp với React**
* **Patterns và Best Practices:**
  + Tích hợp hiệu quả với React Hooks.
  + Quản lý state loading và error trong components.
  + Caching và optimizing requests.
* **Quản lý State và Side Effects:**
  + Sử dụng với useEffect để fetch data.
  + Tích hợp với state management libraries.

**II. CÔNG NGHỆ BACKEND VÀ CƠ SỞ DỮ LIỆU**

1. **Node.js và Express - Nền tảng Backend mạnh mẽ**
   1. **Node.js - Runtime Environment**

Node.js đã tạo nên một cuộc cách mạng trong phát triển ứng dụng web bằng cách mang JavaScript lên môi trường server-side. Đây là một platform được xây dựng trên V8 JavaScript Engine của Chrome, cho phép thực thi mã JavaScript bên ngoài trình duyệt.

* 1. **Express Framework**

Express là web framework phổ biến nhất cho Node.js, cung cấp một loạt tính năng để xây dựng các ứng dụng web và API.

Express cung cấp hệ thống routing mạnh mẽ cho phép:

* URL Mapping: Định tuyến các request HTTP đến các handler functions tương ứng.
* Middleware Chain: Xử lý request thông qua chuỗi các middleware functions.
* Route Parameters: Hỗ trợ các tham số động trong URL.

1. **MongoDB - Hệ quản trị cơ sở dữ liệu NoSQL**
   1. **Kiến trúc Document-Oriented**

MongoDB lưu trữ dữ liệu trong các documents dạng BSON (Binary JSON), mang lại tính linh hoạt cao trong cấu trúc dữ liệu:

* Schema-less: Không yêu cầu schema cố định, cho phép thay đổi cấu trúc dữ liệu dễ dàng.
* Nested Documents: Hỗ trợ lưu trữ các cấu trúc dữ liệu phức tạp với documents lồng nhau.
* Arrays: Cho phép lưu trữ arrays trong documents, phù hợp với dữ liệu có quan hệ một-nhiều.
  1. **Mongoose ODM (Object Document Mapping)**

Mongoose là thư viện ODM (Object Document Mapping) cho MongoDB và Node.js, cung cấp:

* **Schema Definition và Validation**
  + Type Checking: Kiểm tra kiểu dữ liệu tự động.
  + Required Fields: Xác định các trường bắt buộc.
  + Custom Validators: Tạo các quy tắc validation tùy chỉnh.
* **Middleware và Hooks**

Mongoose cung cấp các hooks cho phép:

* Pre/Post Hooks: Thực thi code trước/sau các operations như save, update, remove.
* Query Middleware: Tùy chỉnh và mở rộng các queries.

# PHẦN III. CÀI ĐẶT THỰC NGHIỆM VÀ KẾT QUẢ

1. **Kiến trúc dự án**

* Frontend được xây dựng bằng React và sử dụng Vite làm công cụ build

frontend/

├── node\_modules/ # Chứa các thư viện frontend được cài đặt

├── public/ # Chứa các file tĩnh

├── src/ # Chứa mã nguồn chính của dự án

│ ├── admin/ # Chứa các components UI và pages phần Admin

│ │ ├── components/ # Các component UI Admin có thể tái sử dụng

│ │ │ ├── AdminFooter.jsx

│ │ │ ├── AdminNavbar.jsx

│ │ │ └── AdminSidebar.jsx

│ │ ├── pages/ # Các component các trang Admin

│ │ │ ├──CourseDetail.jsx

│ │ │ ├── ExercisesDetail.jsx

│ │ │ ├── LessonDetail.jsx

│ │ │ └── UserDetail.jsx

│ ├── assets/ # Chứa các hình ảnh, logo của website

│ ├── components/ # Các component UI Client có thể tái sử dụng

│ │ ├── ChatBot.jsx

│ │ ├── Comments.jsx

│ │ ├── FeedBack.jsx

│ │ ├── Footer.jsx

│ │ ├── LeaderboardCard.jsx

│ │ ├── Navbar.jsx

│ │ └── WacthCard.jsx

│ ├── hooks/ # Chứa các custom hooks dùng để quản lý logic gọi API, xử lý các state phức tạp

│ │ ├── useCU\_Progress.jsx

│ │ ├── useCUD\_CoureseData.jsx

│ │ ├── useCUD\_ExerciseData.jsx

│ │ ├── useCUD\_LessonsData.jsx

│ │ ├── useGetCourse.jsx

│ │ ├── useGetExercise.jsx

│ │ ├── useGetLesson.jsx

│ │ ├── useGetProgress.jsx

│ │ └── useGetUsers.jsx

│ ├── pages/ # Các component các trang Client

│ │ ├── Course.jsx

│ │ ├── DetailCourse.jsx

│ │ ├── Home.jsx

│ │ ├── Login.jsx

│ │ ├── Profile.jsx

│ │ ├── Rank.jsx

│ │ └── Register.jsx

│ ├── AdminApp.jsx/ # Component trang Admin

│ ├── index.css/ # CSS toàn cục, áp dụng cho toàn ứng dụng

│ ├── App.jsx # Component gốc, đóng vai trò làm root component của ứng dụng

│ └── main.jsx # Điểm khởi đầu ứng dụng, là nơi React được khởi tạo và kết nối vào DOM

├── .env # Chưa biến môi trường GEMNI\_API\_KEY

├── package.json # Quản lý các dependencies và scripts

├── tailwind.config.js # Cấu hình Tailwind CSS tùy chỉnh các theme hoặc plugin

└── vite.config.js # Cấu hình Vite

* Backend được xây dựng bằng Node.js/Express

backend/

├── controllers/ # Chứa các hàm xử lý logic nghiệp vụ

│ ├── comment.controllers.js

│ ├── courese.controllers.js

│ ├── exercises.controllers.js

│ ├── lesson.controllers.js

│ ├── progress.controllers.js

│ └── user.controllers.js

├── models/ # Chứa các định nghia mô hình dữ liệu

│ ├── comment.models.js

│ ├── courese.models.js

│ ├── exercises.models.js

│ ├── lesson.models.js

│ ├── progress.models.js

│ └── user.models.js

├── routes/ # Định tuyến API, ánh xạ các URL tới các controller tương ứng

│ ├── comment.router.js

│ ├── courese.router.js

│ ├── exercises.router.js

│ ├── lesson.router.js

│ ├── progress.router.js

│ └── user.router.js

├── db/ # Cấu hình database kết nối tới MongoDB

│ └── connect.js

├── utils/ # Chứa các hàm tiện ích

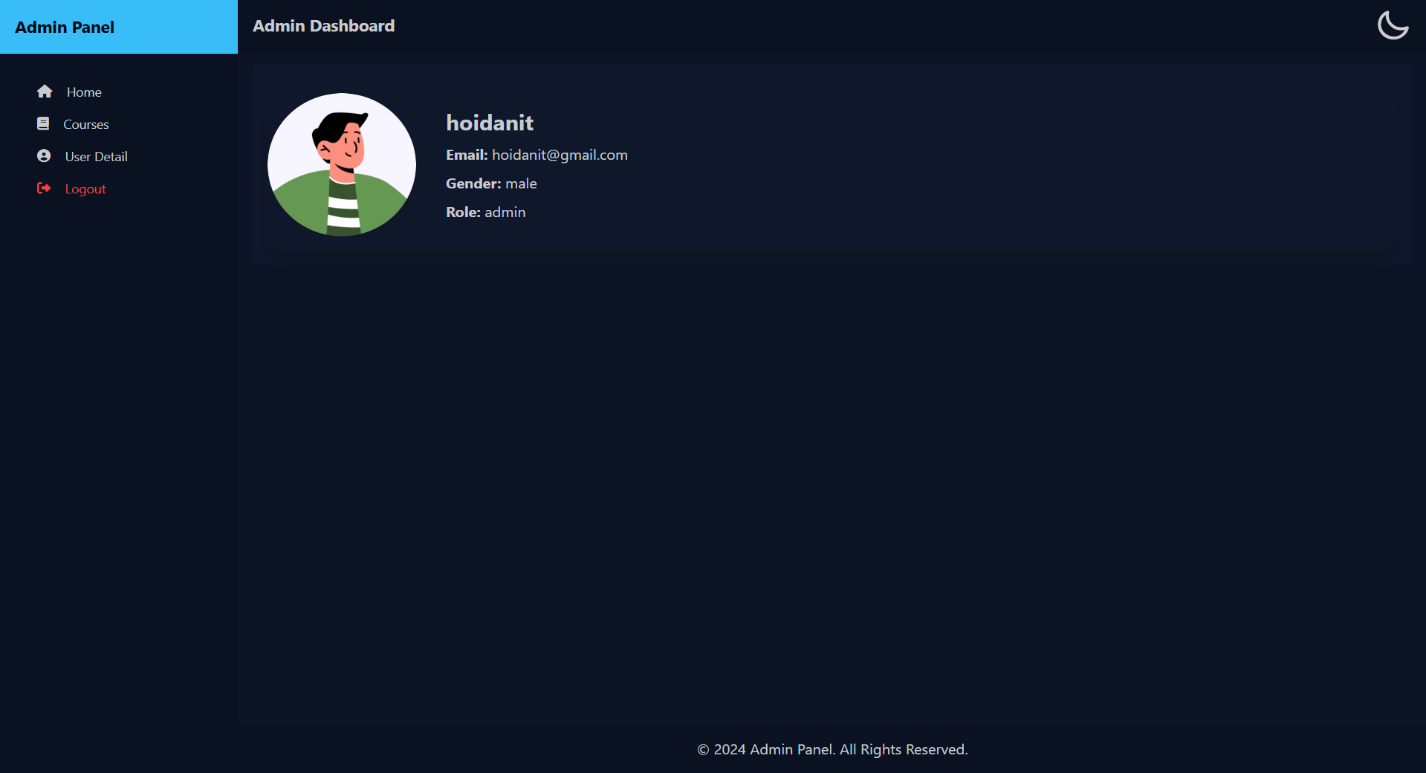
│ └── generateToken.js

└── server.js # File khởi động server

├── node\_modules # Chứa các thư viện backend được cài đặt

├── .env # Chứa các biến môi trường PORT, địa chỉ MongoDB

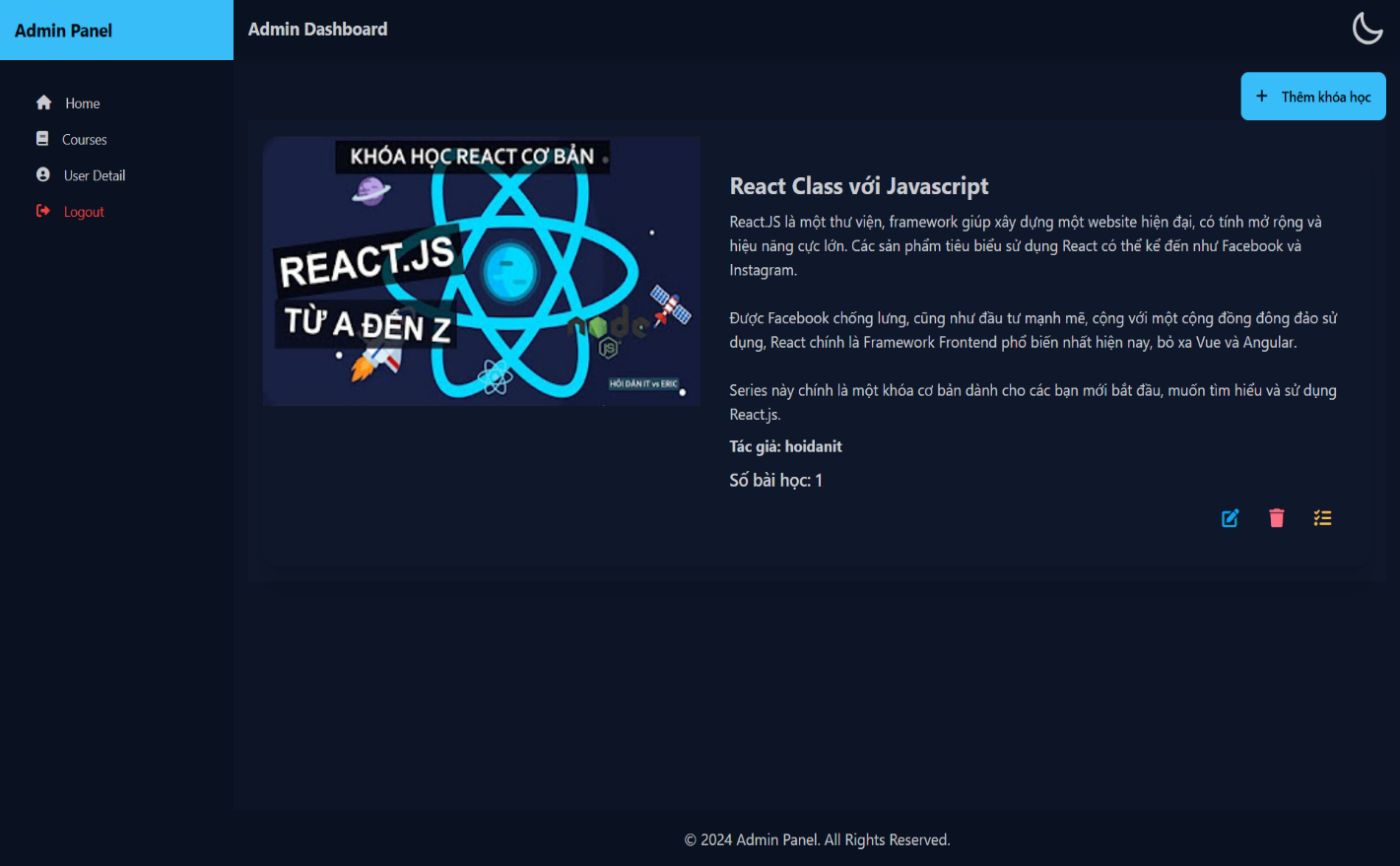
└── package.json # Quản lý các dependencies và scripts

1. **Một số giao diện website**
   1. **Phần quản trị**
      1. **Giao diện tài khoản quản trị**

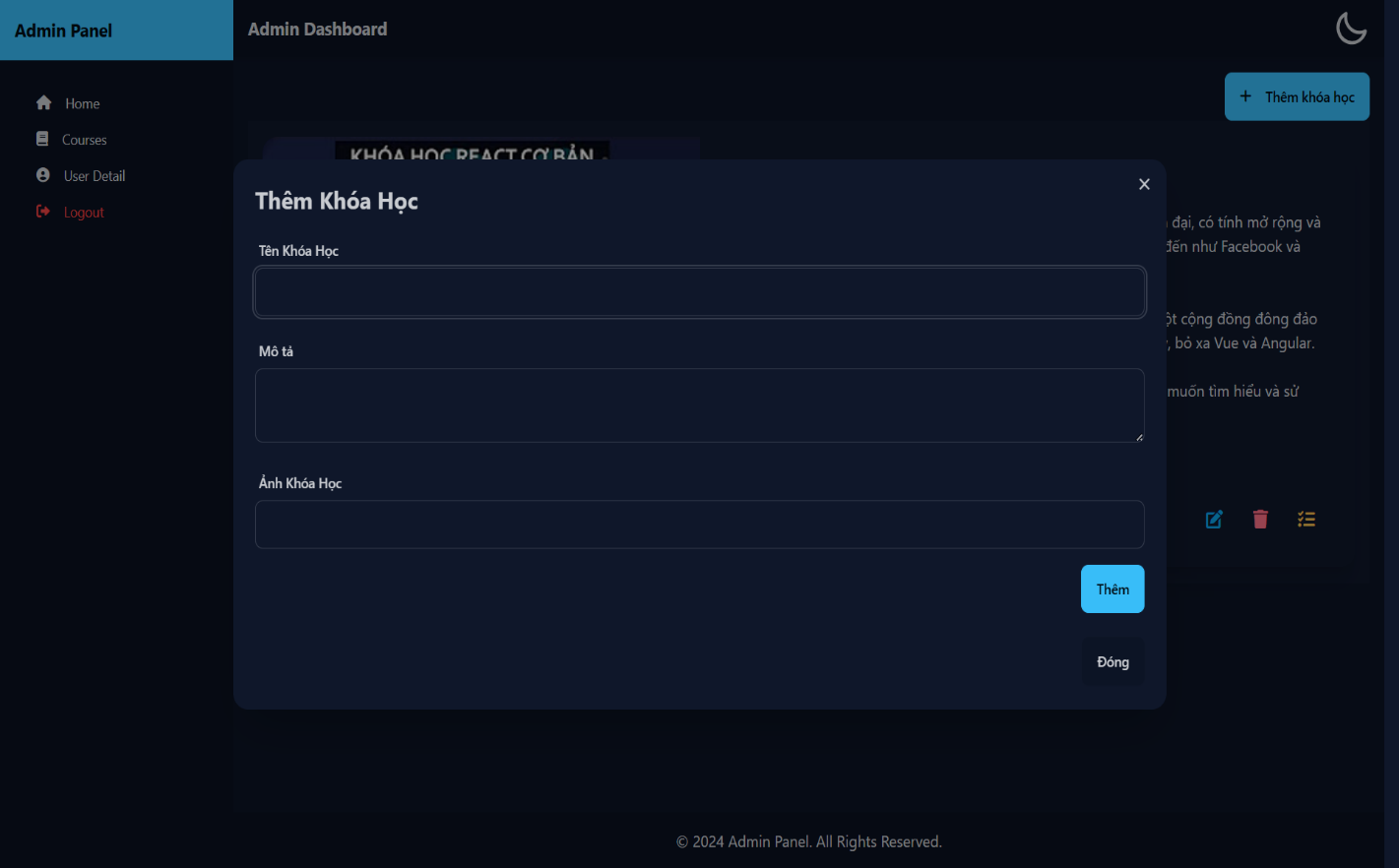
*Hình 1.1 Giao diện tài khoản quản trị.*

* + 1. **Giao diện quản lý danh sách khóa học**

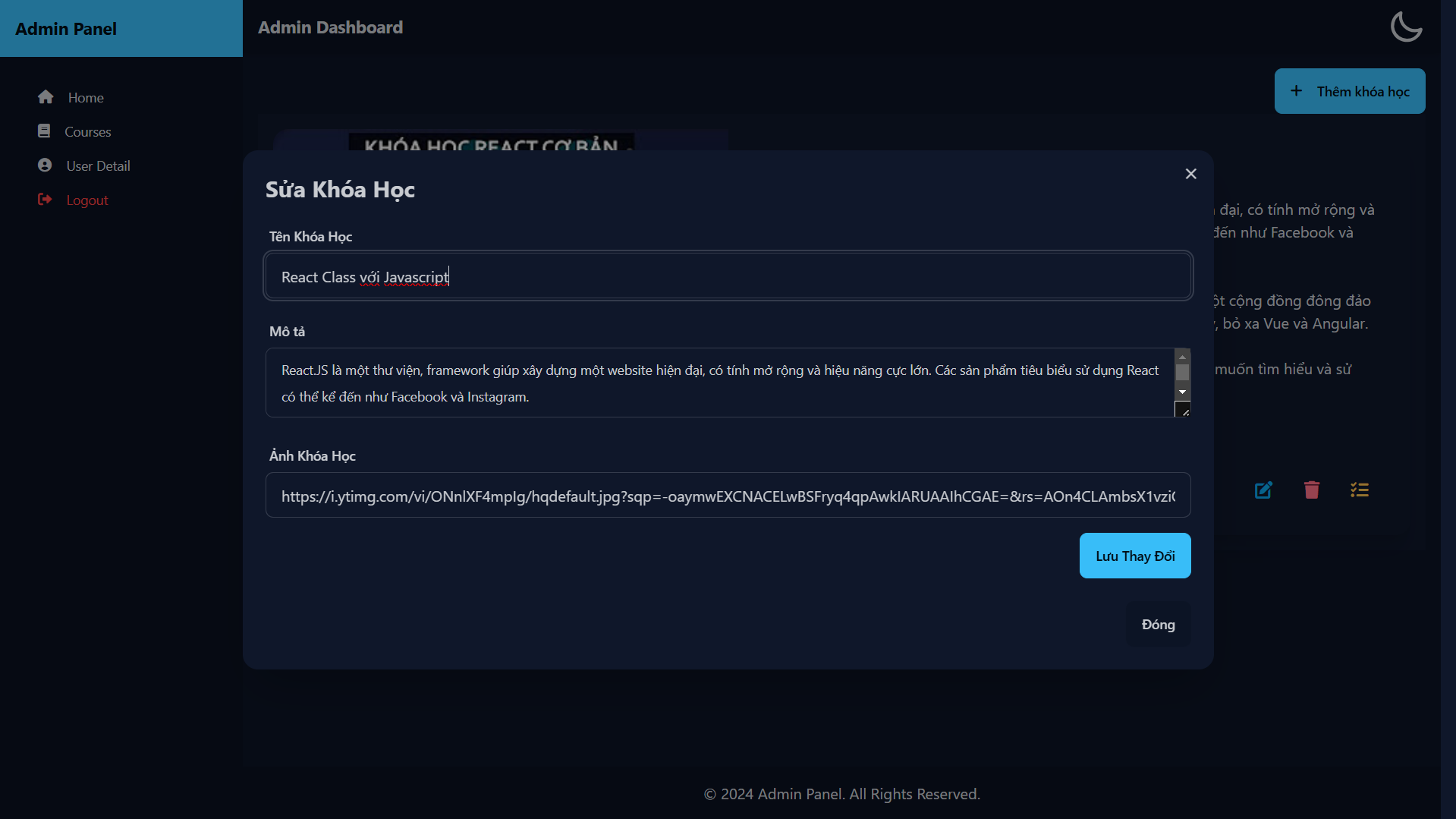
Giao diện quản lý khóa học cung cấp một cái nhìn tổng quan và khả năng thao tác linh hoạt với toàn bộ danh mục khóa học trên nền tảng. Người quản trị có thể dễ dàng theo dõi, chỉnh sửa và cập nhật, xóa thông tin khóa học một cách nhanh chóng và hiệu quả.

* Giao diện quản lý danh sách khóa học

*Hình 1.2 Giao diện quản lý danh sách khóa học.*

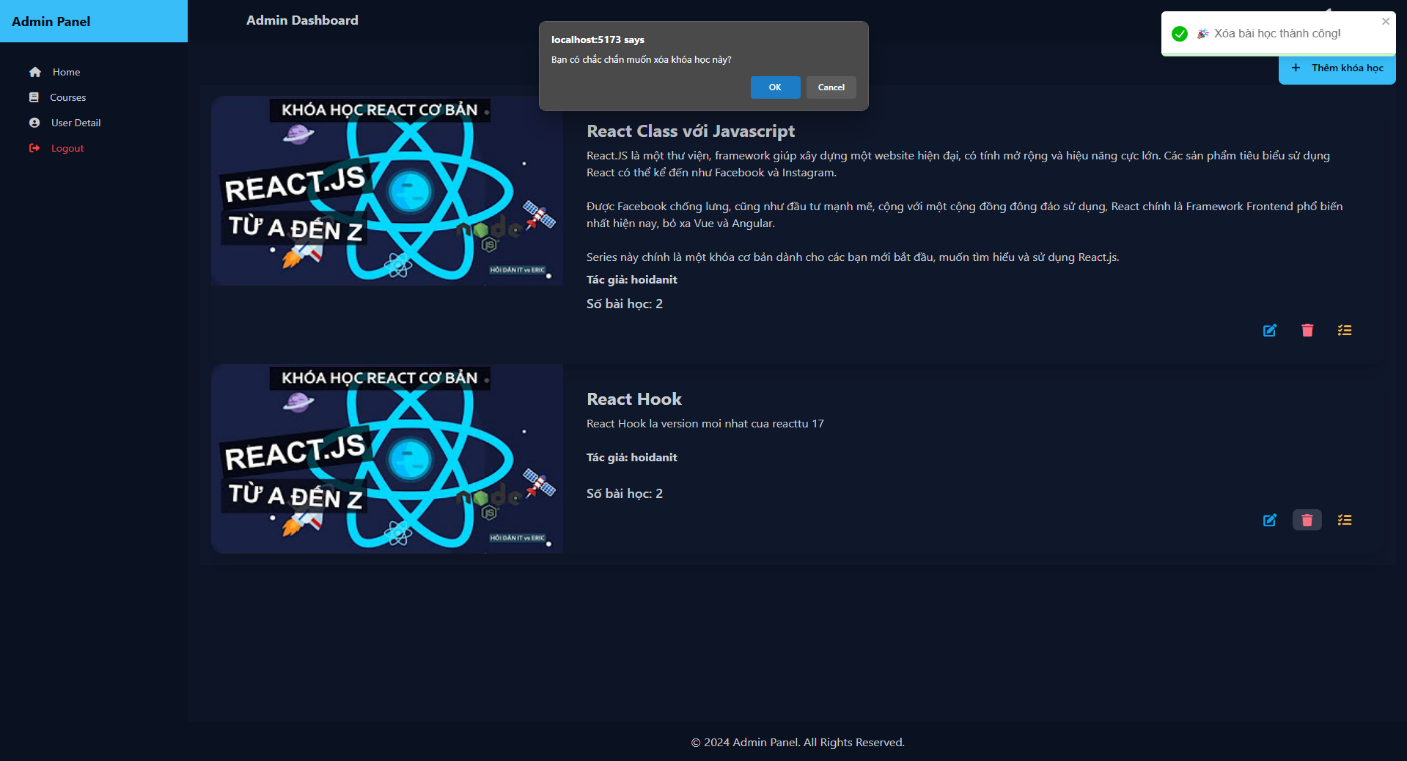
* ****Giao diện thêm khóa học

*Hình 1.3 Giao diện thêm khóa học.*

* ****Giao diện chỉnh sửa thông tin khóa học

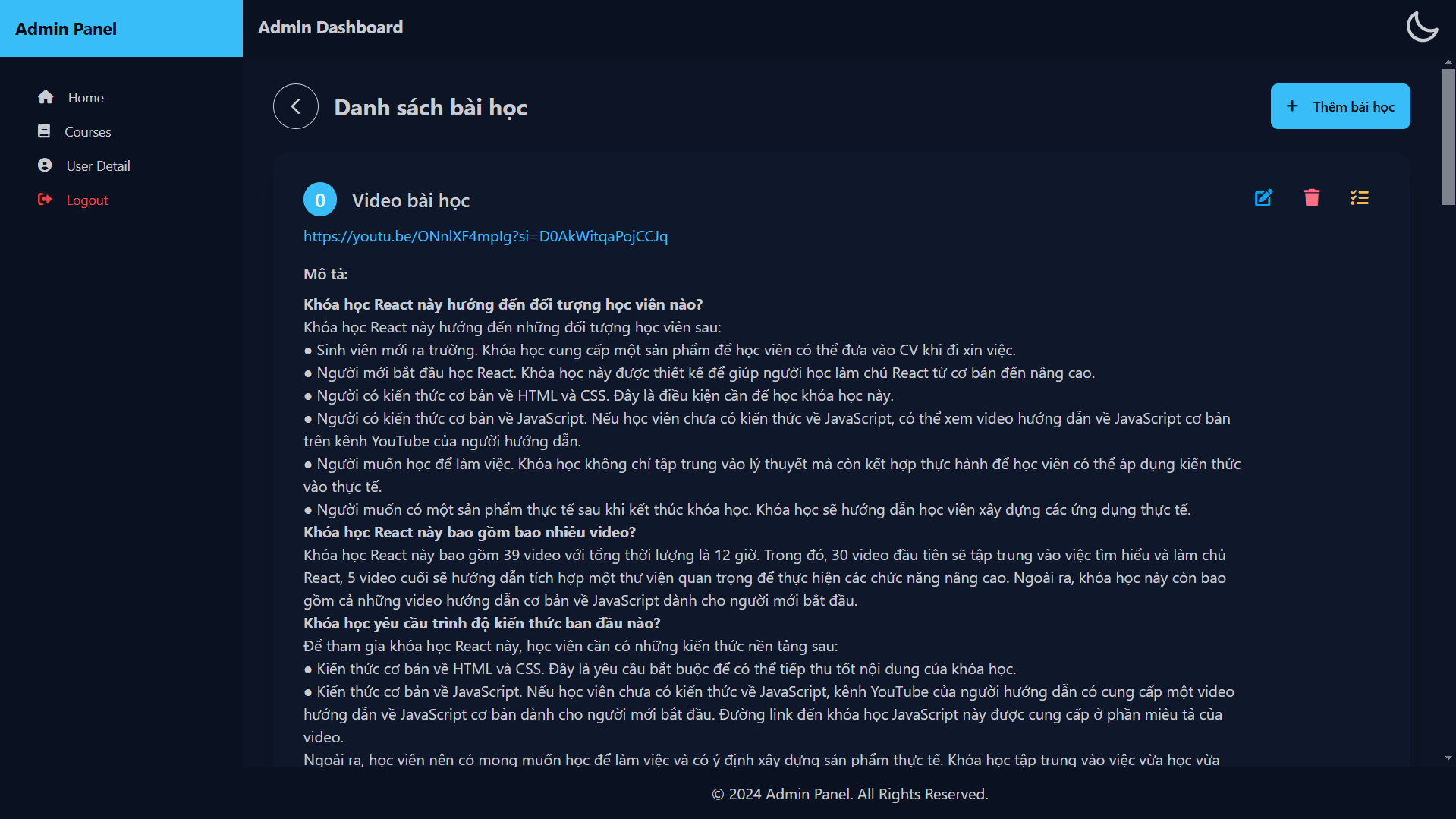
*Hình 1.4 Giao diện chỉnh sửa thông tin khóa học.*

* Giao diện thông báo xóa thông tin khóa học

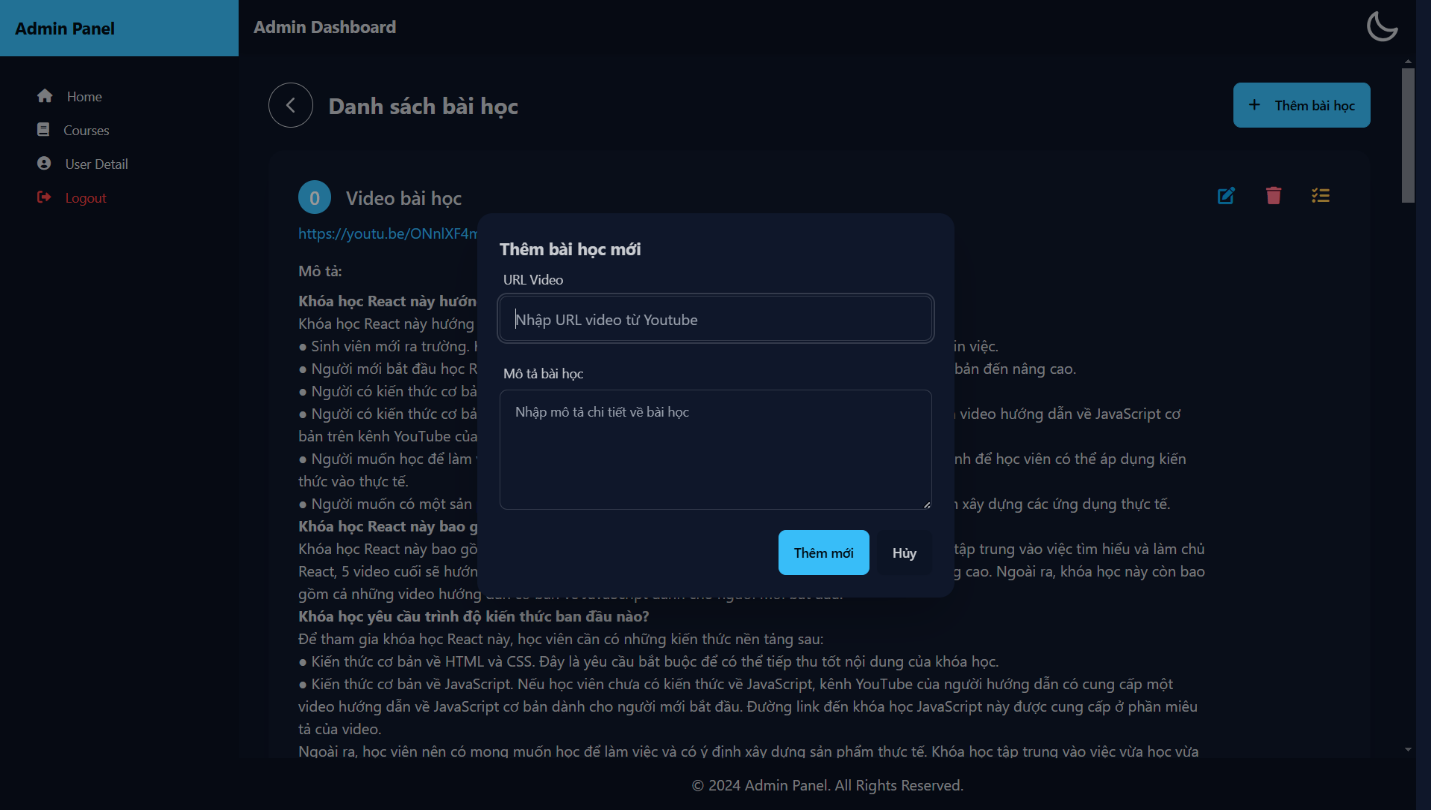
*Hình 1.5 Giao diện thông báo xóa thông tin khóa học.*

* + 1. **Giao diện quản lý danh sách bài học**

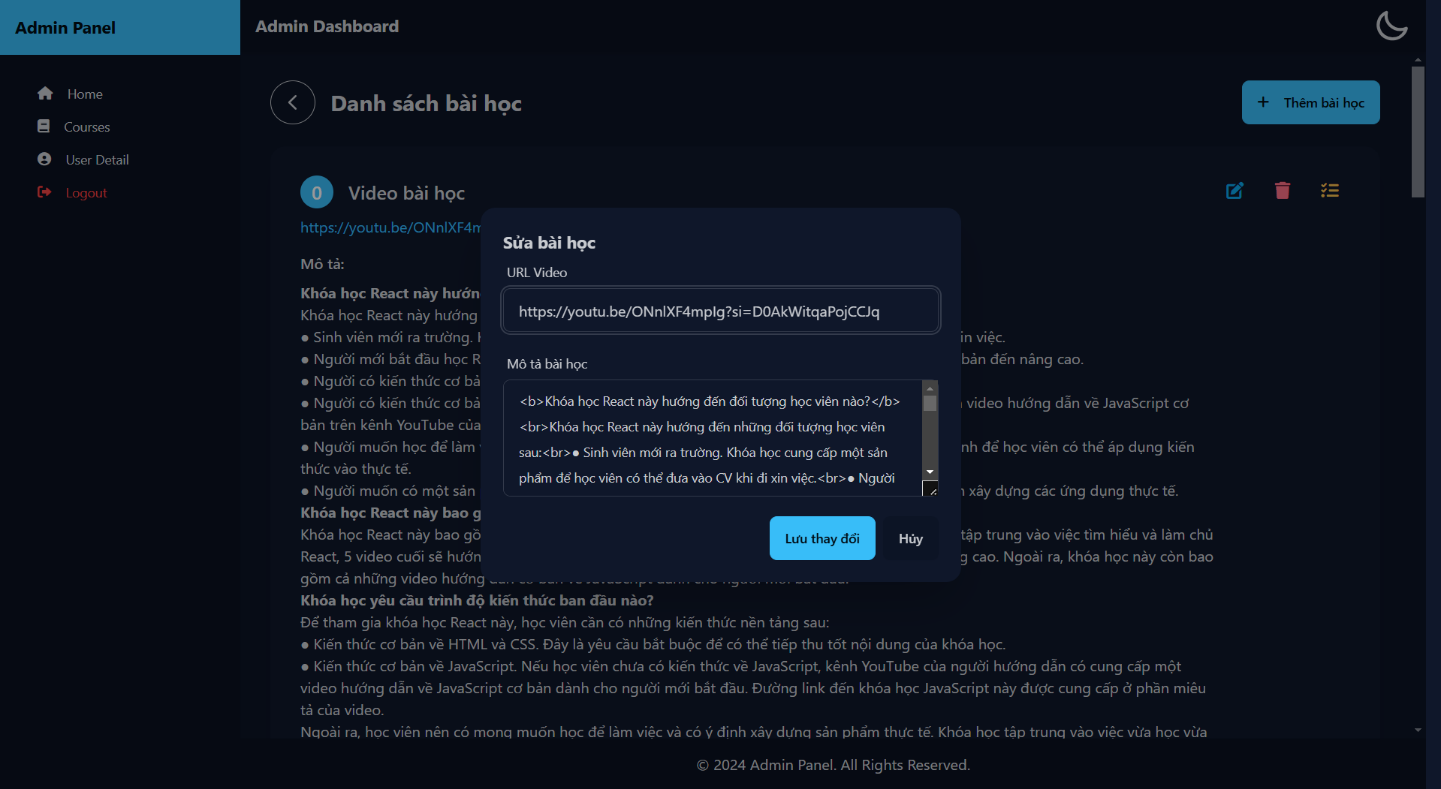
Giao diện quản lý bài học cung cấp một cái nhìn tổng quan về tất cả bài học thuộc một khóa học cụ thể. Người quản trị có thể dễ dàng nắm bắt cấu trúc khóa học, xem trước nội dung từng bài học và thực hiện các chỉnh sửa, cập nhật và xóa thông tin một cách thuận tiện.

* Giao diện danh sách các bài học

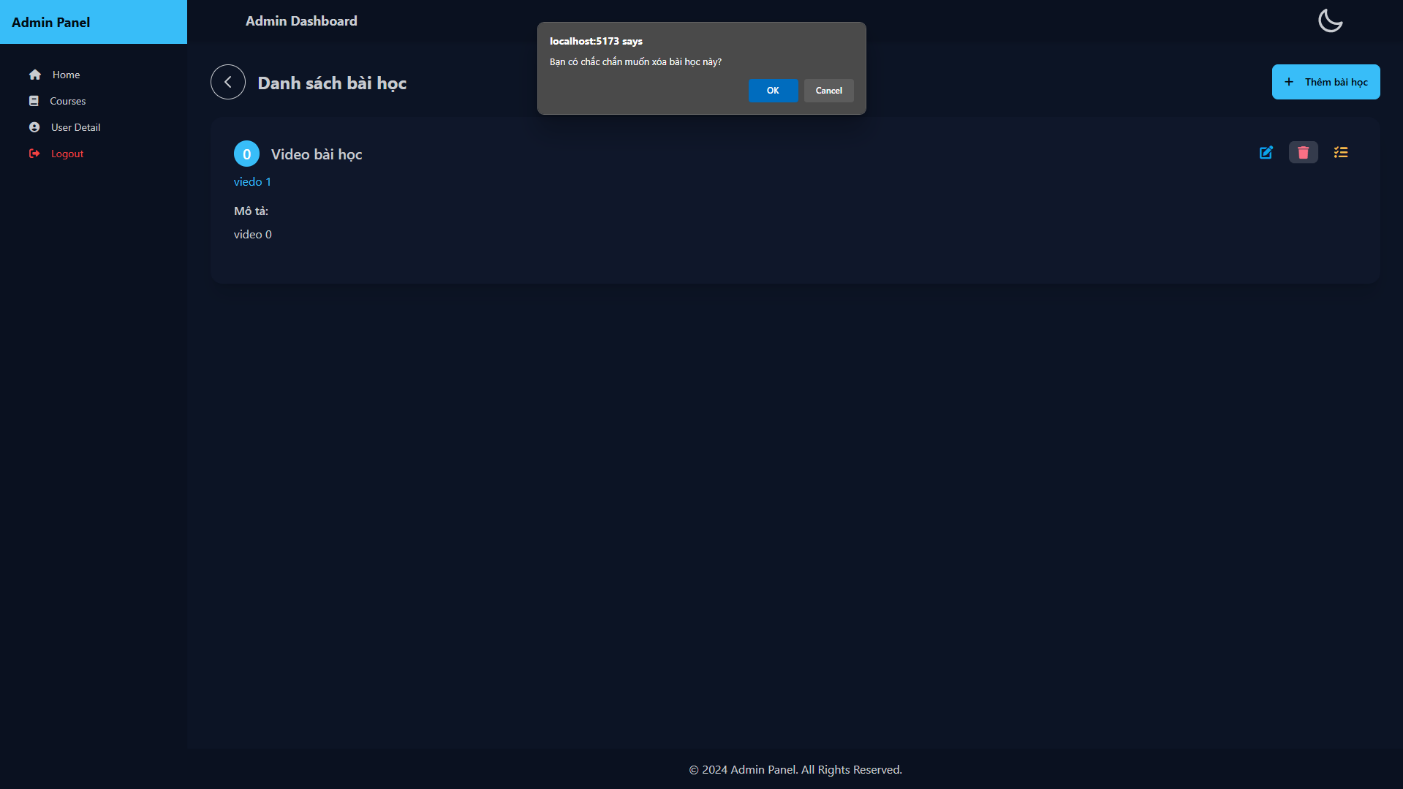
*Hình 1.6 Giao diện danh sách các bài học.*

* Giao diện thêm bài học

*Hình 1.7 Giao diện thêm bài học.*

* Giao diện chỉnh sửa bài học

##### *Hình 1.8 Giao diện chỉnh sửa bài học.*

* Giao diện thông báo xóa thông tin bài học

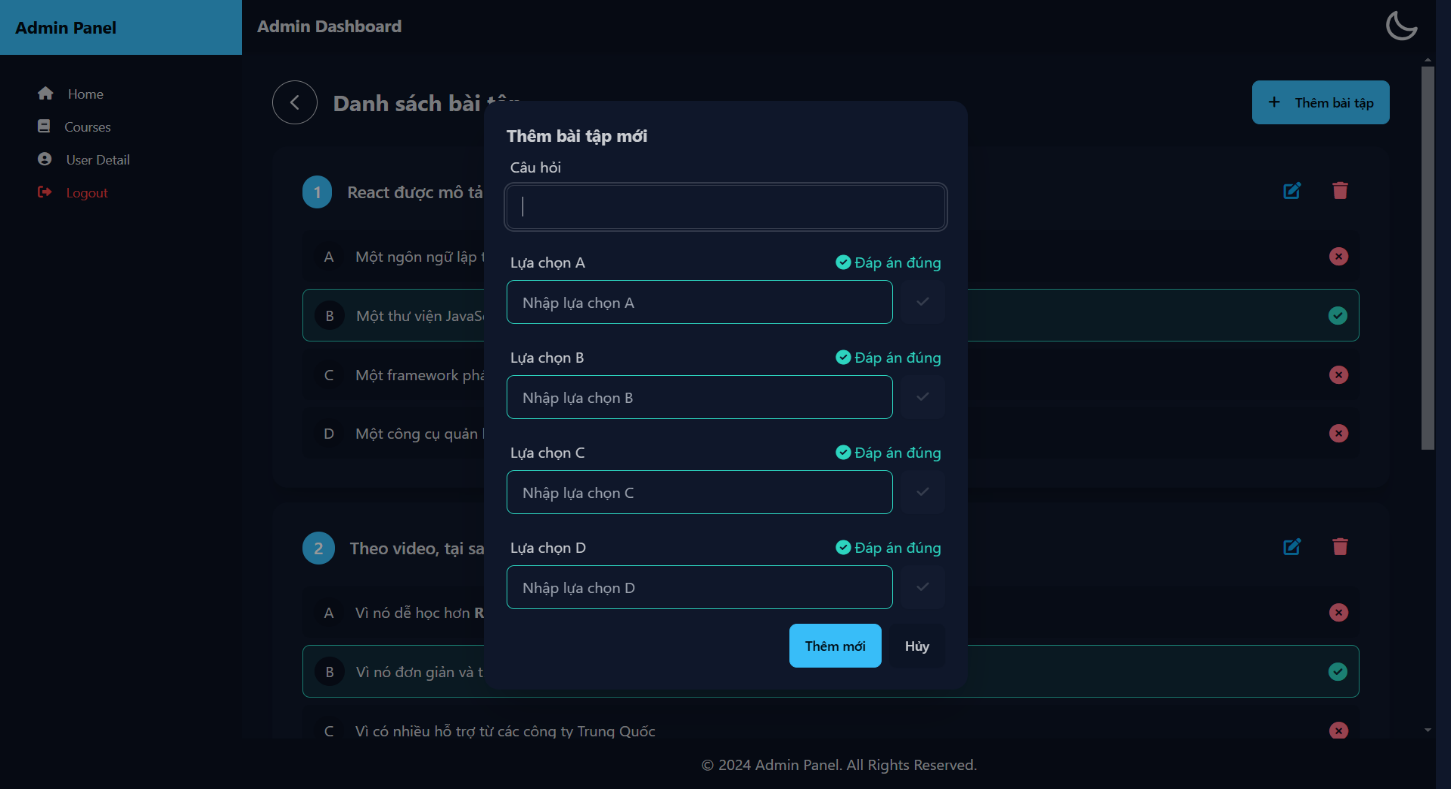
##### *Hình 1.9 Giao diện thông báo xóa thông tin bài học.*

* + 1. **Giao diện quản lý các bài tập**

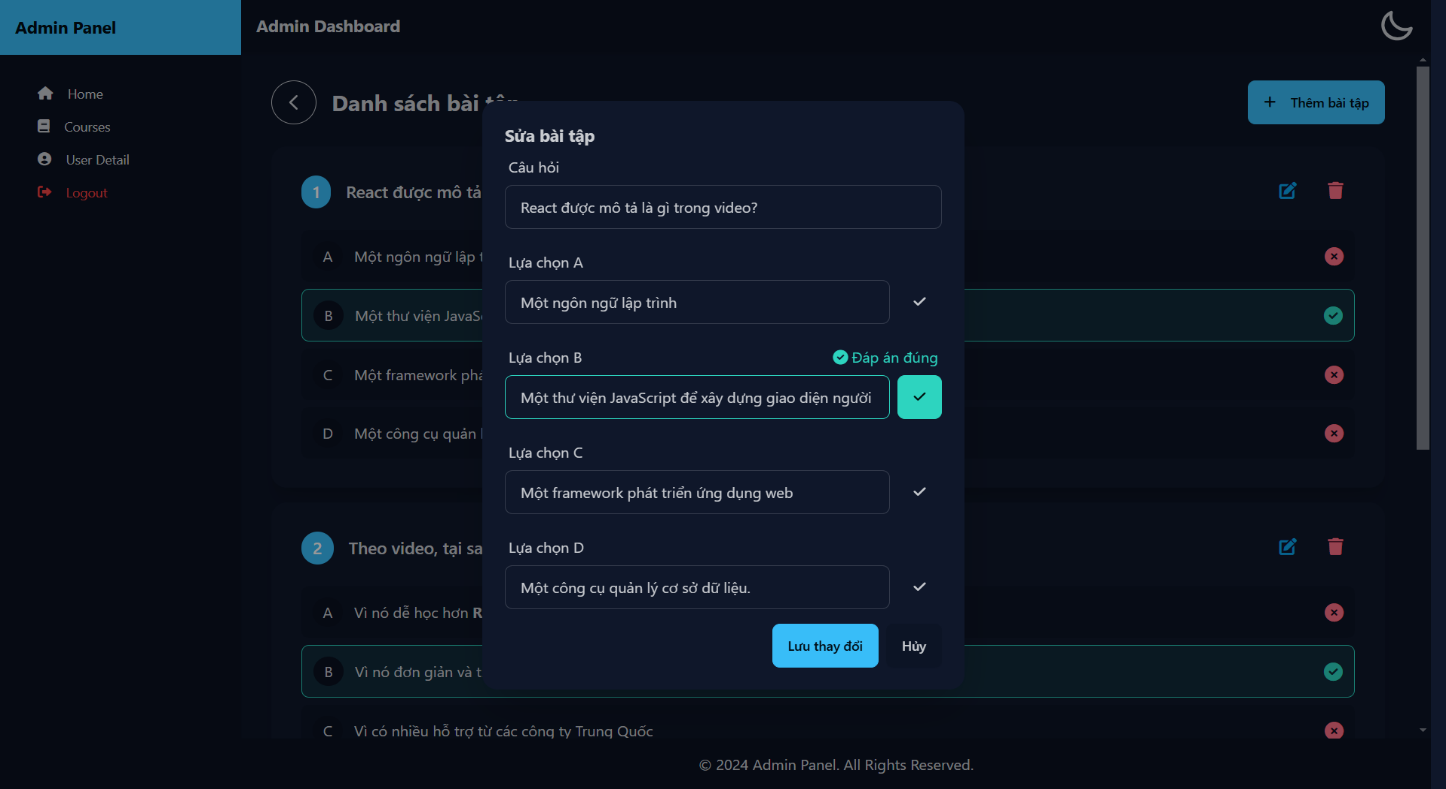
Giao diện này cho phép người quản trị dễ dàng bổ sung và chỉnh sửa, xóa các bài tập tương ứng với từng bài học trong khóa học. Người quản trị cũng có thể theo dõi số lượng câu hỏi trong mỗi bài tập để đảm bảo nội dung bài học đầy đủ và phù hợp.

* Giao diện danh sách bài tập

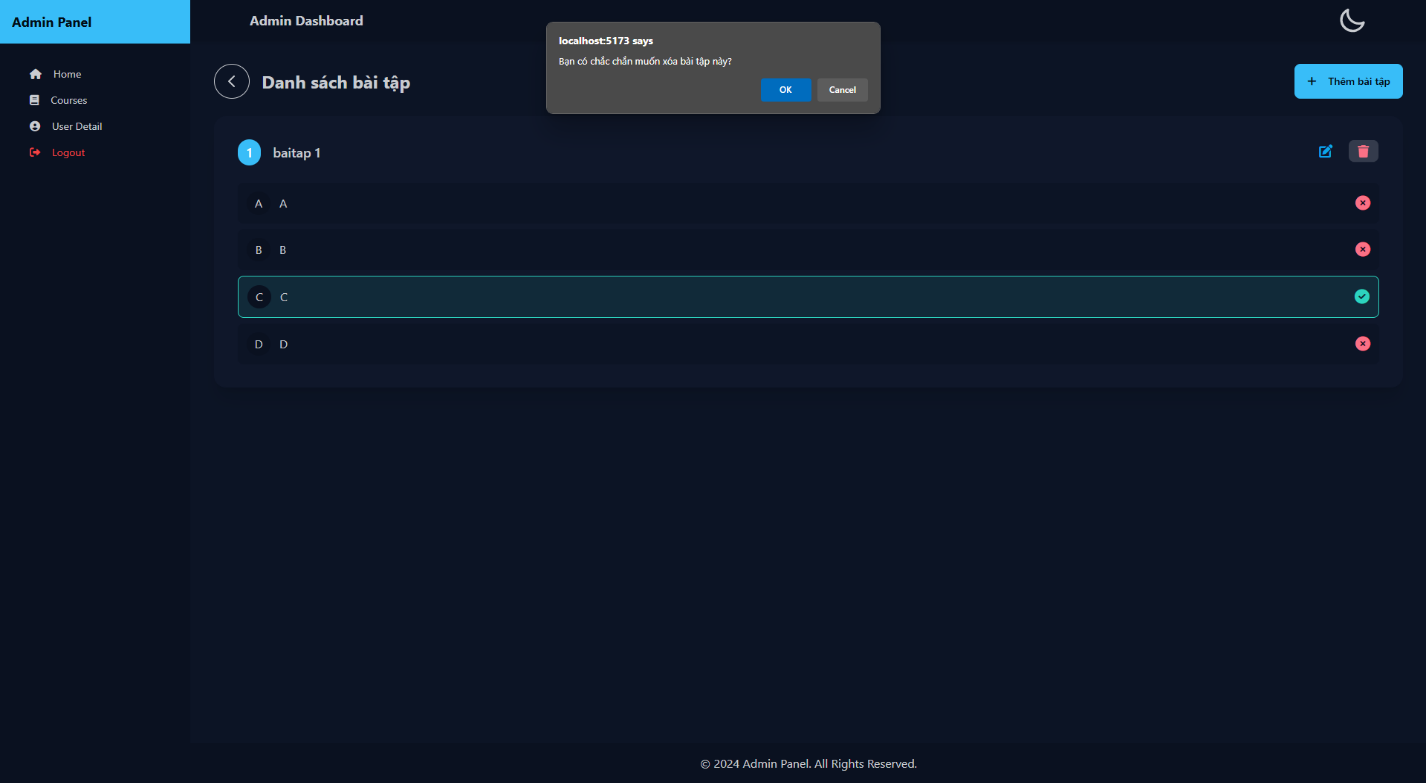
*Hình 1.10 Giao diện danh sách bài tập.*

* Giao diện thêm bài tập

##### *Hình 1.11 Giao diện thêm bài tập.*

* ****Giao diện chỉnh sửa bài tập

*Hình 1.12 Giao diện chỉnh sửa bài tập.*

* Giao diện thông báo xóa bài tập

##### *Hình 1.13 Giao diện thông báo xóa bài tập.*

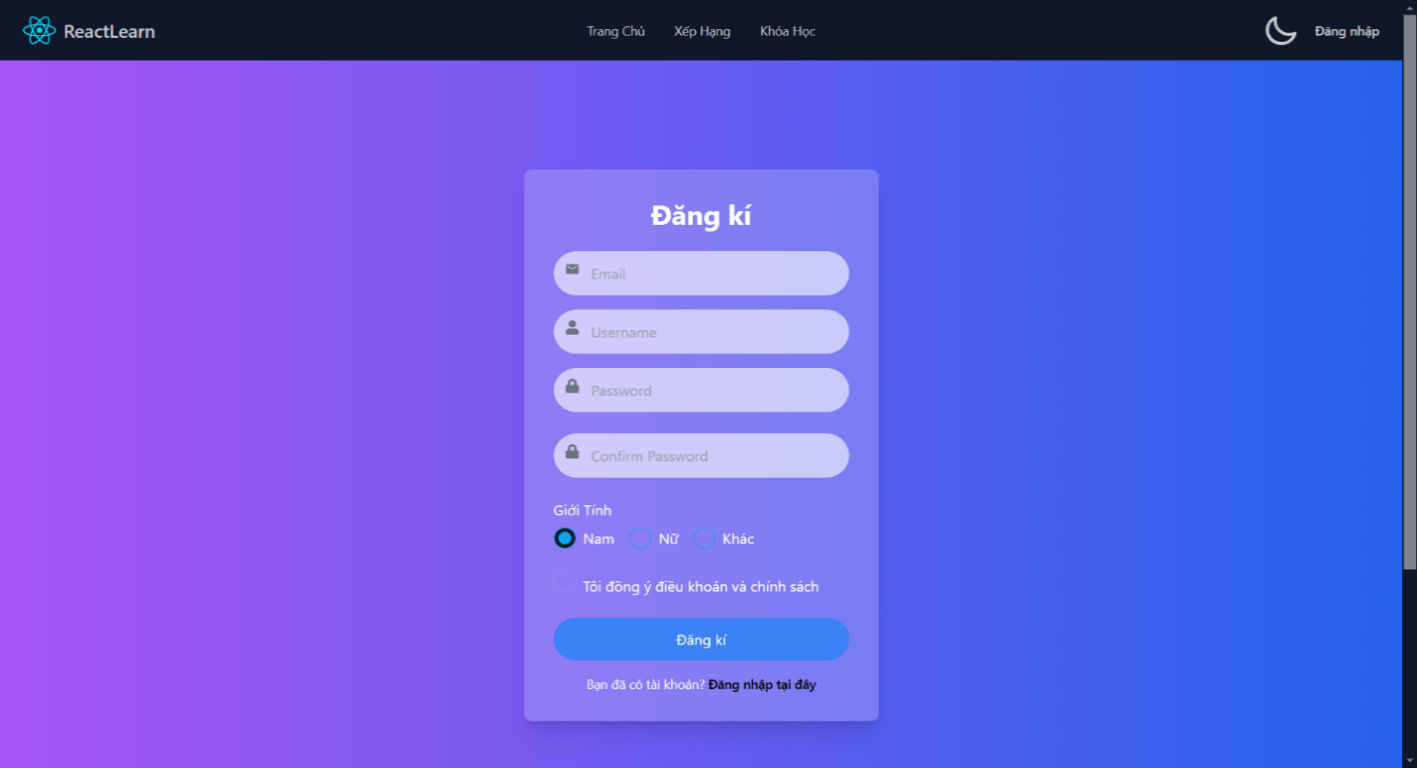
* 1. **Phần người dùng**
     1. **Giao diện người dùng ở thiết bị màn hình lớn**

Giao diện hỗ trợ cả chế độ sáng và tối, cho phép người dùng tùy chỉnh trải nghiệm hiển thị theo sở thích bằng cách đơn giản chạm vào biểu tượng mặt trời (cho chế độ sáng) hoặc mặt trăng (cho chế độ tối).

* + - 1. **Giao diện đăng nhập**

*Hình 1.14 Giao diện đăng nhập.*

* + - 1. **Giao diện đăng ký**

Khi người dùng đăng kí thì password được mã hóa với thư viện brcypt được mã hóa với salt đảm bảo nếu có cùng password cũng sẽ được mã hóa băm thành các chuỗi ngẫu nhiên đảm báo quá trình bảo mật cho người dùng. Kết quả cuối cùng của mật khẩu người dùng gồm salt, Work factor và lưu kết quả là kết quả dã hash mật khẩu và đảm bảo mỗi hàm được băm đều khác nhau không nào tương tự nhau.

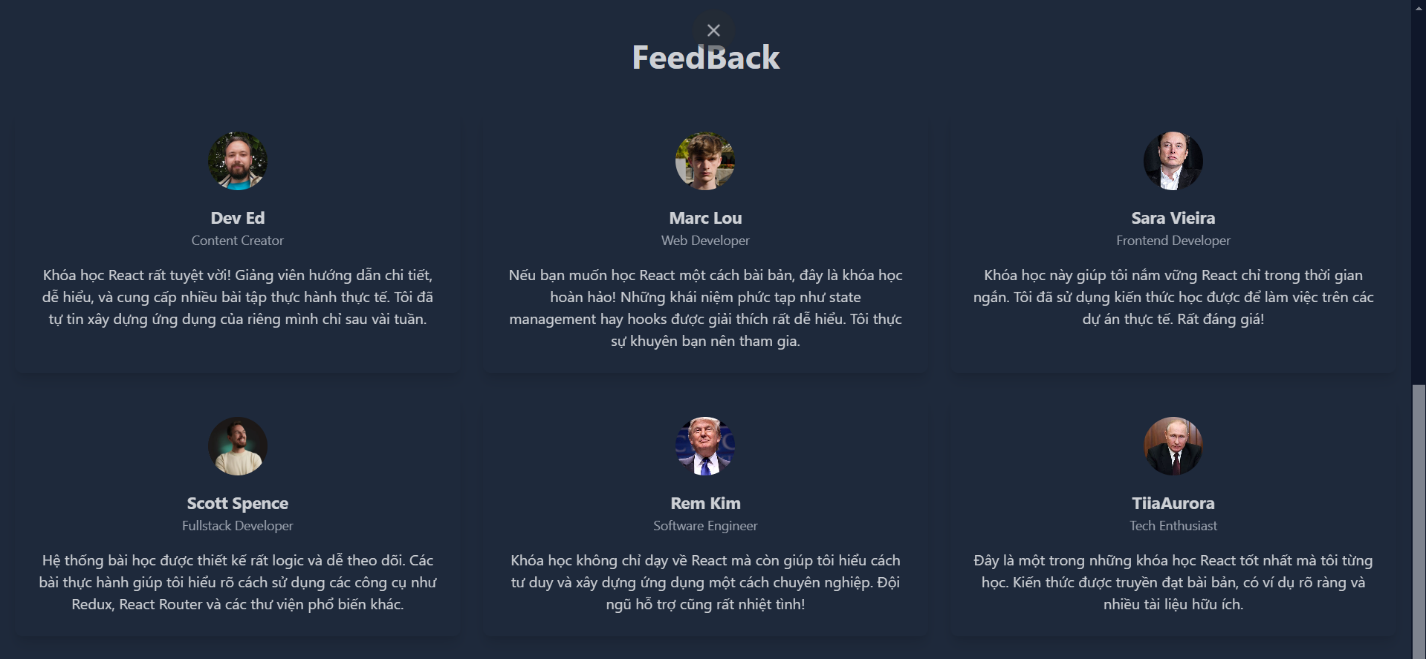
*Hình 1.15 Giao diện đăng ký.*

* + - 1. **Giao diện trang chủ**

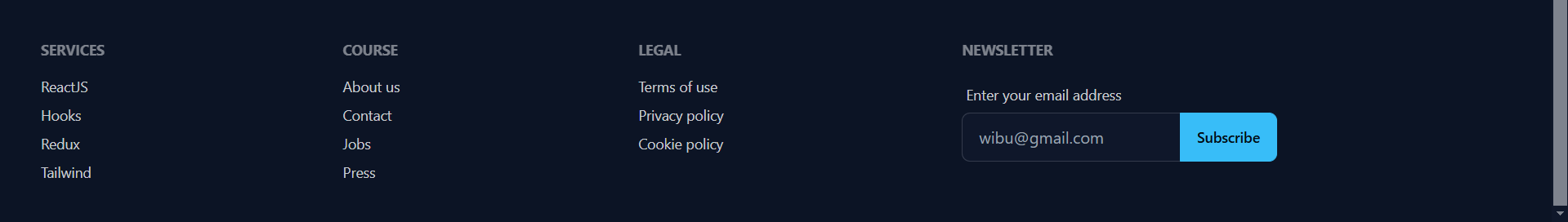
Giao diện này cung cấp góc nhìn toàn cảnh về ReactJS và lộ trình học tập bài bản. Những chia sẻ và đánh giá từ những học viên trước sẽ giúp bạn hiểu rõ hơn về nội dung khóa học cũng như hiệu quả thực tế mà nó mang lại.

* Giao diện phần header

*Hình 1.16 Giao diện phần header.*

* Giao diện phần feedback

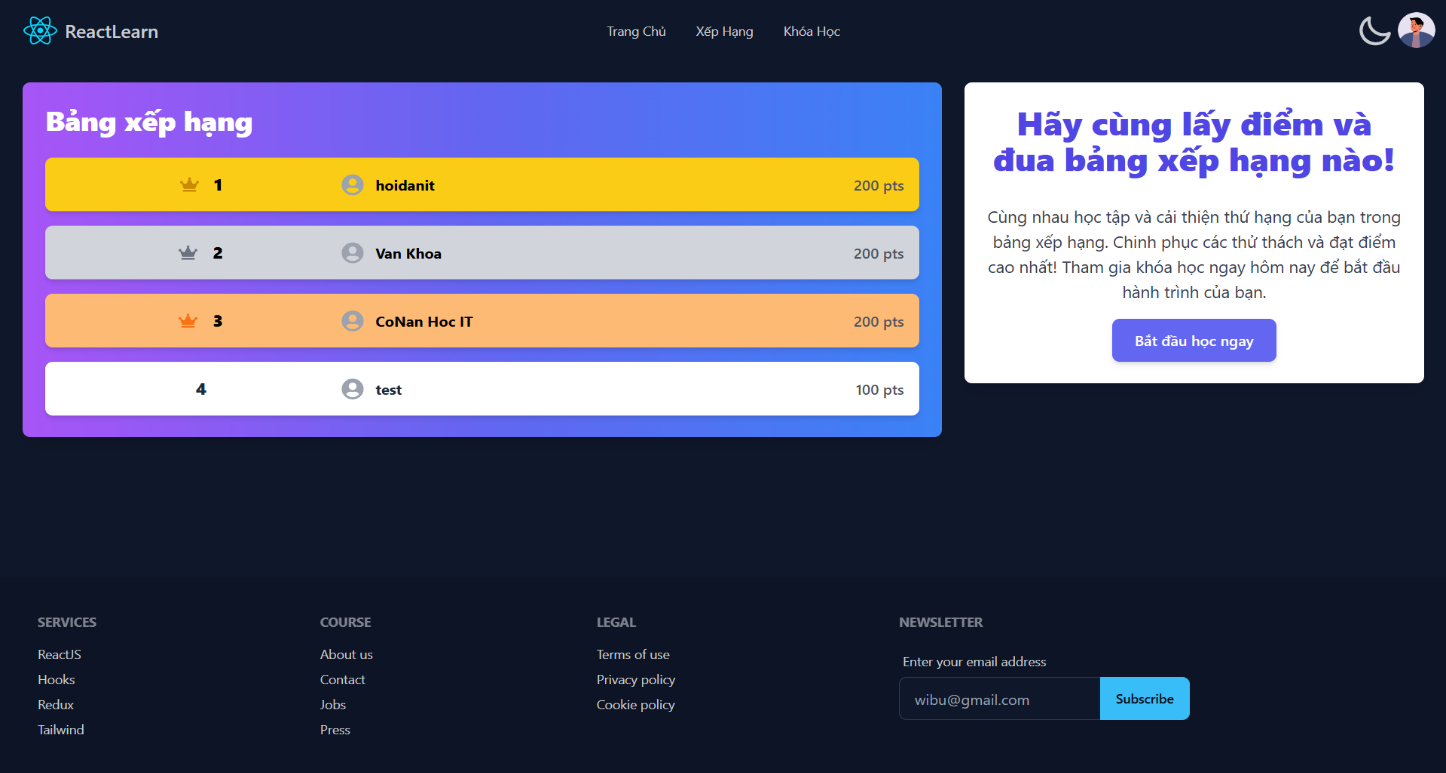
*Hình 1.17 Giao diện phần feedback.*

* Giao diện phần footer

*Hình 1.18 Giao diện phần footer.*

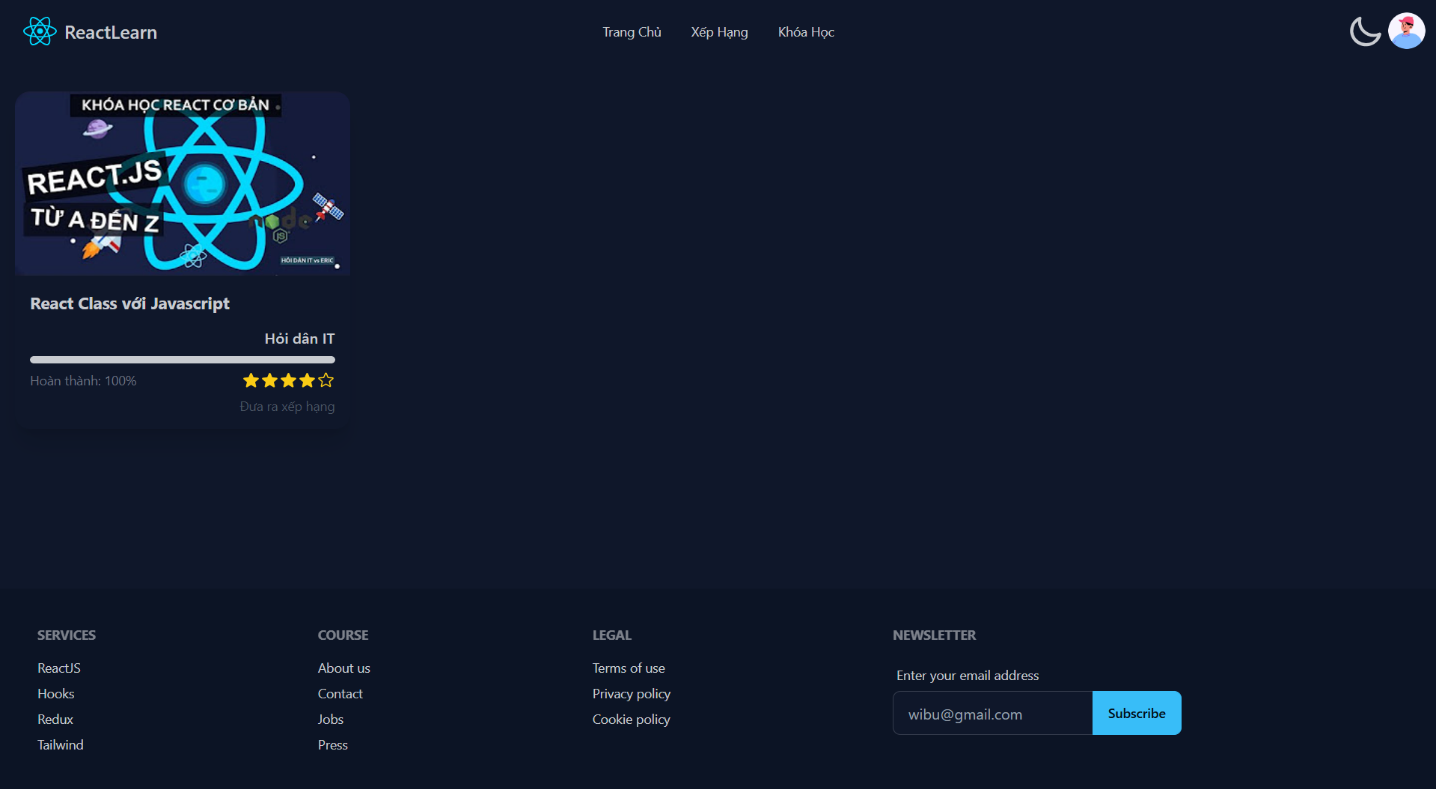
* + - 1. **Giao diện xếp hạng**

Bảng xếp hạng học tập khơi dậy tinh thần cạnh tranh lành mạnh, thúc đẩy người học nỗ lực hoàn thành bài học và đạt điểm PST cao để vươn lên dẫn đầu. Đây là động lực tuyệt vời để các học viên chinh phục kiến thức và khẳng định bản thân.

*****Hình 1.19 Giao diện xếp hạng.*

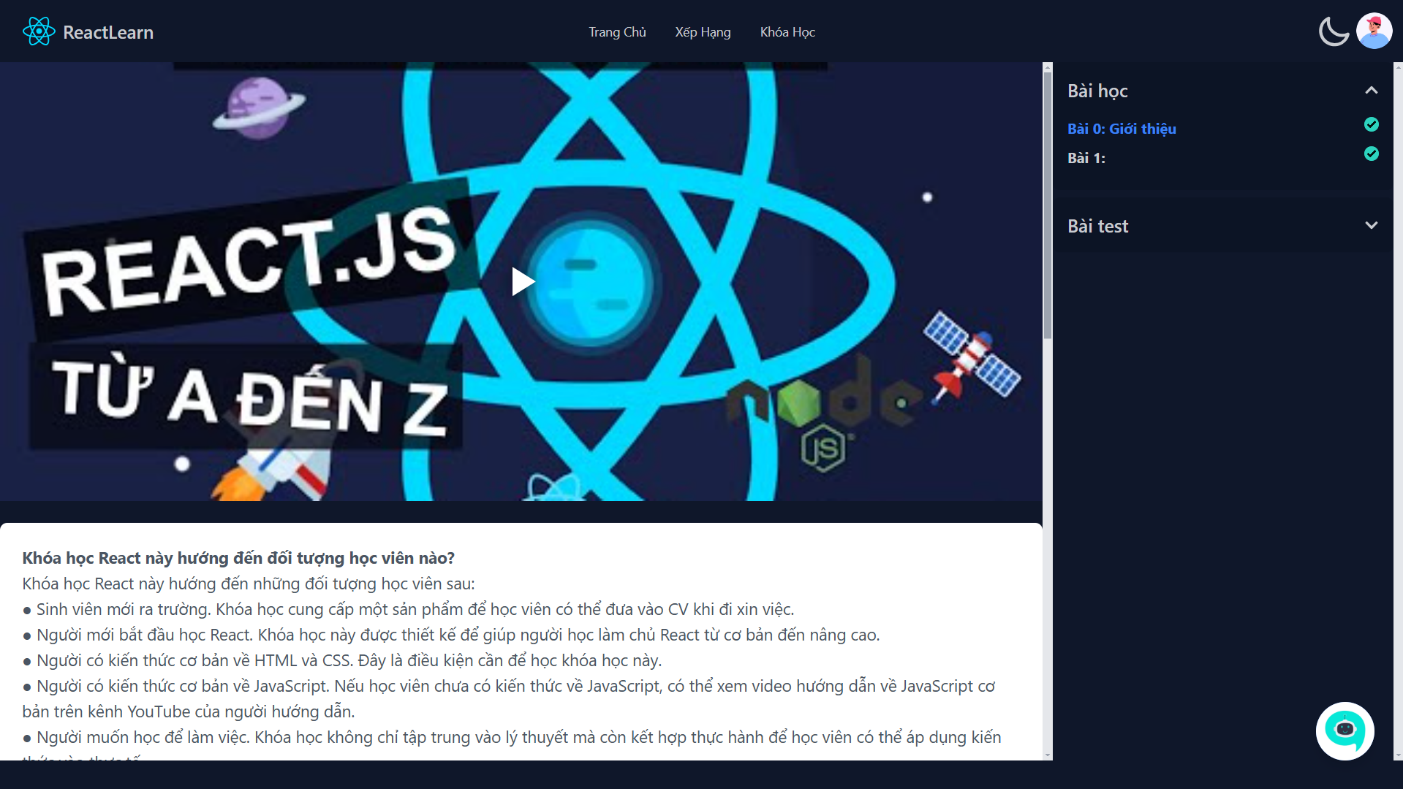
* + - 1. **Giao diện khóa học**

Theo dõi tiến độ học tập thật dễ dàng với thanh tiến trình trực quan hiển thị ngay trên thẻ khóa học. Bên cạnh đó, người học cũng có thể khám phá thêm nhiều khóa học khác phù hợp với nhu cầu của mình ngay tại giao diện này.

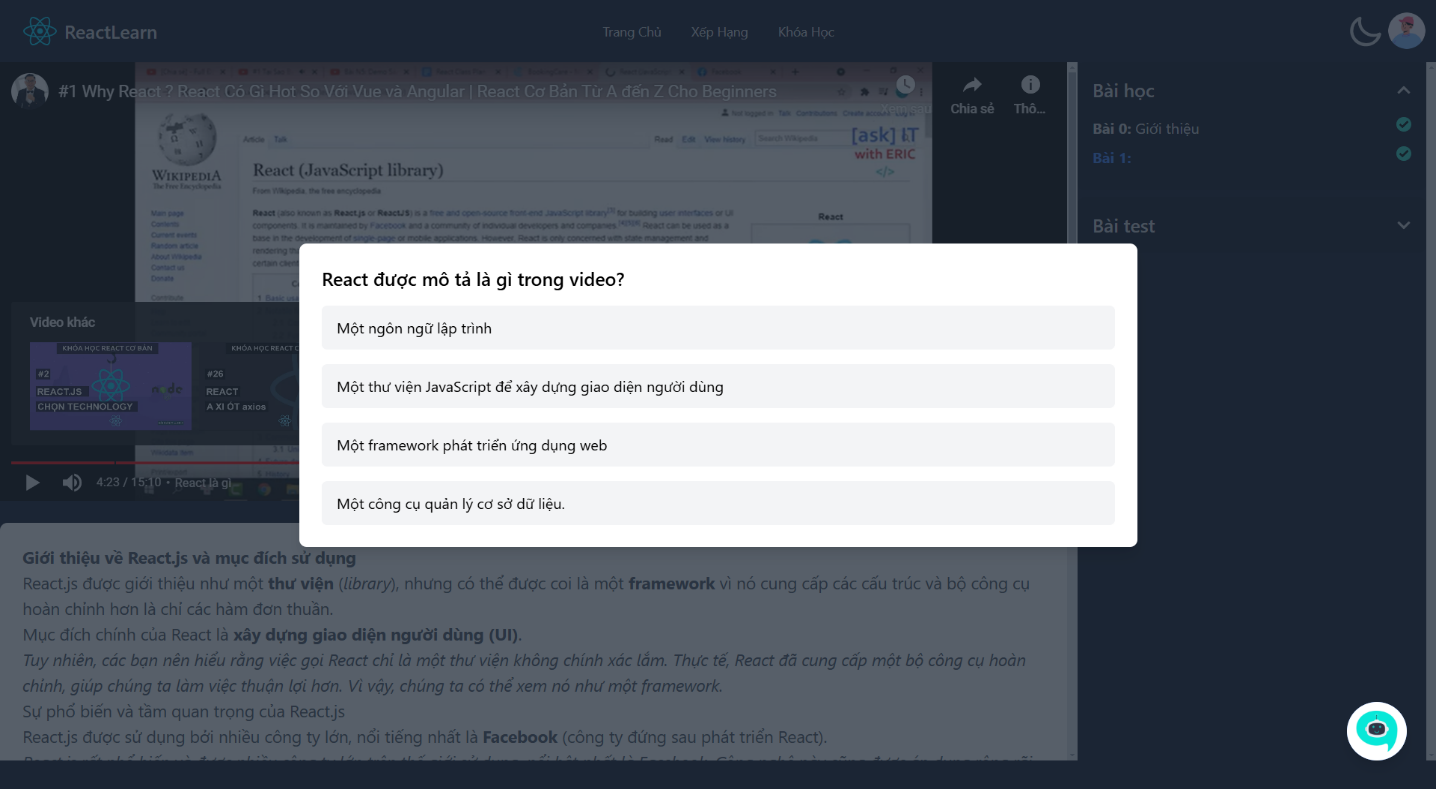
*Hình 1.20 Giao diện khóa học.*

* + - 1. **Giao diện bài học**

Hành trình học tập của học viên bắt đầu từ bài 0 với những video bài giảng chất lượng. Trong quá trình học, các bài tập sẽ xuất hiện để giúp người học củng cố kiến thức ngay lập tức. Nếu trả lời sai quá số lần cho phép, video sẽ tự động quay lại thời điểm trước câu hỏi đó để xem lại nội dung. Sau khi hoàn thành tất cả câu hỏi và video bài giảng, học viên sẽ nhận thông báo "Hoàn thành bài học" cùng biểu tượng dấu tích xanh trên bài học tương ứng.

* Giao diện bài học

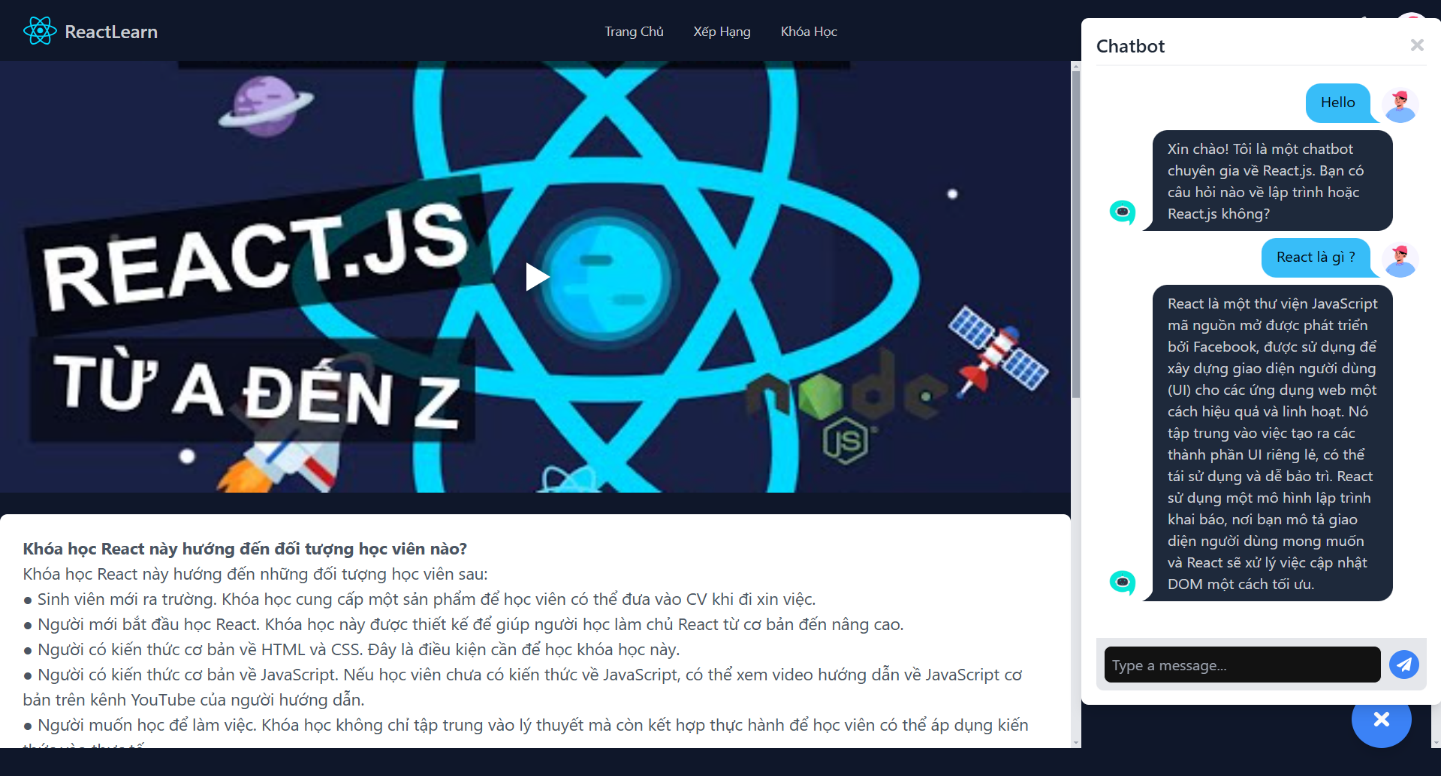
###### *Hình 1.21 Giao diện bài học.*

* Giao diện model bài tập

*Hình 1.22 Giao diện modle bài tập.*

* Giao diện chatbot

Người học có bất kỳ thắc mắc nào về nội dung bài học có thể đặt câu hỏi. Chatbot sẽ trả lời ngay lập tức giải đáp nhanh chóng và chính xác giúp người học củng cố kiến thức vững chắc hơn và vượt qua mọi khó khăn, trong quá trình học tập.

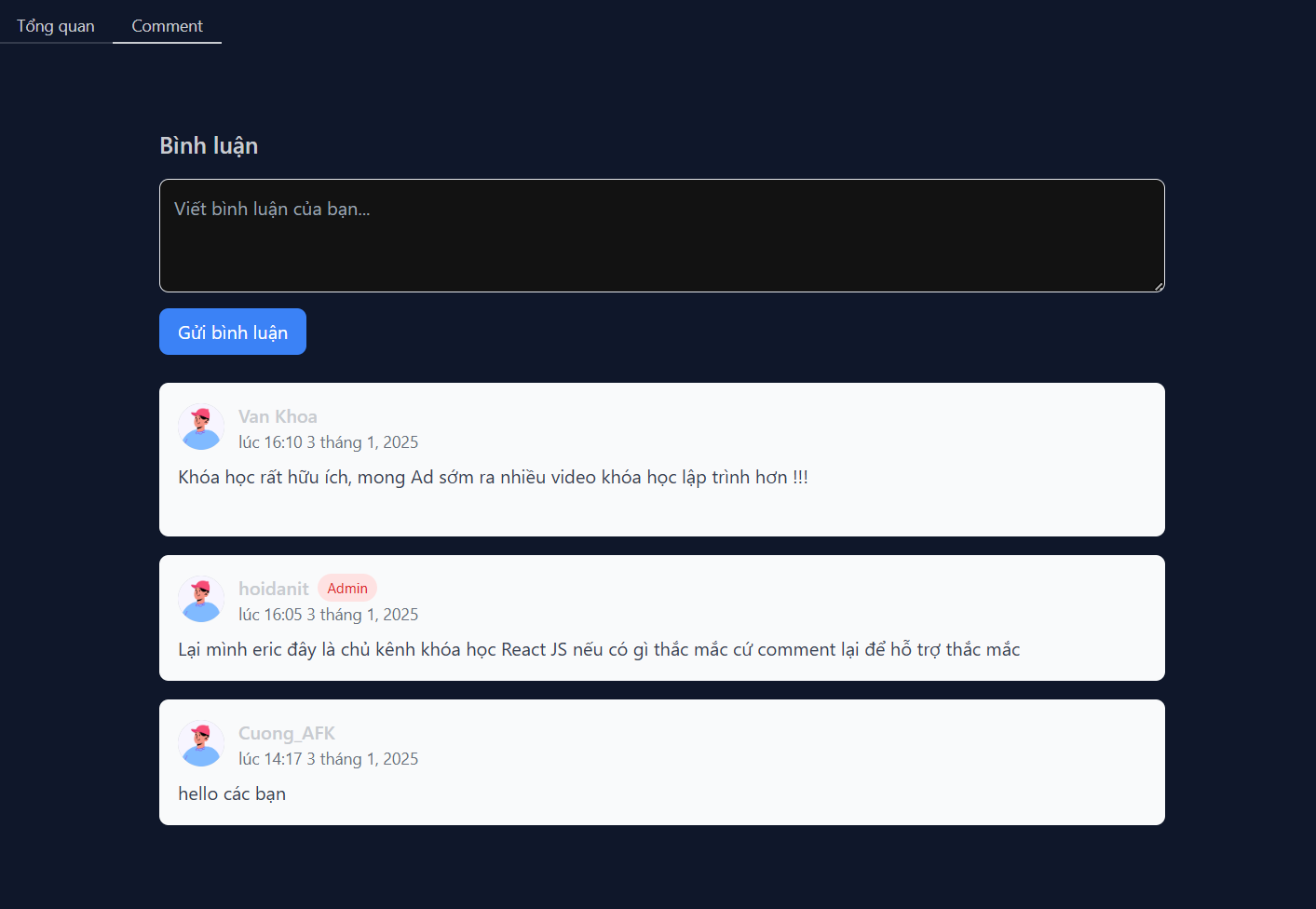
Chatbot hoạt động và xử lý bằng cách gọi API từ trang Google AI Studio [link GoogleAI Studio](https://aistudio.google.com/app/prompts/new_chat" \o "link GoogleAI Studio) .Với cách Data-Augmented Prompting. Chương trình sẽ nhận vào prompt người dùng và kèm đó data mà chương trình có sẵn để cải thiện đầu ra cho người dùng khớp với dữ liệu có sẵn của website mà không phải là nội dung ngoài lề nội dung khóa học.

*Hình 1.23 Giao diện chatbot.*

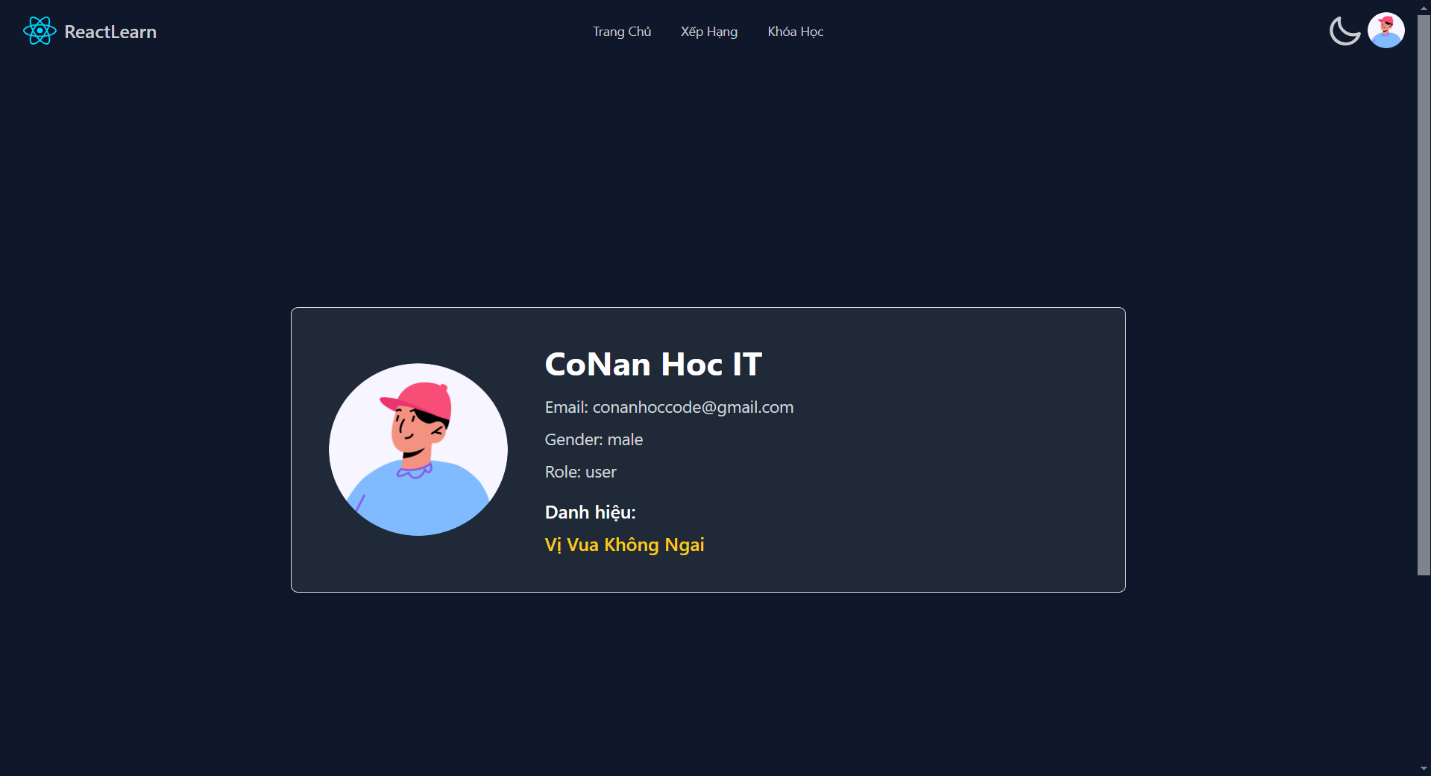
* Giao diện tổng quan

*Hình 1.24 Giao diện tổng quan.*

* Giao diện comment

Giao diện này được thiết kế nhằm mục đích tạo ra một diễn đàn sôi nổi, nơi mọi người có thể kết nối, trao đổi kiến thức và cùng nhau giải đáp thắc mắc. Đây là không gian lý tưởng để chia sẻ với cộng đồng và có cơ hội trò chuyện với chủ khóa học để giải đáp thắc mắc cùng nhau phát triển.

###### *Hình 1.25 Giao diện comment.*

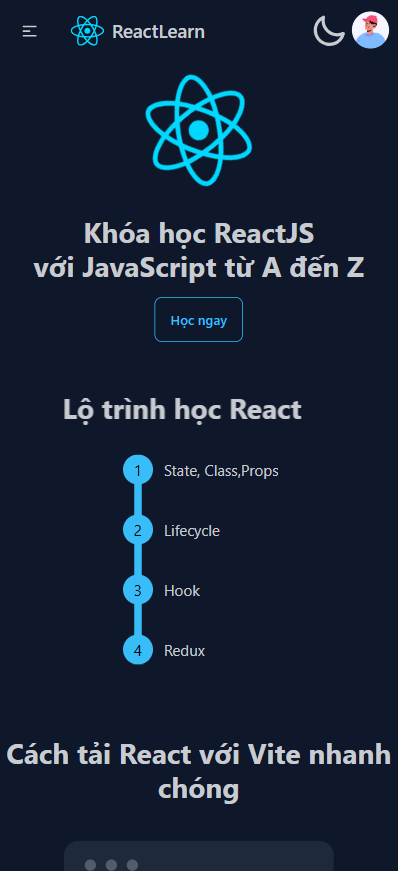
* + - 1. **Giao diện profile**

*Hình 1.26 Giao diện profile.*

* + 1. **Giao diện người dùng ở thiết bị màn hình nhỏ**

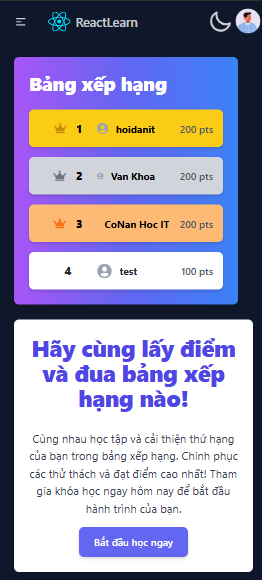
Ngoài việc hiển thị đối với desktop còn reponsive cho mobile nhằm thuận tiện cho người học không có desktop hay màn hình rộng.

* + - 1. **Giao diện trang chủ**

**

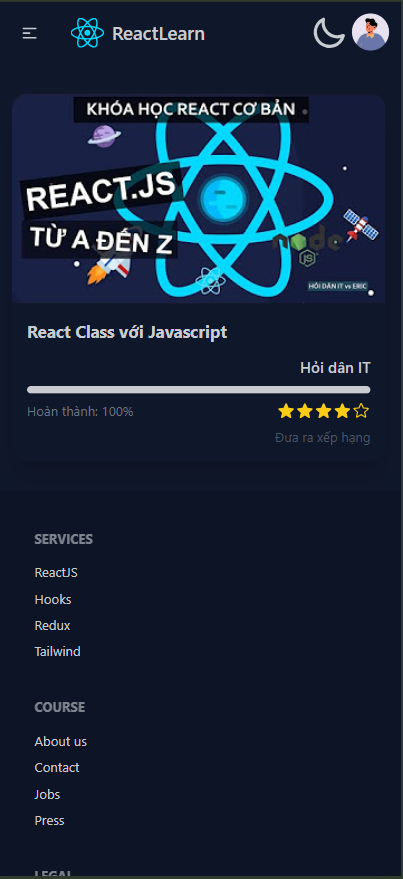
*Hình 1.27 Giao diện trang chủ màn hình nhỏ.*

* + - 1. **Giao diện xếp hạng**

**

*Hình 1.28 Giao diện xếp hạng màn hình nhỏ.*

* + - 1. **Giao diện khóa học**

**

*Hình 1.29 Giao diện khóa học màn hình nhỏ.*

* + - 1. **Giao diện bài học**



*Hình 1.30 Giao diện bài học màn hình nhỏ.*

# PHẦN IV. KẾT LUẬN

# TÀI LIỆU THAM KHẢO

1. [Google Ai Studio](https://ai.google.dev/gemini-api/docs?_gl=1*1tb2weu*_ga*MjU2MTc4MjA1LjE3MjgzMDQ1MTE.*_ga_P1DBVKWT6V*MTczNjA3NzcxNS43Ni4wLjE3MzYwNzc3MTUuNjAuMC4xMDM1ODgxNjU0) ( https://ai.google.dev/gemini-api/docs?\_gl=1\*1tb2weu\*\_ga\*MjU2MTc4MjA1LjE3MjgzMDQ1MTE.\*\_ga\_P1DBVKWT6V\*MTczNjA3NzcxNS43Ni4wLjE3MzYwNzc3MTUuNjAuMC4xMDM1ODgxNjU0 ).
2. [React Document](https://react.dev/learn) ( https://react.dev/learn ).
3. [Nguồn video Khóa Học React JS ( cre HoiDanIt )](https://youtube.com/playlist?list=PLncHg6Kn2JT4C0enPGQPK7ZIlEoZ1ZvRy&si=x0_P_2oN7Ai0rq7o) ( https://youtube.com/playlist?list=PLncHg6Kn2JT4C0enPGQPK7ZIlEoZ1ZvRy&si=x0\_P\_2oN7Ai0rq7o ).
4. [Mongo DB Document](https://www.mongodb.com/docs/manual/) ( https://www.mongodb.com/docs/manual/ ).