

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



ĐỒ ÁN MÔN HỌC: CHUYÊN ĐỀ FRONT-END
ĐỀ TÀI: XÂY DỰNG WEBSITE QUẢN LÝ NGƯỜI DÙNG

.....



Giáo viên hướng dẫn : Nguyễn Minh Hải

Sinh viên thực hiện: Hồ Thị Nguyệt Gấm - 501240317

Khoá : 24

TP. Hồ Chí Minh, ngày 16 tháng 10 năm 2025

LỜI MỞ ĐẦU

Trong thời đại công nghệ số phát triển nhanh chóng như hiện nay, internet và các ứng dụng web đã trở thành một phần không thể tách rời trong cuộc sống hằng ngày của con người. Từ việc tra cứu thông tin, mua sắm trực tuyến, làm việc từ xa cho đến học tập và giải trí, mọi hoạt động đều có sự hiện diện của các nền tảng số. Trong đó, giao diện người dùng – hay còn gọi là Front End – đóng vai trò quan trọng vì đây là phần đầu tiên mà người dùng nhìn thấy và tương tác khi truy cập vào một trang web hoặc ứng dụng. Một giao diện được thiết kế đẹp mắt, bố cục hợp lý và dễ sử dụng sẽ tạo ấn tượng tốt, giữ chân người dùng và nâng cao trải nghiệm tổng thể.

Đồ án Front End được thực hiện nhằm mục tiêu vận dụng toàn bộ những kiến thức đã học về HTML, CSS, JavaScript cũng như tìm hiểu thêm một số thư viện, framework hỗ trợ để xây dựng một giao diện web hoàn chỉnh. Trong quá trình thực hiện, nhóm/Em tập trung vào việc xây dựng bố cục rõ ràng, sắp xếp hợp lý các thành phần trên trang, lựa chọn màu sắc phù hợp và thêm các hiệu ứng đơn giản để tăng tính sinh động cho trang web. Ngoài ra, việc đảm bảo trang web hiển thị tốt trên nhiều loại thiết bị như máy tính, máy tính bảng và điện thoại cũng là một trong những tiêu chí quan trọng của đồ án.

Bên cạnh phần giao diện, quá trình làm đồ án còn giúp nhóm/Em rèn luyện kỹ năng tư duy logic, giải quyết các vấn đề phát sinh trong thực tế. Từ những lỗi nhỏ về cú pháp, cách căn chỉnh bố cục cho đến việc xử lý tương tác bằng JavaScript, mỗi bước đều là một cơ hội để học hỏi thêm và củng cố kiến thức đã được giảng dạy trên lớp. Đây cũng là dịp để nhóm/Em làm quen với cách làm việc giống như trong môi trường thực tế, từ khâu lên ý tưởng, thiết kế bố cục cho đến triển khai mã nguồn và kiểm thử sản phẩm.

Ngoài việc rèn luyện kỹ năng kỹ thuật, quá trình thực hiện đồ án còn giúp nhóm/Em phát triển thêm kỹ năng làm việc nhóm, phân chia công việc hợp lý và hỗ trợ lẫn nhau khi gặp khó khăn. Việc cùng nhau trao đổi, thảo luận và hoàn thiện sản phẩm giúp nâng cao tinh thần hợp tác cũng như khả năng giao tiếp trong môi trường làm việc tập thể.

Thông qua đồ án này, nhóm/Em không chỉ có cơ hội củng cố lại kiến thức lý thuyết mà còn học được cách áp dụng chúng vào những tình huống cụ thể. Đây là một bước quan trọng giúp chuẩn bị tốt hơn cho những dự án lớn hơn trong tương lai, cũng như tạo nền tảng vững chắc cho con đường làm việc trong lĩnh vực phát triển web sau này.

LỜI CẢM ƠN

Em xin gửi lời cảm ơn chân thành đến quý thầy cô trong khoa đã tận tình giảng dạy, truyền đạt những kiến thức bổ ích và tạo điều kiện học tập thuận lợi trong suốt quá trình học tập. Những bài giảng, chia sẻ và kinh nghiệm thực tế mà thầy cô mang đến đã giúp em có được nền tảng vững chắc để vận dụng vào việc thực hiện đồ án này cũng như trong quá trình học tập sau này.

Đặc biệt, em xin gửi lời cảm ơn sâu sắc đến [Tên giảng viên hướng dẫn] – người đã luôn tận tâm hỗ trợ, hướng dẫn và đưa ra những góp ý quý báu trong suốt quá trình thực hiện đồ án. Nhờ sự chỉ dẫn tận tình của thầy/cô, em đã có thêm nhiều kiến thức thực tế, biết cách nhìn nhận và giải quyết vấn đề một cách khoa học, qua đó hoàn thiện sản phẩm một cách tốt nhất có thể.

Bên cạnh đó, em cũng xin cảm ơn các anh chị và bạn bè đã luôn đồng viên, chia sẻ kinh nghiệm, hỗ trợ em trong quá trình tìm hiểu và thực hiện đồ án. Những sự giúp đỡ quý giá đó là nguồn động lực lớn để em hoàn thành bài làm này.

Một lần nữa, em xin chân thành cảm ơn tất cả thầy cô, bạn bè và những người đã đồng hành, giúp đỡ em trong suốt thời gian qua.

MỤC LỤC

CHƯƠNG 1: Giới thiệu đề tài

CHƯƠNG 2: Tài liệu tham khảo

CHƯƠNG 3: Mục tiêu

CHƯƠNG 4: Công nghệ sử dụng

CHƯƠNG 5: Phân tích & thiết kế giao diện

5.1 Bố cục chính

5.2 Quy ước Font & Màu

5.3 Responsive (Tương thích đa thiết bị)

CHƯƠNG 6: Cài đặt & Cách chạy

6.1 Yêu cầu môi trường

6.2 Các bước cài đặt

6.3 Cấu trúc thư mục

6.4 Kiểm tra hiển thị

6.5 Kết luận

CHƯƠNG 7: Kết quả (ảnh minh họa)

CHƯƠNG 8: Hạn chế & Hướng phát triển

8.1 Hạn chế hiện tại

8.2 Hướng phát triển

CHƯƠNG 9: Quy trình thực hiện

CHƯƠNG 10: Kiến thức học được

CHƯƠNG 11: Kiểm thử và đánh giá

CHƯƠNG 12: Kế hoạch mở rộng trong tương lai

CHƯƠNG 13: Nhận xét & Kết luận tổng quát

Chương 1. GIỚI THIỆU

Website quán cà phê là một trong những dạng website phổ biến hiện nay, đặc biệt trong lĩnh vực dịch vụ – ẩm thực. Không chỉ đơn giản là nơi cung cấp thông tin về quán, menu hay địa chỉ, các website hiện đại còn giúp tạo ấn tượng thương hiệu, hỗ trợ đặt bàn online, giới thiệu không gian quán và thu hút khách hàng thông qua hình ảnh đẹp mắt.

Đề tài đồ án Front-End lần này tập trung xây dựng một website quán cà phê với phong cách hiện đại, nhẹ nhàng, sử dụng các công nghệ cơ bản như HTML, CSS, Bootstrap để thiết kế bố cục giao diện thân thiện, dễ nhìn và hoạt động tốt trên cả máy tính lẫn thiết bị di động.

Website bao gồm các phần chính:

- **Header + Navbar:** Có ảnh nền, menu điều hướng, icon mạng xã hội.
- **Carousel (banner):** Tự động chuyển slide, hiển thị hình ảnh quảng bá nổi bật.
- **Các section nội dung:** Giới thiệu quán, menu sản phẩm, không gian quán.
- **Footer:** Thông tin liên hệ, mạng xã hội, bản quyền.

Mục tiêu chính là xây dựng một website có giao diện trực quan, dễ sử dụng, đẹp mắt, đúng chuẩn responsive và áp dụng đúng kiến thức đã học trong môn Front-End.

CHƯƠNG 2. TÀI LIỆU THAM KHẢO

Trong quá trình thực hiện, em đã tham khảo nhiều nguồn tài liệu khác nhau bao gồm:

- <https://getbootstrap.com> — Tài liệu chính thức của Bootstrap 5.
- <https://www.w3schools.com> — Học cú pháp HTML/CSS/JS cơ bản.
- <https://developer.mozilla.org> — MDN Docs, tham khảo chi tiết về các thuộc tính CSS & HTML.
- Video hướng dẫn xây dựng website responsive trên YouTube.
- Một số trang web của quán cà phê thực tế để tham khảo ý tưởng thiết kế.

CHƯƠNG 3: Mục tiêu của đồ án là:

- Vận dụng kiến thức đã học để xây dựng một website giao diện hoàn chỉnh.
- Thiết kế giao diện website quán cà phê thân thiện, dễ sử dụng.
- Đảm bảo bố cục rõ ràng, màu sắc hài hòa, phù hợp phong cách quán.
- Ứng dụng Bootstrap để thiết kế responsive, giúp website hiển thị tốt trên điện thoại, máy tính bảng và PC.
- Tạo carousel có hiệu ứng chuyển slide tự động, icon mạng xã hội, phần header có hình nền.
- Thực hiện báo cáo chi tiết quá trình từ thiết kế, cài đặt, kiểm thử đến hoàn thiện.
- Hình thành tư duy thiết kế giao diện người dùng chuyên nghiệp.

CHƯƠNG 4. CÔNG NGHỆ SỬ DỤNG

Trong quá trình xây dựng website, em đã sử dụng các công nghệ cơ bản và phổ biến hiện nay để đảm bảo website vừa dễ triển khai, vừa có tính ổn định cao. Cụ thể như sau:

4.1. HTML (HyperText Markup Language)

HTML là ngôn ngữ đánh dấu được sử dụng để tạo khung và cấu trúc nội dung cho website. Trong đồ án, em đã sử dụng HTML5 để xây dựng các phần chính như header, carousel, nội dung, footer.

4.2. CSS (Cascading Style Sheets)

CSS được dùng để định dạng giao diện website, từ màu sắc, kích thước chữ, bố cục đến hiệu ứng khi người dùng tương tác.

Trong dự án, em đã sử dụng CSS để:

- Tạo màu nền và hình nền cho header.
- Căn chỉnh bố cục của các phần như carousel, section menu, footer.
- Làm cho các icon mạng xã hội hiển thị đẹp và có hiệu ứng hover.
- Định dạng font chữ, khoảng cách giữa các tiêu đề, căn giữa các phần tử.

4.3. Bootstrap 5

Bootstrap là framework CSS phổ biến giúp xây dựng giao diện nhanh chóng, đồng thời hỗ trợ responsive tự động.

Trong đồ án, Bootstrap được dùng cho:

- **Grid system:** phân chia cột (col-12, col-md-6...) để giao diện tự co giãn theo thiết bị.
- **Navbar component:** tạo menu đẹp mắt có thể ẩn/hiện khi thu nhỏ màn hình.
- **Carousel component:** sử dụng sẵn của Bootstrap để tạo slider ảnh chuyển động tự động.
- **Utilities classes:** căn chỉnh margin, padding nhanh gọn.

4.4. JavaScript

JavaScript được dùng để xử lý một số hiệu ứng và hành vi động, ví dụ như:

- Tự động chuyển slide carousel sau vài giây.
- Xử lý sự kiện click của menu khi ở chế độ mobile.
- Cuộn mượt khi click vào các liên kết trong navbar.

4.5. Công cụ hỗ trợ

Ngoài ra, em còn sử dụng thêm các công cụ sau:

- **VS Code:** Trình soạn thảo chính.
- **Live Server:** Xem website trực tiếp trong quá trình code.
- **GitHub:** Lưu trữ và quản lý mã nguồn.
- **Trình duyệt Chrome/Edge:** Kiểm tra giao diện và debug.

CHƯƠNG 5. PHÂN TÍCH & THIẾT KẾ GIAO DIỆN

Để website có bố cục hợp lý, em đã tiến hành thiết kế giao diện trước khi bắt tay vào code. Việc này giúp xác định rõ vị trí của từng thành phần, tránh việc sửa nhiều lần khi lập trình.

5.1. Bố cục chính

Bố cục của website được chia thành các phần sau:

- **Header:** Chiếm toàn bộ chiều ngang trang, có ảnh nền, logo quán ở góc trái, menu nằm

ngang ở giữa, icon mạng xã hội bên phải.

- **Carousel:** Nằm ngay dưới header, chiếm chiều rộng toàn trang, hiển thị các hình ảnh nổi bật.
- **Phần giới thiệu:** Trình bày thông tin về quán, phong cách phục vụ, không gian.
- **Phần menu sản phẩm:** Hiển thị các loại đồ uống kèm hình ảnh.
- **Phần liên hệ / bản đồ:** Gắn Google Map + thông tin liên hệ.
- **Footer:** Màu nền tối, chứa thông tin bản quyền và mạng xã hội.

(Ảnh minh họa bố cục có thể chiếm 1 trang Word trong báo cáo.)

5.2. Quy ước Font & Màu

- **Font chữ:** Sử dụng font mặc định của Bootstrap kết hợp font Google như “Red Hat Display” để tạo cảm giác hiện đại.
- **Màu chủ đạo:** Tông nâu – trắng, tạo cảm giác ấm cúng, gần gũi như phong cách quán cà phê.
- **Màu nhấn:** Màu cam nhẹ dùng cho nút bấm và tiêu đề nổi bật.

5.3. Responsive (Tương thích đa thiết bị)

Giao diện được thiết kế theo nguyên tắc mobile-first, sau đó mở rộng lên tablet và desktop. Nhờ Bootstrap, các thành phần sẽ tự động xếp chồng khi màn hình nhỏ, và chia cột khi màn hình lớn.

Ví dụ:

- Ở màn hình nhỏ: menu sẽ chuyển thành nút toggle.
- Carousel vẫn full width, ảnh tự co giãn.
- Phần sản phẩm chuyển từ 4 cột sang 1–2 cột.

CHƯƠNG 6. CÀI ĐẶT & CÁCH CHẠY

6.1. Yêu cầu môi trường

- Trình duyệt hiện đại (Chrome, Edge, Firefox)
- Visual Studio Code + tiện ích Live Server

- Kết nối Internet để tải Bootstrap CDN

6.2. Các bước cài đặt

1. Tải mã nguồn về từ GitHub hoặc folder nén.
2. Mở thư mục bằng VS Code.
3. Chạy Live Server (hoặc mở file index.html bằng trình duyệt).
4. Kiểm tra các phần header, carousel, section... đã hoạt động.

6.3. Cấu trúc thư mục

BaoCaoDoAn/

```
|  
|— index.html  
|— main.css  
|— /images  
|   |— deco.jpg  
|   |— a1.webp  
|   |— a2.webp  
|   |— a3.webp  
|   |— a4.webp  
|   |— a5.webp  
|   |— a6.webp  
|   |— a7.webp  
|   |— a8.webp  
|   |— a9.webp  
|   |— khonggian.webp  
|   |— rongrai.jpg  
|   |— logo.jpg  
|   |— nexflit.jpg
```

| |— nuoc.jpg

| |— trangtri.jpg

6.4. Kiểm tra hiển thị

Sau khi cài đặt xong, mở trình duyệt để kiểm tra giao diện ở các độ phân giải khác nhau (PC, tablet, mobile). Nếu mọi thứ hiển thị đúng → website đã sẵn sàng.

6.5. Kết luận

Phần cài đặt khá đơn giản nhờ sử dụng HTML tĩnh và Bootstrap CDN, không cần cấu hình server phức tạp.

CHƯƠNG 7. Kết quả (Ảnh minh họa)

- Trang chủ hiển thị đầy đủ banner, carousel, các khối nội dung rõ ràng.
- Các section chuyển tiếp mượt mà, bố cục cân đối.
- Giao diện đẹp trên máy tính, điện thoại, máy tính bảng.
- Hiệu ứng hover, animation nhẹ làm trang sinh động hơn.
- Thời gian tải trang nhanh (do tối ưu hình ảnh, dùng CDN cho Bootstrap).
- Các phần chính:
 - Header: nền hình ảnh đẹp, menu nổi bật.
 - Carousel: trượt mượt, nội dung căn giữa.
 - Thân trang: bố cục rõ ràng, font hài hòa.
 - Footer: liên kết mạng xã hội + thông tin liên hệ.



Mô chi Mô chi

Trang chủ Giới thiệu Dịch vụ Liên hệ



Không gian riêng tư



Menu nước đa dạng



Sạch sẽ, thoáng mát



Chill Chill!!

Mochi Cafe In Bed: Chúng tôi là nơi mà bạn có thể relax sau những giờ làm việc, học tập mệt mỏi. Với thiết kế dưới dạng sleepbox, có rèm kéo với 10 box trên 1 phòng, đảm bảo sự riêng tư cho khách hàng 100%. Ở đây bạn có thể hẹn hò, học tập, nghỉ ngơi,...



Bao gồm 2 loại box: Vip và Thường. Với box Vip sẽ miễn phí netflix với không gian rộng rãi đủ cho cả 2 người



Box Thường sẽ thích hợp để nghỉ ngơi, chạy deadline. Nhưng vẫn đủ riêng tư cho các cặp đôi!!!



Bạn sẽ nhận được những gì khi đến với chúng tôi???



Liên hệ với chúng tôi ngayyy!

Một môi đã có Mochi!!!

📍 Mochi Cafe In Bed
☎ Phone: +001515151515
✉ Email: mail@mai.com

Name

Enter email

Comment

Send

Thông tin

Hiện tại tại mình chỉ có 1 chi nhánh duy nhất! Không nhận cọc! Nói không với cam! Đặt box trước trên web để nhận nhiều ưu đãi hơn nhé!

Copyright

Loại box

Box Vip

Box Thường

Thông tin nhanh

Gọi ngay tới: 0256458554

CHƯƠNG 8. Hạn chế & Hướng phát triển

8.1. Hạn chế hiện tại

- Website mới dừng ở mức Front-End, chưa có backend thực thụ.
- Chưa có cơ sở dữ liệu động — các phân sản phẩm, nội dung vẫn còn tĩnh.
- Một số phần animation còn cơ bản, chưa có hiệu ứng nâng cao.
- Thiếu trang quản trị (Admin).
- Form liên hệ chưa có chức năng gửi mail thực tế.

8.2. Hướng phát triển

- Tích hợp backend (PHP hoặc Node.js) + MySQL để tạo dữ liệu động.
- Xây dựng trang quản trị cho phép thêm/sửa/xóa sản phẩm, bài viết.
- Thêm hệ thống đăng nhập/đăng ký cho người dùng.
- Tích hợp API (Google Maps, Chatbot hỗ trợ khách hàng).
- Tối ưu SEO để website có thể hiển thị tốt hơn trên Google.
- Nâng cấp trải nghiệm người dùng với hiệu ứng nâng cao và chuyển cảnh mượt.

9. Quy trình thực hiện

Quá trình thực hiện đồ án được chia thành nhiều giai đoạn nhằm đảm bảo tiến độ và chất lượng sản phẩm. Cụ thể:

Giai đoạn	Thời gian	Công việc chính
1. Lên ý tưởng & chọn đề tài	Tuần 1	<ul style="list-style-type: none"> - Khảo sát các website cùng lĩnh vực để tham khảo giao diện và tính năng. - Lựa chọn đề tài phù hợp với kiến thức học phần Front-End. - Vẽ sơ đồ bố cục và phác thảo giao diện ban đầu. - Thiết kế khung HTML cơ bản cho các trang (index, about, contact...).
2. Thiết kế giao diện tĩnh	Tuần 2	<ul style="list-style-type: none"> - Đặt layout chung: header – main – footer. - Gắn các class Bootstrap để tạo lưới (grid).
3. Áp dụng CSS & Bootstrap	Tuần 3	<ul style="list-style-type: none"> - Tùy chỉnh màu sắc, font, khoảng cách. - Hoàn thiện giao diện trang chủ và các section chính.
4. Hoàn thiện responsive	Tuần 4	<ul style="list-style-type: none"> - Kiểm tra hiển thị trên nhiều thiết bị. - Chỉnh sửa CSS theo từng breakpoint.
5. Thêm JavaScript cơ bản	Tuần 5	<ul style="list-style-type: none"> - Carousel, menu di động, hiệu ứng cuộn. - Test toàn bộ trang.
6. Kiểm thử & hoàn thiện báo cáo	Tuần 6	<ul style="list-style-type: none"> - Ghi nhận lỗi, chỉnh sửa. - Viết báo cáo & chuẩn bị thuyết trình.

Việc chia nhỏ quy trình giúp người thực hiện quản lý thời gian hiệu quả, phát hiện lỗi sớm và

đảm bảo mỗi phần được xây dựng chắc chắn trước khi chuyển sang phần tiếp theo.

10. Kiến thức học được

Trong quá trình thực hiện đồ án, người thực hiện đã tích lũy được nhiều kiến thức và kỹ năng thực tế:

- **HTML5**
 - Cấu trúc semantic rõ ràng (header, main, footer, section...).
 - Cách tạo form, carousel, menu, nhúng hình ảnh và video.
- **CSS3 & Bootstrap 5**
 - Tạo layout responsive bằng hệ thống lưới (Grid System).
 - Sử dụng biến màu, hiệu ứng hover, animation cơ bản.
 - Tùy chỉnh Bootstrap để phù hợp với giao diện riêng.
- **JavaScript cơ bản**
 - Viết hàm để xử lý sự kiện người dùng.
 - Thêm tương tác như ẩn/hiện menu di động, hiệu ứng khi cuộn trang.
- **Kỹ năng thiết kế**
 - Phối màu, chọn font chữ hài hòa, khoảng cách hợp lý.
 - Biết cách trình bày nội dung theo chiều dọc, dễ đọc.
- **Quản lý file & dự án**
 - Biết cách tổ chức thư mục hợp lý.
 - Biết chạy dự án bằng VS Code và Live Server.

- Biết kiểm tra hiển thị trên nhiều thiết bị.
- **Kỹ năng viết báo cáo & trình bày**
 - Soạn thảo báo cáo theo bố cục chuẩn.
 - Trình bày nội dung ngắn gọn, rõ ràng, có dẫn chứng cụ thể.

11. Kiểm thử và đánh giá

11.1. Kiểm thử giao diện (UI Testing)

- Kiểm tra toàn bộ bố cục trên trình duyệt desktop: không bị vỡ layout, chữ không bị đè.
- Kiểm tra màu sắc, font chữ, khoảng cách có đồng bộ không.
- Kiểm tra carousel chạy đúng, không lỗi khi chuyển slide.
- Kiểm tra header khi cuộn trang: menu vẫn cố định ở trên.
- Đảm bảo hình ảnh không bị méo khi co kéo cửa sổ.

11.2. Kiểm thử chức năng (Functional Testing)

- Menu: khi nhấp vào các mục, chuyển đúng đến section/trang tương ứng.
- Nút “Liên hệ” → chuyển đến trang contact hoặc form liên hệ.
- Nút carousel hoạt động đúng.
- Form liên hệ có kiểm tra nhập liệu cơ bản (bắt buộc nhập tên, email, nội dung).

11.3. Kiểm thử trải nghiệm thực tế

- Kiểm tra hiển thị trên điện thoại thật (Android & iPhone).
- Thử bằng nhiều trình duyệt (Chrome, Edge, Firefox).

- Thử lướt trang từ đầu đến cuối → cảm giác mượt, không bị lag.
- Kiểm tra tốc độ tải trang → tải nhanh, không cần reload nhiều.

11.4. Đánh giá tổng thể

Website đáp ứng đầy đủ yêu cầu của một đồ án Front-End cơ bản:

- Giao diện rõ ràng, bắt mắt.
- Có đầy đủ các trang và mục cần thiết.
- Tương thích đa thiết bị.
- Có các hiệu ứng cơ bản tạo tính sinh động.
- Dễ cài đặt và kiểm thử.

12. Kế hoạch mở rộng trong tương lai

12.1. Mở rộng về kỹ thuật

- Kết nối cơ sở dữ liệu MySQL hoặc Firebase để nội dung động.
- Xây dựng trang quản trị với CRUD cơ bản (thêm/sửa/xóa sản phẩm, tin tức).
- Tích hợp API đăng nhập Google/Facebook.
- Áp dụng framework Front-End hiện đại như React hoặc Vue để tối ưu hiệu suất.
- Triển khai website lên hosting thực tế.

12.2. Mở rộng về nội dung

- Thêm nhiều chuyên mục, bài viết, sản phẩm hơn.
- Tích hợp blog hoặc khu vực chia sẻ kiến thức.

- Cập nhật nội dung thường xuyên để giữ website luôn mới.

12.3. Mở rộng về trải nghiệm người dùng

- Tối ưu tốc độ tải trang bằng nén ảnh, lazy loading.
- Thêm animation chuyển cảnh nâng cao (fade, slide, parallax).
- Gợi ý nội dung liên quan cho người dùng.
- Cải thiện UI/UX qua phản hồi thực tế.

12.4. Định hướng dài hạn

- Xây dựng thành một sản phẩm hoàn chỉnh, có backend quản lý.
- Đưa website hoạt động thật để phục vụ người dùng thực tế.
- Nâng cấp thành dự án nhóm, mở rộng thêm nhiều tính năng nâng cao.

13. Nhận xét & Kết luận tổng quát

13.1. Nhận xét cá nhân

Việc thực hiện đồ án Front-End đã giúp người học rèn luyện nhiều kỹ năng quan trọng, từ thiết kế giao diện đến kiểm thử và trình bày báo cáo. Quá trình làm tuy có nhiều khó khăn ban đầu (đặc biệt là responsive), nhưng dần dần đã thành thạo hơn trong việc sử dụng công cụ.

13.2. Đánh giá kết quả đạt được

- Hoàn thành đúng yêu cầu của môn học.
- Giao diện đẹp, bố cục hợp lý, có tính thực tế.
- Có thể mở rộng dễ dàng nếu có thời gian.

- Đáp ứng tiêu chí kỹ thuật, trình bày tốt.

13.3. Cảm nhận sau khi hoàn thành đồ án

Đây là một trải nghiệm quý giá, giúp người học thấy rõ sự khác biệt giữa kiến thức lý thuyết và ứng dụng thực tế. Qua đó, kỹ năng lập trình, thẩm mỹ và khả năng giải quyết vấn đề được nâng cao đáng kể.

13.4. Kết luận

Đồ án Front-End không chỉ là một bài tập lập trình, mà còn là bước khởi đầu cho quá trình xây dựng các sản phẩm web thực tế. Thông qua việc thiết kế, triển khai và kiểm thử, người học đã có được nền tảng vững chắc để tiếp tục học các công nghệ nâng cao hơn.

Link github: [nguyetgam/htng](https://github.com/nguyetgam/htng)

- `express-flash-messages`: (Thường được dùng với tên `connect-flash`) Là middleware cho phép ta lưu trữ các thông báo tạm thời (flash messages) và hiển thị chúng cho người dùng ở lần request tiếp theo (ví dụ: thông báo thành công/thất bại sau khi đăng ký/đăng nhập).
- `express-ejs-layouts`: Là một thư viện giúp quản lý layout (bố cục) cho các view sử dụng EJS trong Express. Nó cho phép ta định nghĩa layout chung và tái sử dụng cho nhiều trang khác nhau, giúp code gọn gàng và dễ bảo trì hơn.



Hình 1.2.2 ExpressJS

1.1.1. MongoDB

MongoDB là một hệ quản trị cơ sở dữ liệu (DBMS) NoSQL mã nguồn mở, hướng tài liệu (document-oriented). Thay vì lưu trữ dữ liệu trong các bảng với cấu trúc cố định như SQL, MongoDB lưu trữ dữ liệu trong các document có định dạng giống JSON (gọi là BSON).

❖ Tăng cường hiệu suất: Mongoose

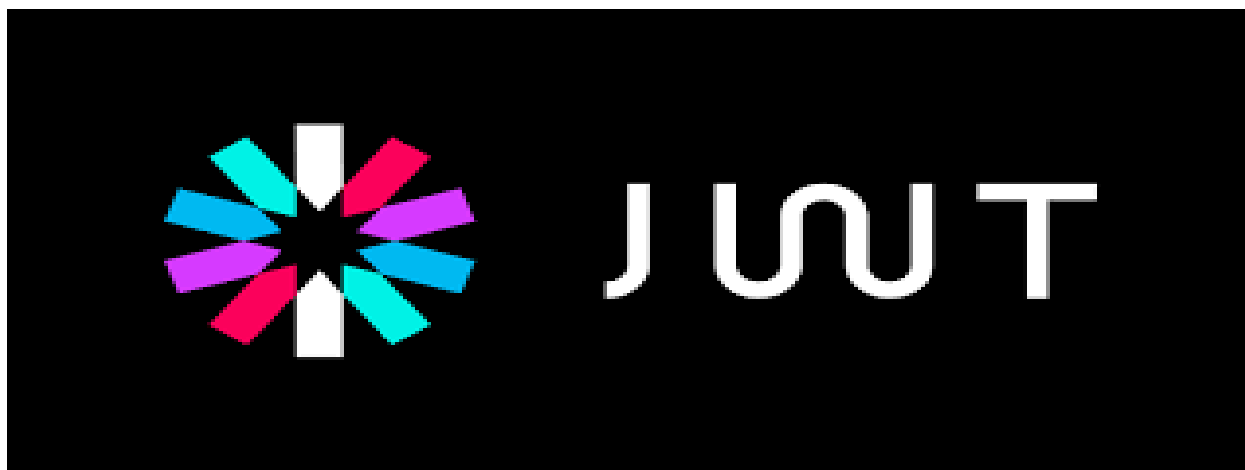
Mongoose là một thư viện Object Document Modeling (ODM) cho MongoDB và Node.js. Nó cung cấp một cách thức đơn giản và trực quan để tương tác với cơ sở dữ liệu MongoDB, cho phép ta định nghĩa các schema (cấu trúc dữ liệu), thực hiện các truy vấn, và quản lý các mối quan hệ giữa các document.



Hình 1.2.3 MongoDB

1.1.2. JWT

JSON Web Token (JWT) là một tiêu chuẩn mở (RFC 7519) định nghĩa một cách thức nhỏ gọn và độc lập để truyền thông tin giữa các bên dưới dạng đối tượng JSON. Thông tin này có thể được xác minh và tin cậy vì nó được ký điện tử. JWT thường được sử dụng để xác thực (authentication) và ủy quyền (authorization) trong các ứng dụng web.



Hình 1.2.4 JWT

1.1.3. Nodemailer

Nodemailer là một module cho Node.js cho phép ta gửi email một cách dễ dàng. Nó hỗ trợ nhiều phương thức gửi mail khác nhau (SMTP, Sendmail, ...), có thể đính kèm file, và sử dụng HTML trong nội dung email.

❖ Tăng cường hiệu suất:

nodemailer-express-handlebars là một plugin cho Nodemailer cho phép ta sử dụng template engine Handlebars (tương tự EJS) để tạo nội dung email. Điều này giúp ta dễ dàng tạo ra các email động với nội dung được tùy biến theo dữ liệu từ ứng dụng.

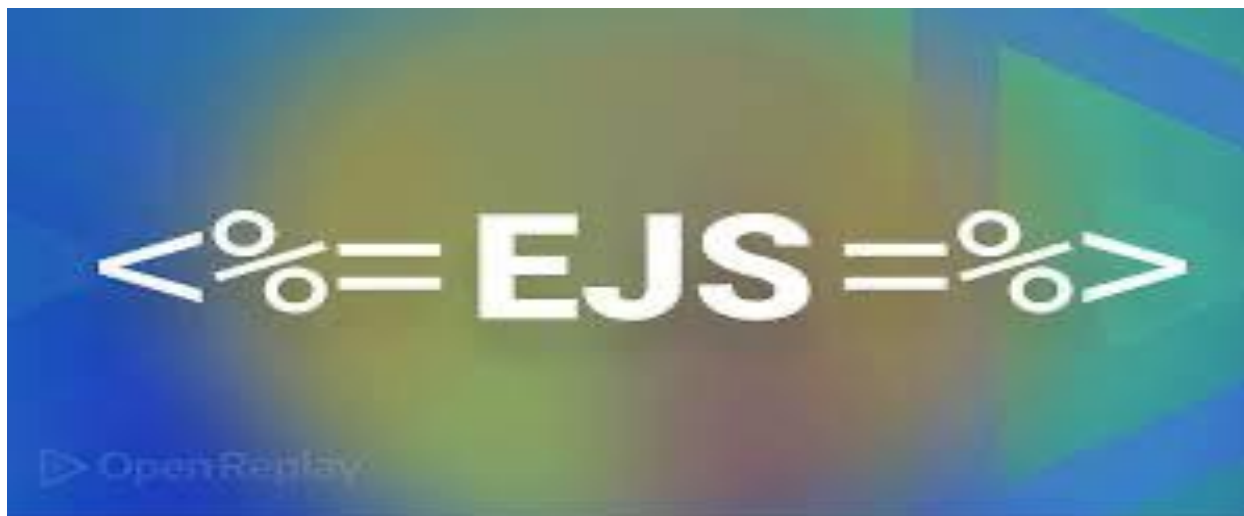


Nodemailer

Hình 1.2.5 Nodemailer

1.1.4. EJS

EJS (Embedded JavaScript templates) là một template engine đơn giản cho phép ta nhúng mã JavaScript vào trong các file HTML. Nó giúp ta dễ dàng tạo ra các trang web động bằng cách kết hợp dữ liệu từ server với các mẫu HTML.



Hình 1.2.6 EJS

1.1.5. Dotenv

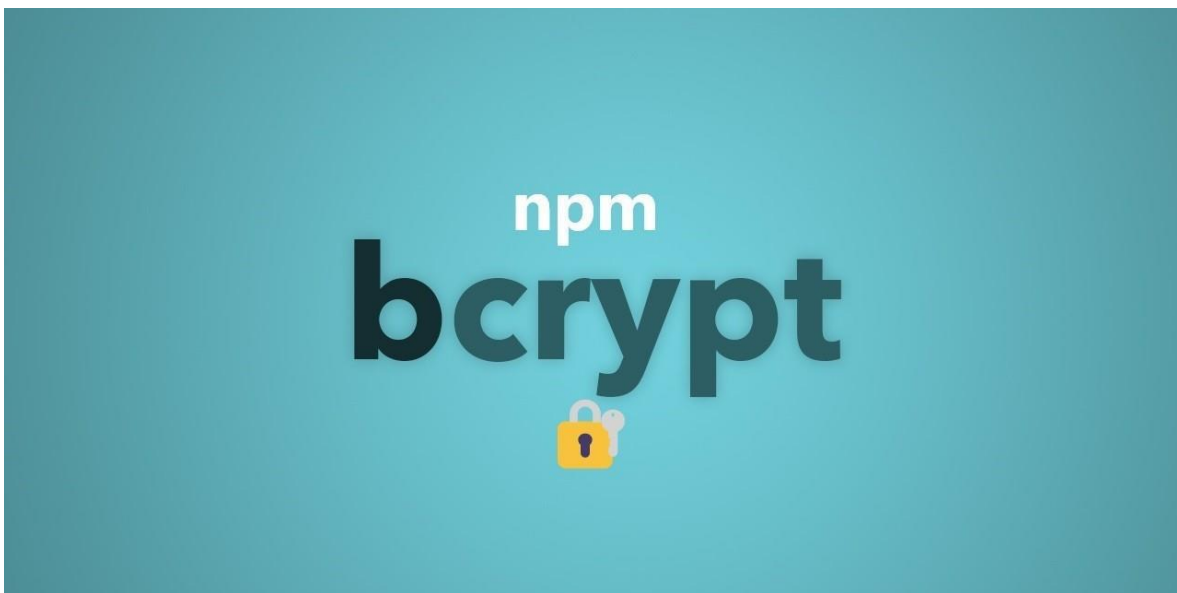
Dotenv là một module cho phép ta tải các biến môi trường (environment variables) từ một file .env vào trong ứng dụng Node.js. Việc này giúp ta quản lý các cấu hình (config) một cách dễ dàng và bảo mật hơn, đặc biệt là khi ta cần cấu hình khác nhau cho các môi trường phát triển, thử nghiệm, và production.



Hình 1.2.7 Dotenv

1.1.6. Bcrypt

Bcrypt là một hàm băm mật khẩu (password hashing function) được thiết kế để an toàn và chậm, làm cho việc tấn công brute-force (thử tất cả các mật khẩu có thể) trở nên khó khăn hơn. Nó thường được sử dụng để lưu trữ mật khẩu người dùng một cách an toàn trong cơ sở dữ liệu.



Hình 1.2.8 Bcrypt

1.2. Lý do chọn công nghệ

1.2.1. NodeJS

- JavaScript đồng nhất: Sử dụng JavaScript cho cả frontend và backend giúp đơn giản hóa việc phát triển và dễ dàng chia sẻ code giữa hai phần.
- Hiệu suất tốt: Node.js hoạt động nhanh và phù hợp với các ứng dụng cơ bản như API nhờ cơ chế xử lý bất đồng bộ (non-blocking).
- Hệ sinh thái lớn: Node.js có một hệ sinh thái npm (Node Package Manager) rất lớn với nhiều thư viện và công cụ hữu ích.

1.2.2. Express

- Đơn giản và linh hoạt: Express cung cấp một framework tối giản nhưng mạnh mẽ, cho phép ta dễ dàng xây dựng các API và ứng dụng web.
- Middleware: Hệ thống middleware của Express giúp ta dễ dàng mở rộng chức năng của ứng dụng.
- Phổ biến: Express là framework phổ biến nhất cho Node.js, có cộng đồng hỗ trợ lớn.

1.2.3. MongoDB

- Linh hoạt: Schema linh hoạt của MongoDB cho phép ta dễ dàng thay đổi cấu trúc dữ liệu khi cần thiết.
- Khả năng mở rộng: MongoDB được thiết kế để có thể mở rộng theo chiều ngang (horizontal scaling), phù hợp với các ứng dụng có lượng dữ liệu lớn.
- Hiệu suất tốt: MongoDB có hiệu suất đọc/ghi tốt, đặc biệt là với các ứng dụng có nhiều thao tác ghi.

1.2.4. JWT

- Xác thực và ủy quyền: Cung cấp một cách thức an toàn và hiệu quả để xác thực và ủy quyền người dùng.

1.2.5. Nodemailer

- Gửi email: Cho phép ứng dụng gửi email thông báo, xác nhận, v.v.

Chương 2. THIẾT KẾ KIẾN TRÚC HỆ THỐNG

2.1. Kiến trúc ứng dụng

Ứng dụng sẽ hoạt động dựa trên các thành phần chính sau:

2.1.1. Client (Frontend)

- Là nơi người dùng tương tác với hệ thống, gửi các yêu cầu (request) như đăng ký, đăng nhập, hoặc xem danh sách người dùng.
- Client giao tiếp với server thông qua API endpoints (ví dụ: `/api/login`, `/api/users`).
- Dữ liệu trả về từ server (response) thường ở định dạng JSON để dễ dàng xử lý trên client.

2.1.2. Server (Backend)

- Xử lý các yêu cầu từ phía client.
- Quản lý logic ứng dụng, xác thực người dùng, và kết nối với cơ sở dữ liệu.
- Phân tầng chính:
 - Routes: Xử lý các endpoint, định tuyến yêu cầu từ client đến controller phù hợp.
 - Controllers: Xử lý logic của ứng dụng, gọi dữ liệu từ Model, xử lý xong trả về response cho client.
 - Models: Quản lý dữ liệu, kết nối với cơ sở dữ liệu (MongoDB).

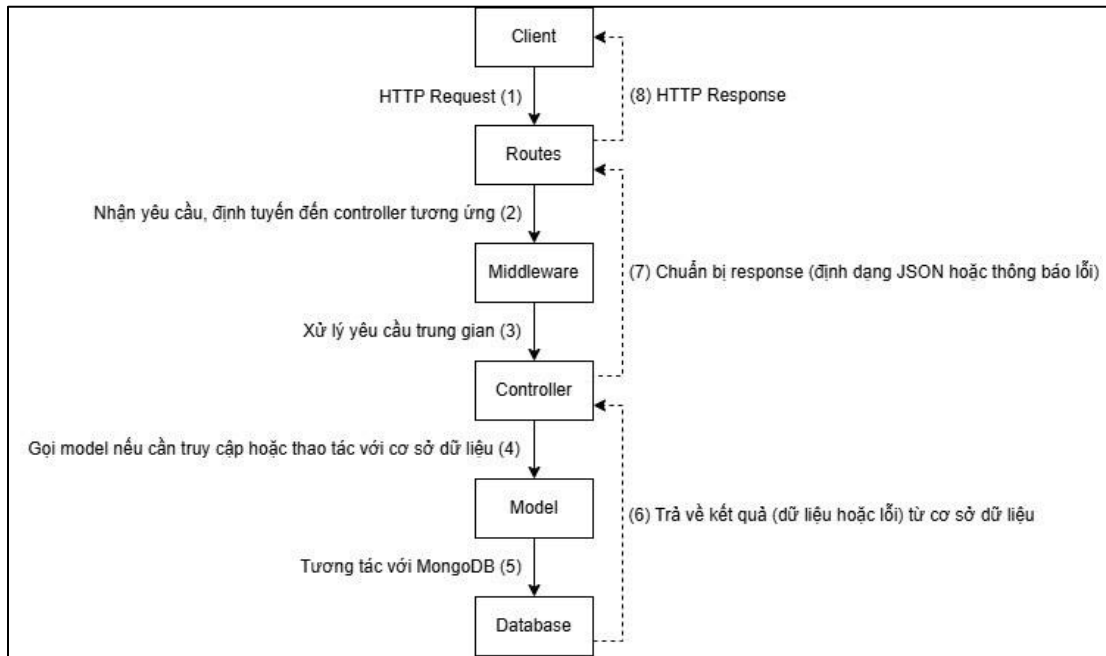
2.1.3. Database (MongoDB)

- Nơi lưu trữ dữ liệu người dùng, bao gồm thông tin đăng nhập, email, và các thông tin khác.
- Hỗ trợ các thao tác thêm, sửa, xóa và đọc dữ liệu (CRUD).

2.1.4. Middleware

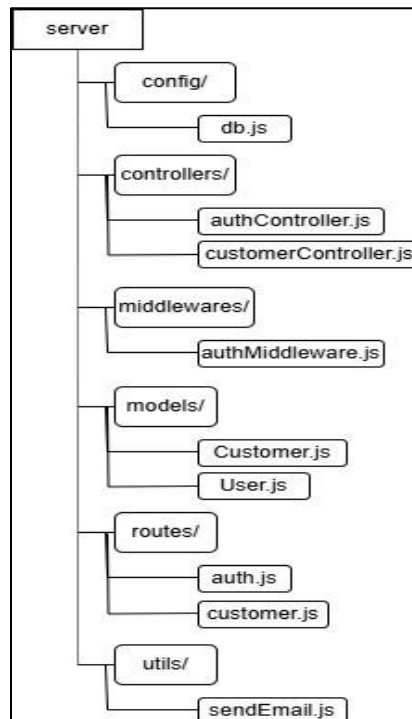
- Các lớp trung gian được sử dụng để xử lý yêu cầu trước khi chúng đến controller.
- Ví dụ: Middleware xác thực người dùng bằng JWT để bảo vệ các API nhạy cảm.

2.1.5. Sơ đồ kiến trúc ứng dụng



Hình 2.1.5 Sơ đồ kiến trúc ứng dụng

2.2. Cấu trúc thư mục



Hình 2.2 Cấu trúc thư mục server

2.2.1. config/

- Thư mục này chứa tệp *db.js* để cấu hình và thiết lập kết nối với cơ sở dữ liệu MongoDB.
- Vai trò chính: Quản lý kết nối cơ sở dữ liệu một cách tập trung và dễ bảo trì.

2.2.2. controllers/

- Chứa các tệp xử lý logic chính của ứng dụng.
- *authController.js*: Xử lý xác thực người dùng, bao gồm đăng nhập, đăng ký và cấp phát JWT token.
- *customerController.js*: Xử lý các thao tác liên quan đến khách hàng như thêm mới, sửa đổi, xóa hoặc lấy danh sách khách hàng.

2.2.3. middlewares/

- Thư mục này chứa các middleware – các lớp trung gian xử lý trước khi yêu cầu tới controller.
- *authMiddleware.js*: Xác thực token JWT để bảo vệ các route yêu cầu quyền truy cập.

2.2.4. models/

- Chứa các schema của MongoDB, định nghĩa cấu trúc dữ liệu lưu trong cơ sở dữ liệu.
- *Customer.js*: Định nghĩa schema và model của khách hàng (Customer).
- *User.js*: Định nghĩa schema và model của người dùng (User).

2.2.5. routes/

- Định nghĩa các endpoint API và liên kết với controller tương ứng.
- *auth.js*: Chứa các endpoint liên quan đến xác thực (đăng nhập, đăng ký).
- *customer.js*: Chứa các endpoint quản lý khách hàng (CRUD khách hàng).

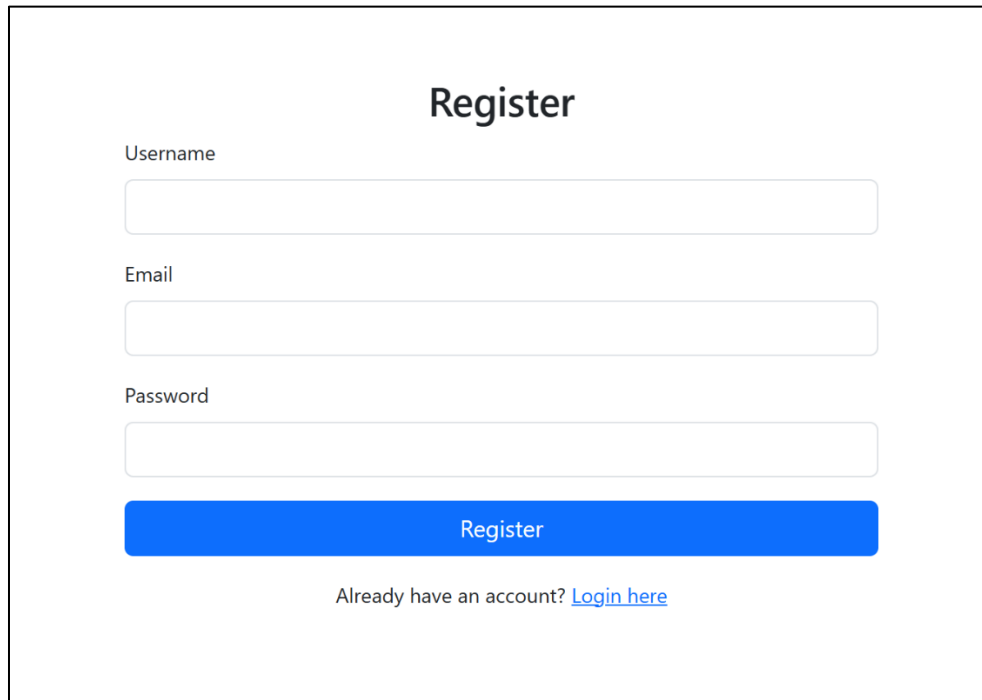
2.2.6. utils/

- Chứa các tiện ích dùng chung trong ứng dụng.
- *sendEmail.js*: Thư viện gửi email (sử dụng Nodemailer), thường dùng cho xác thực tài khoản hoặc gửi thông báo.

Chương 3. CÀI ĐẶT THỰC NGHIỆM VÀ KẾT QUẢ

3.1. Một số giao diện website

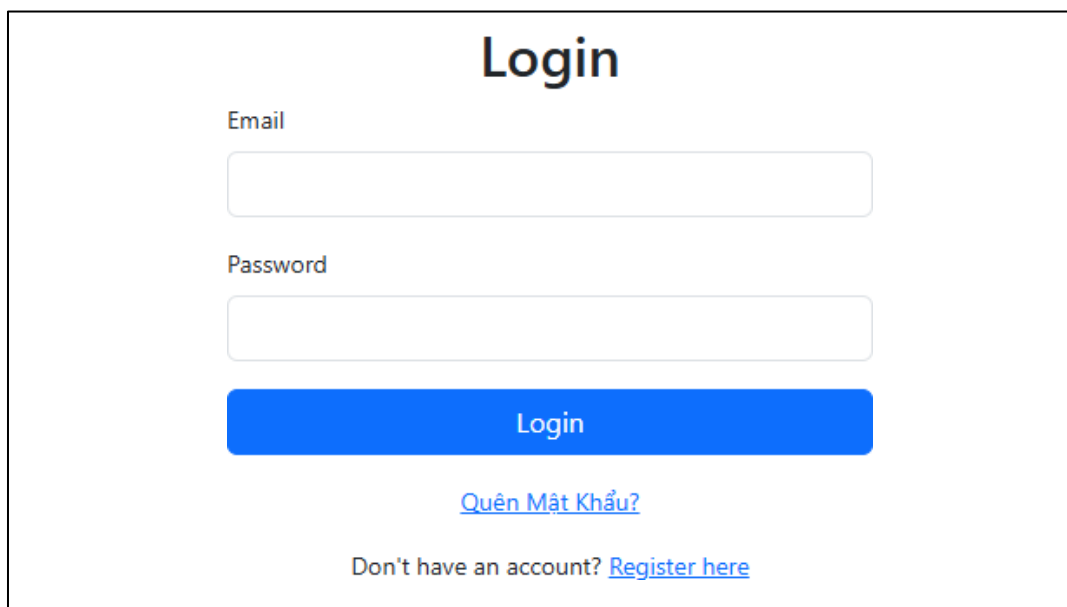
3.1.1. Giao diện đăng ký



The image shows a web registration form titled "Register". It contains three input fields: "Username", "Email", and "Password". Below these fields is a blue button labeled "Register". At the bottom, there is a link that says "Already have an account? [Login here](#)".

Hình 3.1.1 Giao diện đăng ký

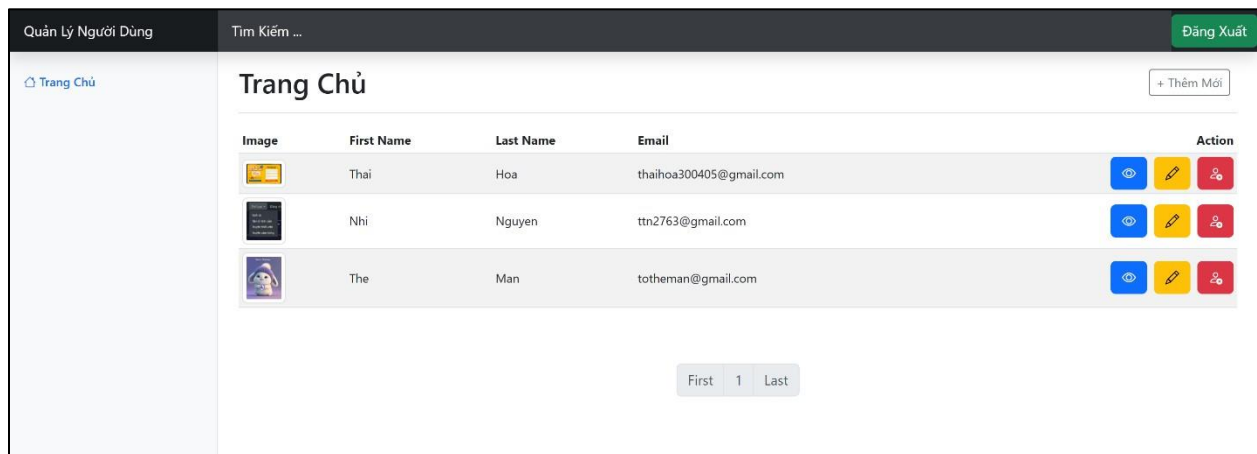
3.1.2. Giao diện đăng nhập



The image shows a web login form titled "Login". It contains two input fields: "Email" and "Password". Below these fields is a blue button labeled "Login". Below the button is a link that says "[Quên Mật Khẩu?](#)". At the bottom, there is a link that says "Don't have an account? [Register here](#)".

Hình 3.1.2 Giao diện đăng nhập

3.1.3. Giao diện danh sách người dùng



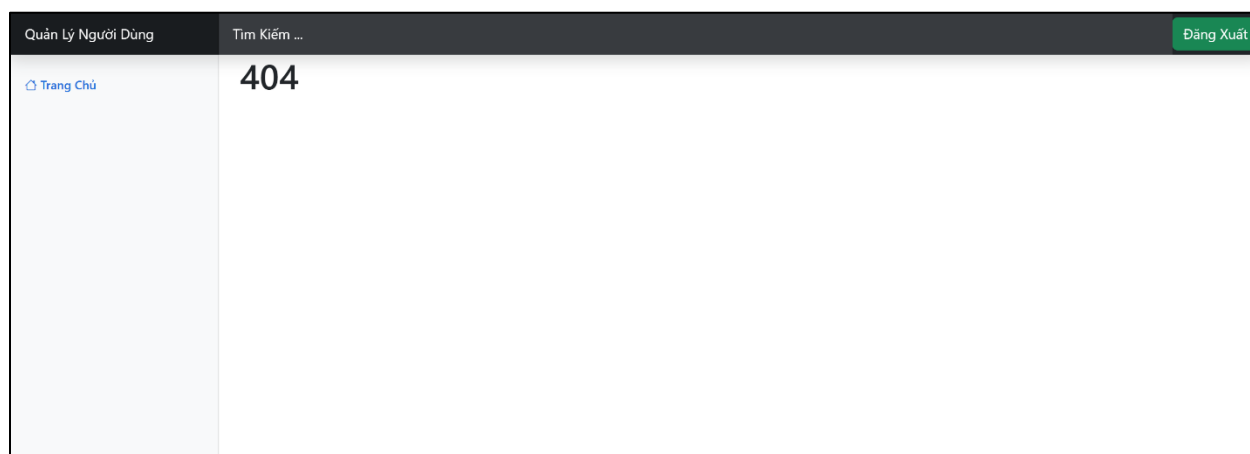
Hình 3.1.3 Giao diện danh sách người dùng

3.1.4. Giao diện thêm mới người dùng

The screenshot displays the 'Thêm Người Dùng' (Add New User) form. The top navigation bar is the same as in the previous image. The left sidebar shows 'Trang Chủ'. The main content area is titled 'Thêm Người Dùng' and includes a breadcrumb 'Trang Chủ / Thêm Người Dùng' and a 'Mã Người Dùng' (User Code) field. The form contains several input fields: 'First Name' (with placeholder 'First Name'), 'Last Name' (with placeholder 'Last Name'), 'Telephone' (with placeholder 'Telephone'), and 'Email' (with placeholder 'Email'). There is a 'Customer Details' section with a text area for 'Customer Details'. At the bottom, there is a 'Hình Ảnh' (Image) section with a 'Chọn tệp' (Choose file) button and a message 'Không có tệp nào được chọn' (No file selected). A blue 'Add Customer' button is at the bottom left.

Hình 3.1.4 Giao diện thêm mới người dùng

3.1.5. Giao diện lỗi 404



Hình 3.1.5 Giao diện lỗi 404

3.2. Các chức năng chính

3.2.1. Chức năng đăng ký và đăng nhập

3.2.1.1. Đăng ký tài khoản

❖ Mục tiêu:

- Thu thập thông tin từ người dùng bao gồm: username, email, password.
- Mã hóa mật khẩu trước khi lưu vào cơ sở dữ liệu.
- Gửi email xác thực để kích hoạt tài khoản.

❖ Luồng hoạt động:

1. Người dùng nhập thông tin cần thiết (username, email, password).
2. Hệ thống kiểm tra:
 - Email đã tồn tại hay chưa.
 - Email có hợp lệ hay không.
3. Nếu thông tin hợp lệ:
 - Lưu người dùng vào cơ sở dữ liệu với trạng thái chưa được kích hoạt.
 - Tạo mã thông báo (token) để xác thực tài khoản qua email.
 - Gửi email xác thực.
4. Người dùng nhận email và nhấp vào liên kết để kích hoạt tài khoản

The screenshot shows a registration form with the title "Register" in a large, bold, black font. Below the title are three input fields: "Username" containing "TTM1234", "Email" containing "totheman2@gmail.com", and "Password" containing three dots. A blue "Register" button is positioned below the password field, with a green hand cursor icon pointing at it. At the bottom, there is a link that says "Already have an account? [Login here](#)".

Hình 3.2.1.1a Đăng ký tài khoản

The screenshot shows the same registration form as Figure 3.2.1.1a, but with a success message displayed in a light green box at the top: "Registration successful! Please check your email to verify your account." The input fields for "Username", "Email", and "Password" are now empty. The blue "Register" button remains at the bottom, and the link "Already have an account? [Login here](#)" is still present at the very bottom.

Hình 3.2.1.1b Thông báo đăng ký thành công

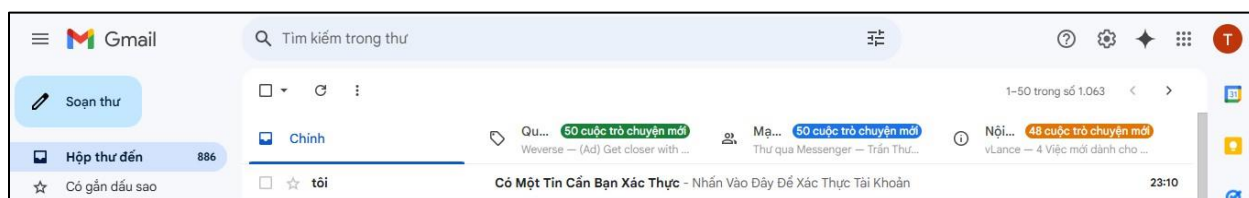
3.2.1.2. Xác thực email

❖ Mục tiêu:

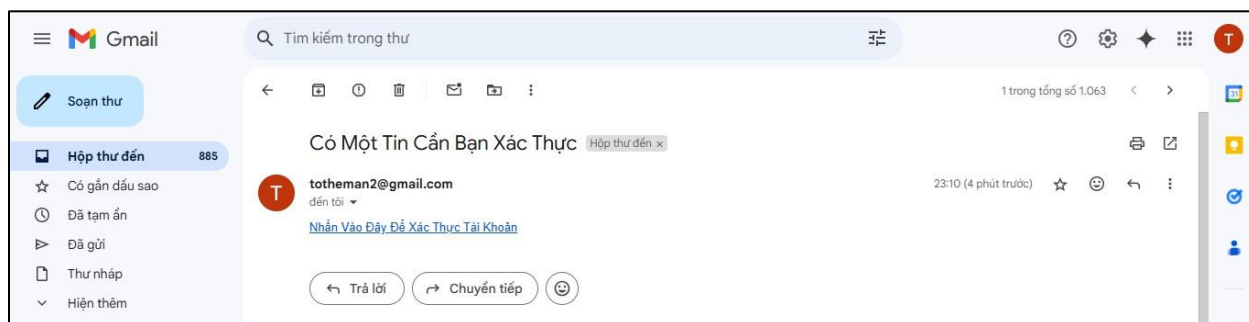
- Xác thực tài khoản người dùng khi nhấp vào liên kết trong email.

❖ Luồng hoạt động:

1. Người dùng nhấp vào liên kết xác thực trong email.
2. Hệ thống kiểm tra token hợp lệ hay không.
3. Nếu token hợp lệ:
 - Cập nhật trạng thái tài khoản sang isVerified: true.
 - Cho phép người dùng đăng nhập.
4. Nếu token không hợp lệ, thông báo lỗi.



Hình 3.2.1.2a Email xác thực gửi tới



Hình 3.2.1.2b Đường dẫn để xác thực tài khoản

3.2.1.3. Đăng nhập

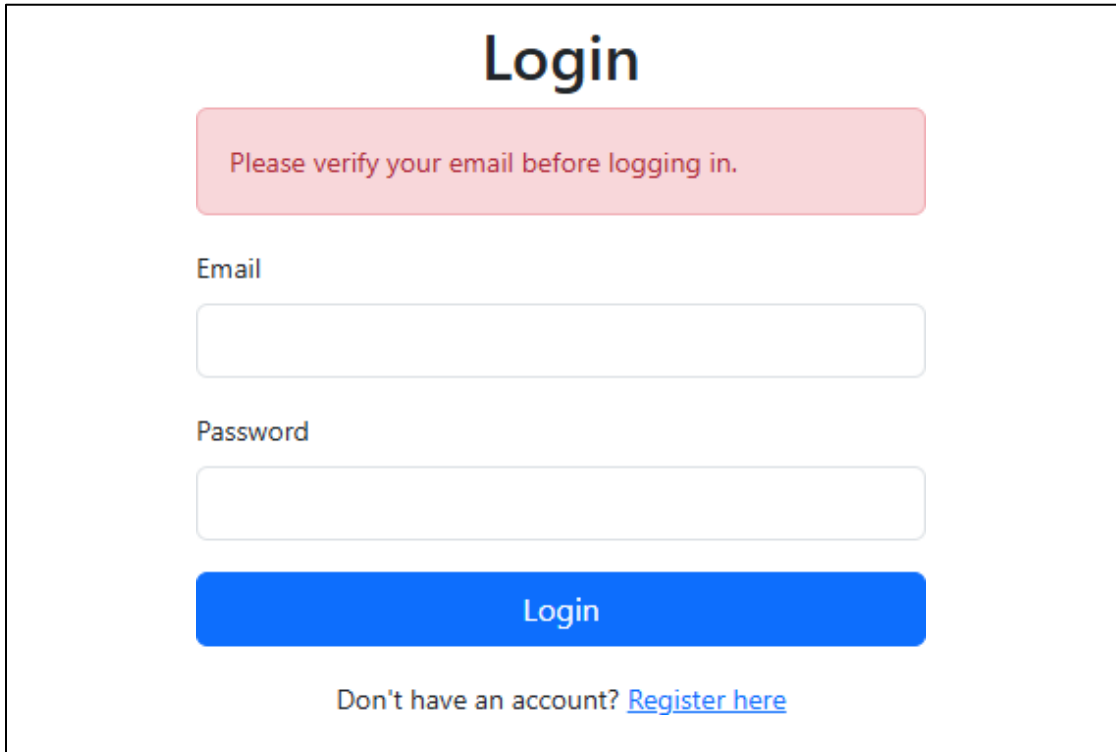
❖ Mục tiêu:

- Cho phép người dùng đăng nhập nếu tài khoản đã được xác thực.
- Tạo JSON Web Token (JWT) để duy trì phiên đăng nhập.

❖ Luồng hoạt động:

1. Người dùng nhập email và mật khẩu.
2. Hệ thống kiểm tra:
 - Email có tồn tại không.

- Tài khoản đã được xác thực chưa (isVerified: true).
 - Mật khẩu có hợp lệ không.
3. Nếu hợp lệ:
- Tạo và trả về token JWT.
4. Nếu không hợp lệ, trả về lỗi tương ứng.

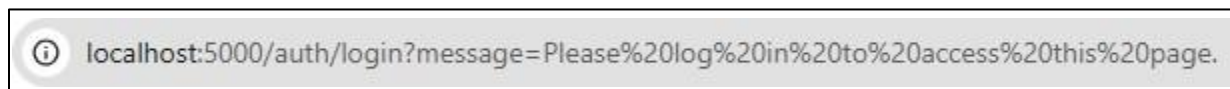


The screenshot shows a login interface with the title "Login" in a large, bold, dark blue font. Below the title is a red rectangular box containing the text "Please verify your email before logging in." in a red font. Underneath this box are two input fields: "Email" and "Password", both with light blue borders. Below the input fields is a solid blue button with the text "Login" in white. At the bottom of the form, there is a link that says "Don't have an account? [Register here](#)".

Hình 3.2.1.3 Khi chưa xác thực tài khoản

3.2.1.4. Kiểm tra trạng thái đăng nhập

- ❖ Mục tiêu:
 - Xác định người dùng đã đăng nhập chưa trước khi truy cập các route yêu cầu quyền.
 - Bảo vệ các tài nguyên quan trọng, chỉ cho phép người dùng được xác thực mới có quyền truy cập.
 - Tăng cường bảo mật, tránh các truy cập trái phép hoặc không hợp lệ.



Hình 3.2.1.4 Khi chưa đăng nhập

3.2.2. Chức năng quên mật khẩu

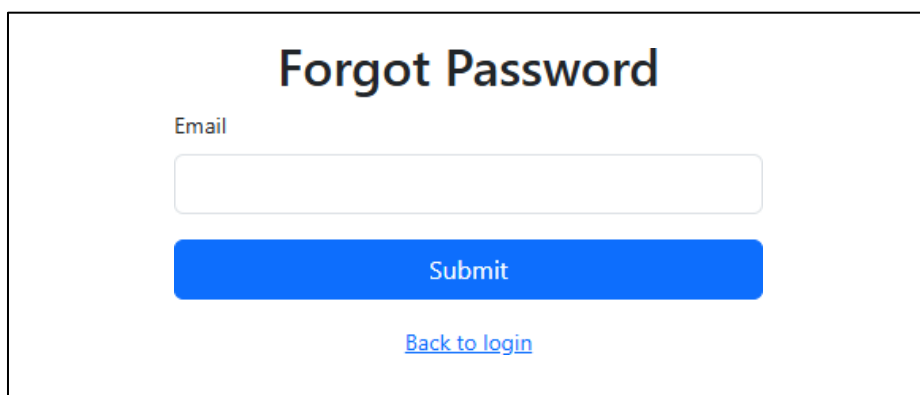
❖ Mục tiêu:

- Hỗ trợ người dùng đặt lại mật khẩu khi quên mật khẩu đăng nhập.
- Gửi email chứa liên kết đặt lại mật khẩu để xác minh.
- Cập nhật mật khẩu mới một cách an toàn và bảo mật.

❖ Luồng hoạt động:

1. Người dùng yêu cầu đặt lại mật khẩu:

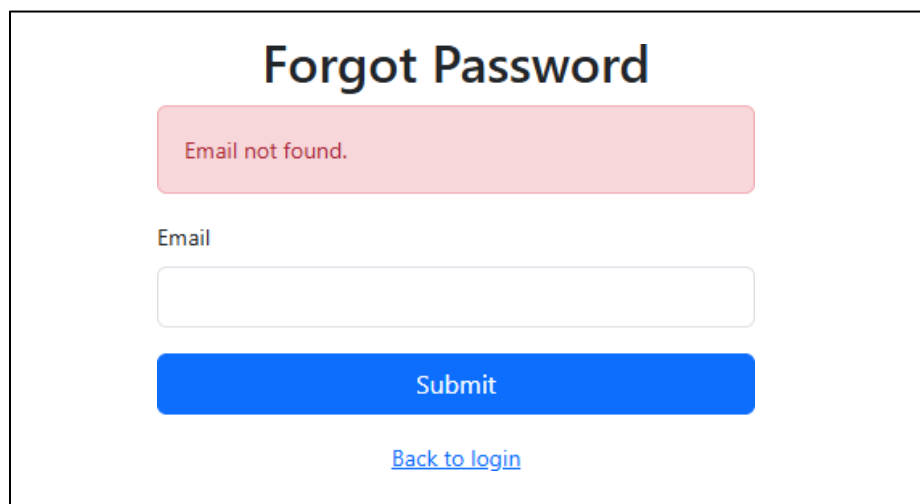
- Trên trang đăng nhập, người dùng chọn "Quên Mật Khẩu?".
- Hệ thống hiển thị trang nhập email.

The screenshot shows a web form titled "Forgot Password" in a large, bold, black font. Below the title, the word "Email" is displayed in a smaller font. Underneath is a white rectangular input field. Below the input field is a solid blue button with the word "Submit" in white text. At the bottom of the form, there is a blue hyperlink that reads "Back to login".

Hình 3.2.2a Giao diện trang quên mật khẩu

2. Hệ thống kiểm tra email:

- Kiểm tra email có tồn tại trong cơ sở dữ liệu không.
- Nếu không tồn tại: Thông báo lỗi.

This screenshot shows the same "Forgot Password" form as in the previous image, but with an error message. A light red rectangular box with rounded corners is positioned above the email input field, containing the text "Email not found." in a red font. The "Email" label, input field, "Submit" button, and "Back to login" link remain in their original positions and colors.

Hình 3.2.2b Thông báo lỗi khi email không tồn tại

- Nếu tồn tại: Tạo token bảo mật (resetPasswordToken) kèm thời gian hết hạn.

Hình 3.2.2c Thông báo khi email tồn tại

3. Gửi email xác nhận:

- Gửi email chứa liên kết đặt lại mật khẩu



Hình 3.2.2d Email đặt lại mật khẩu

4. Người dùng nhập vào liên kết:

- Hệ thống xác minh token:
 - Nếu hợp lệ: Hiện thị form nhập mật khẩu mới.

Hình 3.2.2e Giao diện trang đặt lại mật khẩu

- Nếu không hợp lệ hoặc token hết hạn: Thông báo lỗi.

5. Đặt lại mật khẩu:

- Người dùng nhập mật khẩu mới.
- Hệ thống mã hóa mật khẩu và lưu vào cơ sở dữ liệu.
- Xóa token đặt lại mật khẩu.

3.2.3. Chức năng thêm mới người dùng

Quản Lý Người Dùng

Tìm Kiếm ...

Đăng Xuất

Trang Chủ

Thêm Người Dùng

Trang Chủ / Thêm Người Dùng

Mã Người Dùng

First Name: Kieu

Last Name: Anh

Telephone: 09818116841

Email: kieuanh@gmail.com

Customer Details

Cao đẳng Công nghệ Thông tin TPHCM

Hình Ảnh

Chọn tệp: anh-dep-44.jpg.webp

Add Customer

Hình 3.2.2a Nhập thông tin người dùng mới

Quản Lý Người Dùng

Tìm Kiếm ...

Đăng Xuất

Trang Chủ

+ Thêm Mới

User added successfully!

Image	First Name	Last Name	Email	Action
	Kieu	Anh	kieuanh@gmail.com	
	Thai	Hoa	thaihoa300405@gmail.com	
	Nhi	Nguyen	ttn2763@gmail.com	
	The	Man	totheman@gmail.com	

First 1 Last

Hình 3.2.2b Thông báo thêm người dùng thành công

3.2.4. Chức năng xem chi tiết người dùng

Quản Lý Người Dùng

Tim Kiếm ...

Đăng Xuất

Trang Chủ


The Man

ShareExport

Trang Chủ / The

Lần Cuối Cập Nhật: Thu, 23 Jan 2025 16:39:53 GMT Mã Người Dùng: 678736aa3e74e78311a42fbc

Image:



Name:

The Man

Tel:

079138108

Email:

totheman@gmail.com

Details:

Tôi là Mẫn

Date Created:

2025-01-15 11:16:38

Date Modified:

2025-01-23 23:39:53

Hình 3.2.3 Giao diện xem chi tiết người dùng

3.2.5. Chức năng sửa thông tin người dùng

Quản Lý Người Dùng

Tim Kiếm ...

Đăng Xuất

Trang Chủ

Editing: The Man

?

Trang Chủ / The Man

Lần Cuối Cập Nhật: Thu, 23 Jan 2025 16:39:53 GMT Mã Người Dùng: 678736aa3e74e78311a42fbc

First Name

The

Last Name

Man

Telephone

079138108

Email


totheman@gmail.com

Customer Details

Tôi là Mẫn

Cập Nhật Hình Ảnh

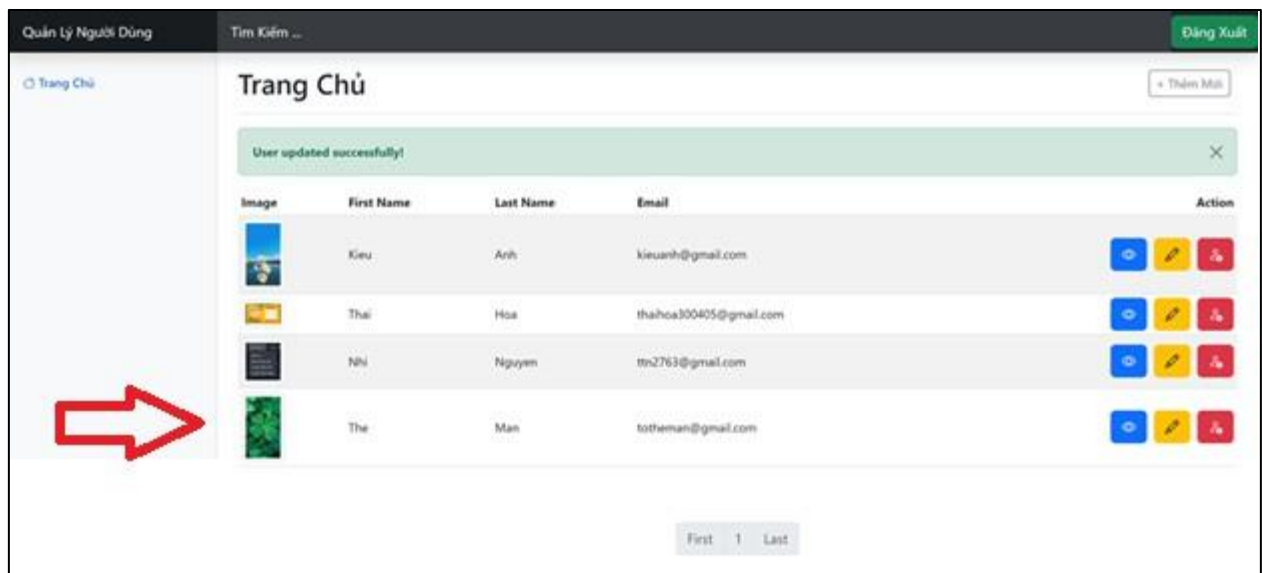
Chọn tệp hình-anh-may-man-trong-thi-cu-1.jpg



Update Customer

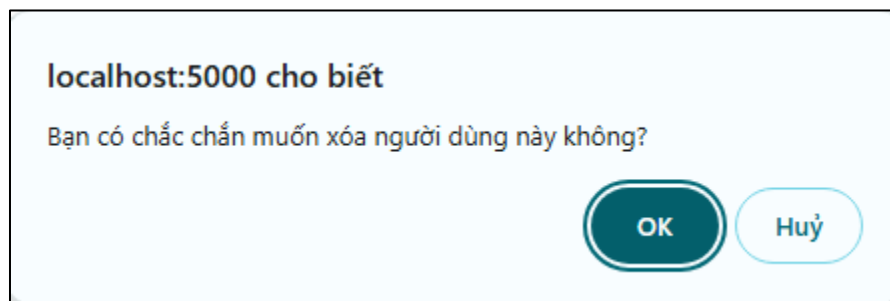
Delete Customer

Hình 3.2.4a Cập nhật hình ảnh người dùng



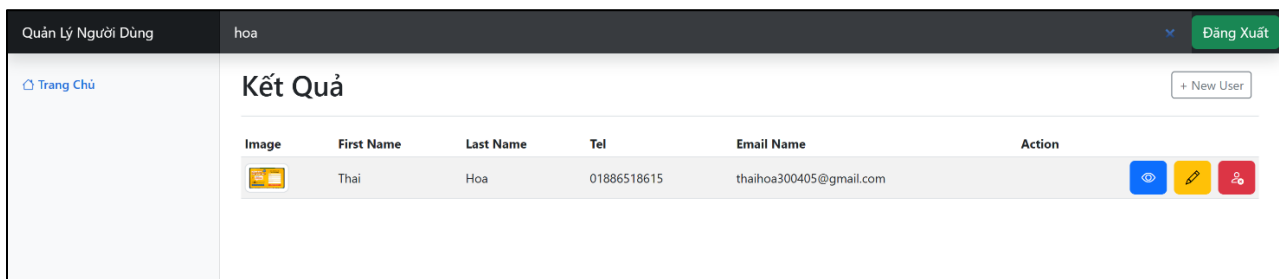
Hình 3.2.4b Thông báo thay đổi thành công

3.2.6. Chức năng xóa người dùng



Hình 3.2.5 Thông báo khi muốn xóa người dùng

3.2.7. Chức năng tìm kiếm



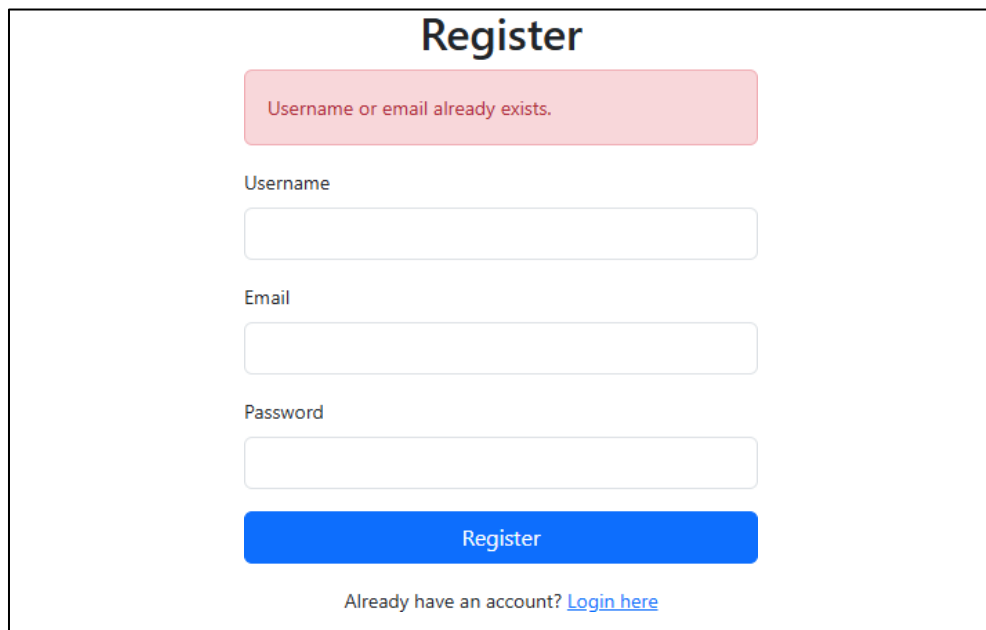
Hình 3.2.6 Tìm kiếm người dùng theo tên

Chương 4. XỬ LÝ LỖI VÀ BẢO MẬT

4.1. Xử lý lỗi trong ứng dụng

4.1.1. Lỗi 400 – Bad Request

- ❖ Nguyên nhân:
 - Dữ liệu đầu vào từ client không hợp lệ (ví dụ: thiếu email, password không đúng định dạng).
 - Thông tin tài khoản đã tồn tại (username hoặc email trùng).
- ❖ Cách xử lý:
 - Phản hồi lỗi với HTTP status code 400 kèm thông báo chi tiết: *"Username or email already exists."*



The image shows a web form titled "Register". At the top, there is a red error message box that says "Username or email already exists.". Below this, there are three input fields labeled "Username", "Email", and "Password". At the bottom of the form is a blue button labeled "Register". Below the button, there is a link that says "Already have an account? [Login here](#)".

Hình 4.1.1 Thông báo lỗi 400

4.1.2. Lỗi 401 – Unauthorized

- ❖ Nguyên nhân:
 - Người dùng không xác thực (chưa đăng nhập hoặc token không hợp lệ).
 - Người dùng chưa xác minh email trước khi đăng nhập.
- ❖ Cách xử lý:
 - Xác thực thông qua JWT hoặc kiểm tra isVerified trong cơ sở dữ liệu.

- Phản hồi với HTTP status code 401 kèm thông báo chi tiết:
 - *"Invalid email or password."*
 - *"Please verify your email before logging in."*

Login	Login
Invalid email or password.	Please verify your email before logging in.
Email <input type="text"/>	Email <input type="text"/>
Password <input type="password"/>	Password <input type="password"/>
Login	Login
Don't have an account? Register here	Don't have an account? Register here

Hình 4.1.2 Thông báo lỗi 401

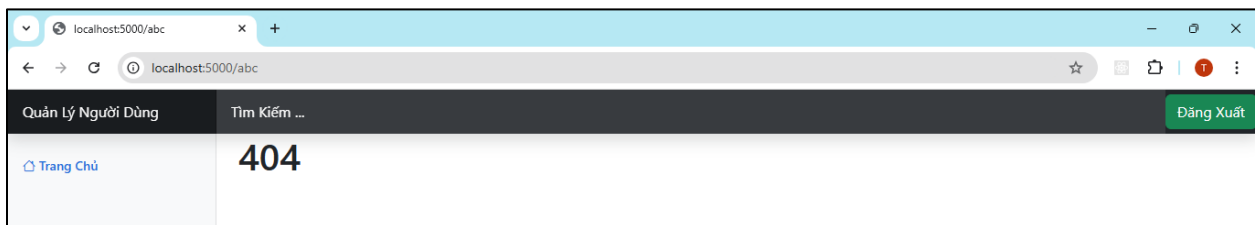
4.1.3. Lỗi 404 – Not Found

❖ Nguyên nhân:

- Không tìm thấy người dùng dựa trên token hoặc ID trong cơ sở dữ liệu.
- Truy cập một route không tồn tại.

❖ Cách xử lý:

- Kiểm tra sự tồn tại của tài nguyên trước khi thực hiện logic.
- Phản hồi với HTTP status code 404 và thông báo như: *"User not found. Please register again."*



Login

User not found. Please register again.

Email

Password

Login

Don't have an account? [Register here](#)

Hình 4.1.3 Thông báo lỗi 404

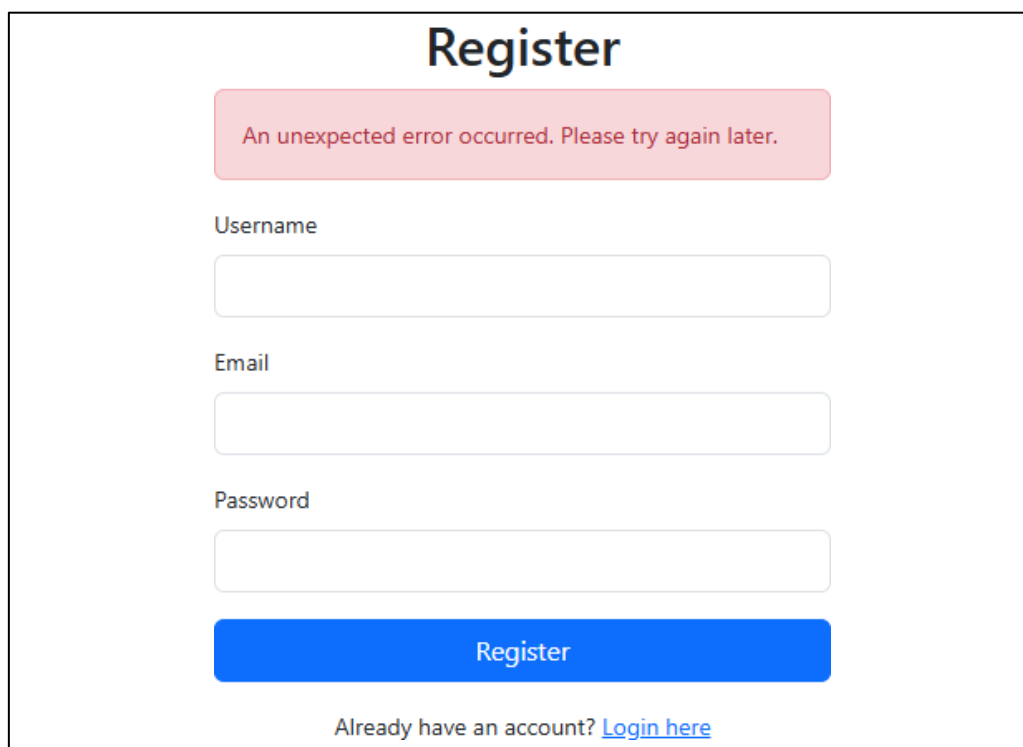
4.1.4. Lỗi 500 – Internal Server Error

❖ Nguyên nhân:

- Lỗi logic trong ứng dụng (ví dụ: lưu người dùng thất bại).
- Lỗi không mong đợi khi gửi email hoặc xử lý token.

❖ Cách xử lý:

- Sử dụng khối try-catch để bắt lỗi và ngăn ứng dụng bị crash.
- Ghi log chi tiết để phục vụ việc gỡ lỗi.
- Phản hồi với HTTP status code 500 và thông báo thân thiện: *"An unexpected error occurred. Please try again later."*



Register

An unexpected error occurred. Please try again later.

Username

Email

Password

Register

Already have an account? [Login here](#)

Hình 4.1.4 Thông báo lỗi 500

4.2. Bảo mật ứng dụng

4.2.1. Mã hóa mật khẩu

- Mục đích: Bảo mật thông tin nhạy cảm như mật khẩu để tránh trường hợp bị đánh cắp hoặc lộ lọt.
- Phương pháp:
 - Sử dụng thư viện bcryptjs để mã hóa mật khẩu trước khi lưu vào cơ sở dữ liệu.
 - Khi người dùng đăng nhập, mật khẩu được nhập sẽ được so sánh với mật khẩu đã được mã hóa trong cơ sở dữ liệu thay vì lưu trữ mật khẩu dạng thuần văn bản.
- Lợi ích: Ngay cả khi dữ liệu cơ sở dữ liệu bị rò rỉ, kẻ tấn công không thể sử dụng trực tiếp các mật khẩu được mã hóa.

4.2.2. Bảo vệ route

- Mục đích: Ngăn người dùng không xác thực hoặc không có quyền truy cập vào các route bảo mật.
- Phương pháp:

- Sử dụng JWT (JSON Web Token) để xác thực danh tính người dùng.
 - Mỗi khi người dùng đăng nhập thành công, server tạo một JWT chứa thông tin người dùng và gửi lại phía client.
 - Middleware được sử dụng để kiểm tra JWT trong mỗi yêu cầu tới các route bảo mật. Nếu JWT không hợp lệ, yêu cầu sẽ bị từ chối.
- Lợi ích:
- Cung cấp cách xác thực hiệu quả mà không cần lưu trữ trạng thái trên server (stateless authentication).
 - Giảm nguy cơ các yêu cầu không hợp lệ truy cập vào hệ thống.

Chương 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1. Kết luận

❖ Kết luận đạt được

- Đã xây dựng hoàn thiện Website quản lý người dùng với đầy đủ các tính năng cơ bản.
- Đáp ứng được các yêu cầu cơ bản của một hệ thống quản lý người dùng.
- Hệ thống API được thiết kế rõ ràng, dễ sử dụng và dễ mở rộng trong tương lai.
- Đã ứng dụng thành công các công nghệ hiện đại như Node.js, Express.js, MongoDB.
- Gửi email tự động qua Nodemailer, hỗ trợ xác thực và thông báo đến người dùng.
- Nhóm đã vận dụng và củng cố các kiến thức đã học, đồng thời rèn luyện kỹ năng phát triển dự án thực tế.

❖ Lý thuyết, kỹ năng và kinh nghiệm

- Biết cách áp dụng các công nghệ backend và sử dụng thành thạo Visual Studio Code để lập trình.
- Hiểu rõ quy trình xây dựng một hệ thống backend quản lý người dùng, từ thiết kế kiến trúc đến triển khai các chức năng.
- Nâng cao kiến thức về các công nghệ như Node.js, Express.js, JWT, MongoDB.
- Hiểu thêm việc sử dụng các thư viện và công cụ quan trọng như:
 - Mongoose: Quản lý cơ sở dữ liệu MongoDB.
 - JWT Authentication: Xác thực và phân quyền bảo mật.
 - Nodemailer: Gửi email tự động.
- Hiểu rõ cách tổ chức cấu trúc thư mục và tái sử dụng code trong dự án.

❖ Khuyết điểm

- Một số chức năng còn chưa tối ưu hiệu suất, tốc độ xử lý vẫn còn chậm với dữ liệu lớn.
- Chức năng quản lý chưa đa dạng, chưa có phân quyền người dùng chi tiết.
- Chưa triển khai website lên môi trường thực tế trên Internet, hiện chỉ dừng ở môi trường phát triển cục bộ.
- Cơ sở dữ liệu và code còn cần cải thiện để đạt hiệu quả tốt hơn.

- Độ bảo mật của hệ thống chưa cao, chưa triển khai đầy đủ các biện pháp phòng chống tấn công.

❖ Kinh nghiệm đúc kết

- Cần tăng cường khả năng tự học và nắm bắt các công nghệ mới trong lĩnh vực phát triển web.
- Tăng cường tư duy logic và kỹ năng giải quyết vấn đề.
- Cải thiện kiến thức chuyên môn về lập trình backend và quản lý cơ sở dữ liệu.
- Rèn luyện kỹ năng làm việc nhóm và chia sẻ kiến thức trong nhóm để nâng cao hiệu quả làm việc.

5.2. Hướng phát triển tương lai

❖ Phát triển đề tài

- Thêm chức năng phân quyền người dùng
- Mở rộng chức năng quản lý: Thêm tính năng quản lý chi tiết hơn như nhóm người dùng, lịch sử hoạt động.
- Tích hợp API bên thứ ba: Sử dụng các API để hỗ trợ các quy trình nâng cao như đồng bộ hóa dữ liệu hoặc tích hợp dịch vụ thanh toán.
- Tích hợp Mail Server: Xây dựng tính năng gửi email tự động như thông báo quan trọng, xác nhận tài khoản hoặc đặt lại mật khẩu.

❖ Phát triển bản thân

- Tăng cường khả năng tự học và nắm bắt các công nghệ mới.
- Phát triển khả năng tư duy logic và giải quyết các vấn đề phức tạp trong dự án.
- Cải thiện kiến thức chuyên môn về bảo mật hệ thống và tối ưu hóa hiệu suất.
- Tham gia các dự án thực tế để rèn luyện kỹ năng làm việc nhóm và quản lý dự án hiệu quả.

DANH MỤC TÀI LIỆU THAM KHẢO

- [1] <https://github.com/nodejs>
- [2] <https://github.com/itssubhodiproj/User-Management-System>
- [3] <https://github.com/DevikaRajesh22/UserManagement>
- [4] <https://nodejs.org/en>
- [5] <https://select2.org/>
- [6] <https://stackoverflow.com/>
- [7] <https://vi.wikipedia.org/>
- [8] <https://www.w3schools.com/>