**TRƯỜNG ĐẠI HỌC VĂN LANG**

**KHOA CÔNG NGHỆ THÔNG TIN**

Ảnh có chứa văn bản, hình mẫu

Mô tả được tạo tự động

**Bộ môn: Lập Trình Web Nâng Cao**

**Chủ đề: Website bán khóa học online**

|  |  |
| --- | --- |
| **GVHD:** | **Hoàng Công Trình** |
| **Lớp:** | **242\_71ITSE41403\_0502** |
| **Nhóm:** | **08** |
| **Thành Viên:** | **Trần Minh Nhựt**  **- 2274802010622** |
|  | **Trần Huỳnh Thanh Phong**  **- 2274802010659** |
|  | **Huỳnh Đức Thịnh**  **- 2274802010843** |

**Hồ Chí Minh – Năm 2025**

Mục Lục

[LỜI MỞ ĐẦU 3](#_Toc195528478)

[TÓM TẮT ĐỒ ÁN 4](#_Toc195528479)

[CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI 5](#_Toc195528480)

[1.1. Lý do chọn đề tài: 5](#_Toc195528481)

[1.2. Mục tiêu của đề tài: 5](#_Toc195528482)

[1.3. Phạm vi thực hiện: 5](#_Toc195528483)

[CHƯƠNG 2: PHÂN TÍCH YÊU CẦU HỆ THỐNG 6](#_Toc195528484)

[2.1. Giới thiệu về nhận diện cảm xúc từ khuôn mặt 6](#_Toc195528485)

[2.2. Chức năng chính của hệ thống 6](#_Toc195528486)

[2.3. Phân tích người dùng hệ thống: 7](#_Toc195528487)

[2.4. Các yêu cầu phi chức năng: 7](#_Toc195528488)

[CHƯƠNG 3: THIẾT KẾ HỆ THỐNG 8](#_Toc195528489)

[3.1. Sơ đồ Use Case 8](#_Toc195528490)

[3.2. Sơ đồ kiến trúc hệ thống (MVC) 8](#_Toc195528491)

[3.3. Thiết kế giao diện người dùng: 9](#_Toc195528492)

[CHƯƠNG 4: TRIỂN KHAI VÀ KẾT QUẢ THỰC NGHIỆM 10](#_Toc195528493)

[4.1. Xây dựng giao diện người dùng: 10](#_Toc195528494)

[4.2. Backend với Node.js + Express: 10](#_Toc195528495)

[4.3. Cơ sở dữ liệu MongoDB: 10](#_Toc195528496)

[4.4. Kết nối API và kiểm thử chức năng: 11](#_Toc195528497)

[CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 12](#_Toc195528498)

[5.1. Kết luận: 12](#_Toc195528499)

[5.2. Hướng phát triển: 12](#_Toc195528500)

[TÀI LIỆU THAM KHẢO 13](#_Toc195528501)

[PHỤ LỤC 15](#_Toc195528502)

[A. Hình ảnh minh họa kết quả 15](#_Toc195528503)

# LỜI MỞ ĐẦU

Trong suốt quá trình thực hiện và hoàn thiện đồ án “Trang web học tập”, chúng em đã nhận được sự giúp đỡ, hướng dẫn tận tình của thầy cô.Trước hết, chúng em xin bày tỏ lòng biết ơn sâu sắc đến thầy cô đã tận tình hướng dẫn, truyền đạt kiến thức chuyên môn cũng như kinh nghiệm thực tế, giúp chúng em hoàn thành tốt đồ án này.Do kiến thức và kinh nghiệm còn hạn chế, bản báo cáo không tránh khỏi những thiếu sót. Kính mong nhận được sự góp ý của quý thầy cô để em hoàn thiện hơn trong các đồ án tiếp theo.

Em xin chân thành cảm ơn!

# TÓM TẮT ĐỒ ÁN

* Website được xây dựng trên nền tảng web với các chức năng:
  + Đăng ký, đăng nhập người dùng.
  + Người dùng có thể **xem, tìm kiếm, mua khóa học**.
  + Quản trị viên và giảng viên có thể **thêm, chỉnh sửa nội dung khóa học**.
  + Tích hợp **thanh toán online**.
  + Quản lý người học, tiến độ và phản hồi từ học viên.  
    Website sử dụng **JavaScript, Node.js,ReactJs, Express, MongoDB**.

# CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI

## Lý do chọn đề tài:

* Trong thời đại chuyển đổi số, nhu cầu học tập trực tuyến ngày càng tăng nhanh, đặc biệt sau đại dịch COVID-19. Các nền tảng học online giúp người học tiếp cận kiến thức dễ dàng, mọi lúc mọi nơi mà không bị giới hạn bởi không gian và thời gian.
* Tuy nhiên, không phải hệ thống nào cũng thân thiện, dễ sử dụng hoặc có chi phí hợp lý. Vì vậy, việc xây dựng một website bán khóa học online đơn giản, tiện dụng, dễ mở rộng là cần thiết để đáp ứng nhu cầu thực tế, đồng thời giúp sinh viên tiếp cận với quy trình phát triển hệ thống web hiện đại.

## Mục tiêu của đề tài:

* Thiết kế và phát triển một hệ thống website học trực tuyến cho phép:
  + Giảng viên: tạo, quản lý và chia sẻ khóa học.
  + Học viên: xem, mua và học khóa học.
* Cung cấp giao diện thân thiện, dễ sử dụng
* Tích hợp các chức năng cần thiết như:
  + Quản lý người dùng.
  + Quản lý nội dung khóa học.
  + Theo dõi tiến độ học tập.
  + Tìm kiếm và lọc khóa học.
  + Thanh toán online (giả lập).

## Phạm vi thực hiện:

* Đề tài tập trung xây dựng một website bán khóa học cơ bản, hướng đến đối tượng:
  + **Giảng viên**: Đăng tải, quản lý khóa học và theo dõi học viên.
  + **Học viên**: Đăng ký, tìm kiếm, mua khóa học và theo dõi tiến độ học tập.
  + **Quản trị viên**: Quản lý hệ thống người dùng, phê duyệt khóa học (tùy chọn).

# CHƯƠNG 2: PHÂN TÍCH YÊU CẦU HỆ THỐNG

## 2.1. Giới thiệu về nhận diện cảm xúc từ khuôn mặt

* Phân tích yêu cầu là bước quan trọng trong quá trình xây dựng hệ thống, giúp xác định các chức năng cần thiết, các đối tượng sử dụng hệ thống và cách hệ thống vận hành.
* Hệ thống website bán khóa học online cần đảm bảo các chức năng cơ bản để hỗ trợ quá trình học tập trực tuyến, đồng thời dễ sử dụng và có khả năng mở rộng trong tương lai.

## 2.2. Chức năng chính của hệ thống

* Đăng nhập / Đăng ký người dùng
  + Cho phép người dùng tạo tài khoản (Học viên, Giảng viên).
  + Đăng nhập bằng email và mật khẩu.
  + Phân quyền truy cập dựa trên loại người dùng.
* Xem danh sách khóa học / tìm kiếm / lọc
  + Hiển thị tất cả khóa học có trên hệ thống.
  + Tìm kiếm theo tên khóa học, giảng viên.
  + Lọc theo chủ đề: lập trình, thiết kế, marketing,…
* Mua khóa học
  + Học viên có thể mua khóa học.
  + Thanh toán mô phỏng.
* Theo dõi tiến độ học
  + Sau khi mua khóa học, học viên có thể xem nội dung theo từng bài giảng.
  + Hệ thống lưu lại trạng thái học.
* Quản lý khóa học
  + Tạo mới khóa học, cập nhật nội dung, video, mô tả, tài liệu.
  + Quản lý học viên tham gia khóa học.
* Quản trị người dùng
  + Admin có thể duyệt khóa học mới, xóa/sửa tài khoản người dùng.
  + Quản lý phân quyền hệ thống.

## 2.3. Phân tích người dùng hệ thống:

|  |  |
| --- | --- |
| **Loại người dùng** | **Mô tả chức năng** |
| Người dùng | Đăng ký tài khoản, đăng nhập  Xem và mua khóa học  Học và theo dõi tiến độ |
| Giảng viên | Đăng nhập và tạo khóa học  Quản lý nội dung, theo dõi học viên đăng ký |
| Quản trị viên | Quản lý toàn hệ thống  Phê duyệt khóa học, quản lý người dùng và phân quyền |

## 2.4. Các yêu cầu phi chức năng:

* Hiệu năng: Tốc độ tải trang nhanh, phản hồi tốt.
* Bảo mât: Tài khoản được mã hóa mật khẩu, phân quyền rõ ràng.
* Khả nặng mở rộng: Dễ dàng thêm mới khóa học, người dùng, chủ đề.
* Thân thiện người dùng: Giao diện đơn giản, dễ thao tác trên cả máy tính và điện thoại.

# CHƯƠNG 3: THIẾT KẾ HỆ THỐNG

## 3.1. Sơ đồ Use Case

* Sơ đồ Use Case mô tả các chức năng chính của hệ thống và mối quan hệ giữa người dùng với các chức năng đó.
* Các tác nhân (Actor):
  + Học viên
  + Giảng viên
  + Quản trị viên
* Một số Use Case chính:
  + Học viên: Đăng ký, đăng nhập, xem khóa học, mua khóa học, học bài, theo dõi tiến độ.
  + Giảng viên: Tạo khóa học, cập nhật nội dung, quản lý học viên trong khóa.
  + Admin: Quản lý người dùng, duyệt khóa học, phân quyền.

## 3.2. Sơ đồ kiến trúc hệ thống (MVC)

* Quản lý người dùng, duyệt khóa học, phân quyền.

|  |  |
| --- | --- |
| **Thành phần** | **Chức năng** |
| Model | Tương tác cơ sở dữ liệu: người dùng, khóa học, đơn hàng |
| View | Giao diện người dùng: HTML, CSS, React/Bootstrap |
| Controller | |  | | --- | |  |  |  | | --- | | Xử lý logic và điều hướng luồng dữ liệu giữa Model và View | |

* Kiến trúc công nghệ (ví dụ với MERN Stack):
  + Frontend (View): ReactJS
  + Backend (Controller): Express.js
  + Database (Model): MongoDB
  + Server: Node.js

## 3.3. Thiết kế giao diện người dùng:

* Website được thiết kế thân thiện, đơn giản, dễ thao tác với 4 giao diện chính:
* Trang chủ:
  + Hiển thị danh sách các khóa học nổi bật.
  + Hiển thị danh sách giảng viên nổi bật.
* Trang đăng nhập / đăng ký:
  + Đăng nhập dành cho học viên, giảng viên, quản trị viên.
  + Form đăng ký với xác thực cơ bản.
* Trang danh sách khóa học:
  + Hiển thị các khóa học theo danh mục hoặc giảng viên.
  + Có nút mua hoặc xem chi tiết.
* Trang chi tiết khóa học:
  + Hiển thị nội dung khóa học: mô tả, video, tài liệu đính kèm.
  + Hiển thị tiến độ học nếu đã đăng ký.

# CHƯƠNG 4: TRIỂN KHAI VÀ KẾT QUẢ THỰC NGHIỆM

## 4.1. Xây dựng giao diện người dùng:

* Giao diện được xây dựng bằng công nghệ **React.js** kết hợp với **HTML/CSS/Bootstrap** để đảm bảo:
  + Thân thiện người dùng
  + **Responsive**
  + **Tải nhanh, dễ điều hướng**
* Một số thành phần chính: trang chủ, trang chi tiết khóa học, trang đăng ký/ đăng nhập, trang học viên, trang giảng viên

## 4.2. Backend với Node.js + Express:

* Backend với Node.js + Express
  + Xử lý các yêu cầu HTTP (GET, POST, PUT, DELETE)
  + Xác thực người dùng bằng **JWT (JSON Web Token)**
  + Phân quyền theo vai trò (học viên, giảng viên, admin)

## 4.3. Cơ sở dữ liệu MongoDB:

* Cơ sở dữ liệu được triển khai bằng **MongoDB** (NoSQL) – phù hợp với cấu trúc dữ liệu động, dễ mở rộng.
* Các bảng chính:
  + Applied
  + Cart (giỏ hàng)
  + Classes (lớp)
  + Enrollments (them vào lớp học)
  + Payments (Thanh toán)
  + seoSettings (Thông tin layout)
  + users (tài khoản)

## 4.4. Kết nối API và kiểm thử chức năng:

* Đăng ký / Đăng nhập:
  + Kiểm tra xác thực người dùng
  + Lưu token vào localStorage
* Mua khóa học:
  + Ghi danh vào bảng Enrollments
  + Mô phỏng thanh toán thành công
* Theo dõi tiến độ học:
  + Giao diện hiển thị tiến độ (% bài học đã hoàn thành)
  + Cập nhật mỗi khi người học xem bài giảng
* Quản lý khóa học:
  + Giảng viên có thể thêm/sửa/xóa bài học
  + Quản lý học viên trong khóa

# CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## 5.1. Kết luận:

* Qua quá trình nghiên cứu, thiết kế và triển khai, đề tài **“Website bán khóa học online”** đã đạt được các mục tiêu đặt ra:
  + Hệ thống hoạt động **ổn định**, cung cấp đầy đủ các chức năng cơ bản phục vụ việc học trực tuyến như: đăng ký, xem khóa học, mua khóa học, theo dõi tiến độ học, quản lý nội dung.
  + Website được xây dựng với kiến trúc **hiện đại**, sử dụng **MERN Stack** giúp dễ bảo trì và mở rộng.
  + **Giao diện thân thiện, responsive**, đảm bảo trải nghiệm người dùng tốt trên cả desktop và thiết bị di động.
  + Hệ thống **phân quyền rõ ràng**: học viên – giảng viên – quản trị viên, phục vụ đúng vai trò của từng nhóm người dùng.

## 5.2. Hướng phát triển:

* Tích hợp thanh toán thật:
  + Kết hợp framework thanh toán Stripe để xử lý thanh toán.
  + Tự động xác nhận học viên sau khi thanh toán thành công.
* Thêm tính năng học trực tuyến:
  + Tích hợp **Zoom SDK** hoặc **WebRTC** để giảng viên dạy học trực tiếp.
  + Cho phép học viên tương tác, đặt câu hỏi theo thời gian thực.
* Xây dựng ứng dụng mobile
  + Sử dụng React Native hoặc Flutter để phát triển ứng dụng học tập trên điện thoại.
  + Tăng khả năng tiếp cận và học tập linh hoạt hơn.
* Cá nhân hóa đề xuất khóa học (AI/ML)
  + Ứng dụng trí tuệ nhân tạo để gợi ý khóa học phù hợp với người dùng

# TÀI LIỆU THAM KHẢO

* [Xác minh các JSON Web Tokens (JWT)](https://www.npmjs.com/package/jsonwebtoken)
* [Stripe nền tảng xử lý thanh toán trực tuyến](https://dashboard.stripe.com/test/apikeys)
* [Stripe Elements bộ các thành phần UI do Stripe cung cấp, giúp bạn dễ dàng tích hợp thanh toán thẻ vào ứng dụng của mình mà không phải lo lắng về bảo mật và tuân thủ PCI DSS (chuẩn bảo mật thẻ thanh toán).](https://docs.stripe.com/js/elements_object/create_element?type=card)
* [Stripe Dashboard - Workbench Logs. quản lý phản hồi request khi thanh toán](https://dashboard.stripe.com/test/workbench/logs?direction%5b0%5d=self&direction%5b1%5d=connect_in&method%5b0%5d=post&method%5b1%5d=delete&method%5b2%5d=get&source%5bapi%5d=true&source%5bdashboard%5d=true&filtered=true&selected=req_WfTA4j0T8S8EVN)
* [Doc stripe](https://docs.stripe.com/payments/quickstart?client=react&lang=node)[React Toastify thư viện giúp hiển thị các thông báo tạm thời (toast notifications) trong ứng dụng React.](https://www.npmjs.com/package/react-toastify)
* [Lấy các giá trị thẻ ảo test thanh toán online](https://docs.stripe.com/testing)
* [Test Token](https://jwt.io/#debugger-io')[cài đặt và sử dụng Tailwind CSS với Vite.](https://v3.tailwindcss.com/docs/guides/vite)[TanStack Query thư viện giúp bạn quản lý trạng thái và dữ liệu từ xa trong ứng dụng React.](https://tanstack.com/query/latest/docs/framework/react/overview)
* [Create Component điều hướng NavLink](https://reactrouter.com/6.28.0/start/tutorial#active-link-styling)
* [Theme Scoping trong MUI kiểm soát cách các theme được áp dụng trong các thành phần](https://mui.com/material-ui/integrations/theme-scoping/)
* [Swiper thư viện rất phổ biến dùng để tạo các slider (carousel) và swiper (vuốt)](https://swiperjs.com/react#styles)
* [React Icons thư viện cho phép bạn sử dụng hơn 2.000 biểu tượng khác nhau trong ứng dụng React.](https://react-icons.github.io/react-icons/)
* [Transition của Headless UI giúp bạn dễ dàng thêm các hiệu ứng chuyển động](https://headlessui.com/v1/react/transition)
* [Vite công cụ xây dựng (build tool) mới, được tối ưu hóa cho trải nghiệm phát triển nhanh chóng.](https://vite.dev/guide/env-and-mode)
* [Axios thư viện HTTP client phổ biến cho các ứng dụng JavaScript, được sử dụng để gửi và nhận dữ liệu từ các API hoặc server](https://axios-http.com/vi/docs/interceptors)
* [React Hook Form thư viện nhẹ và hiệu quả để quản lý biểu mẫu trong ứng dụng React.](https://react-hook-form.com/get-started)
* [React Spinners (Loader) thư viện chứa các thành phần "spinner" được thiết kế đẹp mắt và dễ sử dụng cho các ứng dụng React.](https://www.davidhu.io/react-spinners/storybook/?path=/docs/hashloader--docs)
* [SweetAlert2 thư viện JavaScript cho phép tạo các hộp thoại thông báo (alerts) tùy chỉnh với giao diện đẹp mắt và các tính năng mở rộng.](https://sweetalert2.github.io/#examples)
* [ImgBB cung cấp một API giúp bạn tải lên hình ảnh lên nền tảng của họ, lấy URL của hình ảnh và quản lý hình ảnh trực tiếp trong các ứng dụng hoặc trang web của bạn.](https://api.imgbb.com/)
* [Component Pagination của MUI giúp bạn dễ dàng thêm tính năng phân trang vào ứng dụng React của mình.](https://mui.com/material-ui/react-pagination/)
* [Thư viện quản lý state](https://tanstack.com/query/latest/docs/framework/react/overview)
* [ChatGPT](https://chatgpt.com/)

# PHỤ LỤC

## Hình ảnh minh họa kết quả