

**KHOA KỸ THUẬT VÀ CÔNG NGHỆ**  
**BỘ MÔN CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN CHUYÊN NGÀNH**  
**HỌC KỲ I, NĂM HỌC 2024 – 2025**

# **XÂY DỰNG WEBSITE**

## **TÌM ĐỒ THẮT LẠC**

*Giáo viên hướng dẫn:*  
**TS. Nguyễn Nhứt Lam**

*Sinh viên thực hiện:*  
**Họ tên: Nguyễn Anh Tuấn**  
**MSSV: 110121123**  
**Lớp: DA21TTA**

*Trà Vinh, tháng 01 năm 2025*

**KHOA KỸ THUẬT VÀ CÔNG NGHỆ  
BỘ MÔN CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN CHUYÊN NGÀNH  
HỌC KỲ I, NĂM HỌC 2024 – 2025**

# **XÂY DỰNG WEBSITE TÌM ĐỒ THẮT LẠC**

*Giáo viên hướng dẫn:*  
**TS. Nguyễn Nhứt Lam**

*Sinh viên thực hiện:*  
**Họ tên: Nguyễn Anh Tuấn**  
**MSSV: 110121123**  
**Lớp: DA21TTA**

*Trà Vinh, tháng 01 năm 2025*

### NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

This image shows a full page of primary-ruled paper. It features multiple sets of three horizontal lines: a solid top line, a dashed middle line, and a solid bottom line. These lines are evenly spaced across the entire page, providing a guide for letter height and placement for young learners. The background is white, and there are no margins or additional markings.

Trà Vinh, ngày ..... tháng ..... năm .....

**Giáo viên hướng dẫn**  
(Ký tên và ghi rõ họ tên)

## NHẬN XÉT CỦA THÀNH VIÊN HỘI ĐỒNG

[illegible]

Trà Vinh, ngày ..... tháng ..... năm .....

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

## LỜI CẢM ƠN

Lời nói đầu tiên cho em xin gửi lời cảm ơn và biết ơn sâu sắc đến Thầy hướng dẫn Nguyễn Nhứt Lam đã giúp em hoàn thành đề tài “Xây dựng Website Tìm Đồ Thất Lạc” và em cũng xin gửi lời cảm ơn chân thành đến các thầy cô đã tận tình hướng dẫn, giảng dạy trong suốt quá trình học tập, nghiên cứu và rèn luyện ở Trường Đại học Trà Vinh.

Mặc dù em đã rất cố gắng để thực hiện đồ án một cách hoàn chỉnh nhất nhưng do thời gian có hạn, trình độ hiểu biết và nhận thức của em chưa cao nên trong đồ án của em không thể tránh khỏi những thiếu sót, em rất mong nhận được sự đóng góp ý kiến của các thầy, cô và bạn bè để em có thể hoàn thiện đồ án này tốt hơn.

*Em xin chân thành cảm ơn!*

**Sinh viên**

## MỤC LỤC

MỞ ĐẦU .....	12
Lý do chọn đề tài .....	12
Mục đích.....	12
Đối tượng.....	13
Phạm vi nghiên cứu .....	13
CHƯƠNG 1: TỔNG QUAN .....	14
1.1. MÔ TẢ VẤN ĐỀ .....	14
1.1.1. Tình trạng mất đồ cá nhân phổ biến .....	14
1.1.2. Tác động của việc mất đồ cá nhân.....	14
1.1.3. Các giải pháp tìm đồ thất lạc hiện tại .....	14
1.1.4. Thiếu nền tảng chung để kết nối người nhặt đồ và người đánh rơi đồ.....	14
1.1.5. Thiếu tính năng tìm kiếm hiệu quả và chính xác.....	15
1.1.6. Thiếu tính năng thông báo tự động.....	15
1.1.7. Nhu cầu xây dựng nền tảng tìm đồ thất lạc hiệu quả .....	15
1.1.8. Giải pháp sử dụng công nghệ Web để giải quyết vấn đề .....	15
1.2. MỤC TIÊU NGHIÊN CỨU .....	16
1.2.1. Xây dựng nền tảng kết nối người mất đồ và người nhặt được đồ .....	16
1.2.2. Phát triển công cụ tìm kiếm và quản lý thông tin hiệu quả .....	16
1.2.3. Tích hợp tính năng bản đồ và thông báo qua email.....	16
1.2.4. Đảm bảo bảo mật thông tin người dùng và hiệu suất của hệ thống.....	17
1.3. HUỚNG TIẾP CẬN.....	17
1.3.1. Phát triển từ nhu cầu thực tế của người dùng .....	17
1.3.2. Xây dựng nền tảng dễ sử dụng và thân thiện với người dùng .....	17
1.3.3. Tích hợp công nghệ hiện đại (Maps API, hệ thống email).....	17
1.3.4. Đảm bảo tính bảo mật và an toàn thông tin.....	18

1.3.5. Tối ưu hóa hiệu suất và khả năng mở rộng của hệ thống .....	18
1.4. ĐỐI TƯỢNG VÀ PHẠM VI NGHIÊN CỨU .....	18
1.4.1. Đối tượng nghiên cứu .....	18
1.4.2. Phạm vi nghiên cứu .....	18
CHƯƠNG 2: NGHIÊN CỨU LÝ THUYẾT .....	19
2.1. LÝ THUYẾT .....	19
2.1.1. Giới thiệu về ReactJS .....	19
2.1.2. Giới thiệu về NodeJS.....	25
2.1.3. Giới thiệu về MySQL .....	31
2.2. NGHIÊN CỨU TỔNG QUAN NGƯỜI DÙNG .....	33
2.2.1. Xây dựng hệ thống tìm đồ thất lạc .....	33
2.2.2. Tích hợp công nghệ hiện đại .....	33
2.3. TÍNH BẢO MẬT VÀ KHẢ NĂNG ĐÁP ỨNG CỦA WEBSITE.....	34
2.3.1. Tính khả thi.....	34
2.3.2. Hiệu quả trong việc tìm kiếm đồ thất lạc .....	34
2.3.3. Bảo mật và toàn vẹn dữ liệu .....	34
2.4. QUY TRÌNH VÀ PHƯƠNG PHÁP .....	34
2.4.1. Nghiên cứu tài liệu .....	34
2.4.2. Thiết kế và triển khai.....	34
2.4.3. Kiểm thử và đánh giá.....	35
CHƯƠNG 3: HIỆN THỰC HÓA NGHIÊN CỨU .....	36
3.1. CHUẨN BỊ NỀN TẢNG NGHIÊN CỨU.....	36
3.1.1. Lựa chọn công nghệ .....	36
3.1.2. Môi trường.....	36
3.2. QUY TRÌNH THỰC HIỆN NGHIÊN CỨU.....	37
3.2.1. Giao diện người dùng .....	37
3.2.2. Chức năng.....	37

CHƯƠNG 4: KẾT QUẢ NGHIÊN CỨU .....	38
4.1. KẾT QUẢ HOÀN THÀNH DỰ ÁN .....	38
4.1.1. Đăng nhập.....	38
4.1.2. Đăng kí .....	39
4.1.3. Trang chủ.....	39
4.1.4. Đăng tin mới.....	40
4.1.5. Chỉnh sửa bài đăng .....	41
4.1.6. Chi tiết bài đăng.....	41
4.1.7. Chỉnh sửa thông tin người dùng .....	42
4.1.8. Tìm kiếm thông tin bài đăng .....	42
4.1.9. Gửi thông báo qua email khi có bài đăng mới .....	43
4.1.10. Giao diện của trang quản trị website .....	43
4.1.11. Quên mật khẩu, OTP gửi đến email .....	45
4.1. KIỂM THỬ HỆ THỐNG .....	46
CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN .....	47
5.1. KẾT LUẬN CHUNG CHO DỰ ÁN .....	47
5.1.1. Kết quả đạt được.....	47
5.1.2. Hạn chế .....	47
5.1. HƯỚNG PHÁT TRIỂN .....	47
5.2.1. Tích hợp AI: .....	47
5.2.2. Phát triển ứng dụng di động: .....	47
5.2.3. Hỗ trợ đa ngôn ngữ:.....	47



## DANH MỤC HÌNH ẢNH – BẢNG BIỂU

Hình 1: Logo ReactJS .....	19
Hình 2: Virtual DOM .....	20
Hình 3: JSX .....	21
Hình 4: Component .....	22
Hình 5: Logo NodeJS .....	25
Hình 6: Module NodeJS .....	26
Hình 7: Cluster NodeJS .....	27
Hình 8: Đối tượng toàn cục NodeJS.....	28
Hình 9: Xử lý lỗi NodeJS .....	28
Hình 10: Logo MySQL .....	31
Hình 11: Giao diện đăng nhập.....	38
Hình 12: Giao diện đăng kí .....	39
Hình 13: Trang chủ.....	39
Hình 14: Giao diện ĐỒ NHẬT ĐƯỢC.....	40
Hình 15: Giao diện ĐĂNG TIN MỚI .....	40
Hình 16: Giao diện CHỈNH SỬA BÀI ĐĂNG.....	41
Hình 17: Giao diện CHI TIẾT BÀI ĐĂNG .....	41
Hình 18: Giao diện CHỈNH SỬA THÔNG TIN NGƯỜI DÙNG.....	42
Hình 19: Giao diện TÌM KIẾM THÔNG TIN BÀI ĐĂNG.....	42
Hình 20: Thông báo bài đăng mới qua email người dùng.....	43
Hình 21: Giao diện TRANG QUẢN TRỊ.....	43
Hình 22: Giao diện QUẢN LÝ TÀI KHOẢN NGƯỜI DÙNG.....	44
Hình 23: Giao diện QUẢN LÝ BÀI ĐĂNG CÓ TRÊN HỆ THỐNG .....	44
Hình 24: Giao diện QUẢN LÝ BÀI ĐĂNG ĐÃ CÓ CHỦ SỞ HỮU .....	45

Hình 25: Nhập email yêu cầu gửi OTP .....	45
Hình 26: OTP gửi về email .....	45
Hình 27: Nhập OTP và đặt lại mật khẩu mới.....	46

## TÓM TẮT ĐỒ ÁN CHUYÊN NGÀNH

### Những nghiên cứu về đề tài

Trong những năm gần đây, vấn đề mất mát tài sản cá nhân như ví, điện thoại, đồ dùng cá nhân hay tài liệu quan trọng ngày càng trở thành một thách thức lớn đối với cộng đồng. Việc tìm kiếm và hồi phục các đồ vật thất lạc không chỉ tốn thời gian mà còn có thể dẫn đến nhiều rủi ro về bảo mật thông tin cá nhân. Chính vì vậy, việc nghiên cứu và phát triển các giải pháp công nghệ để hỗ trợ người dùng trong việc tìm lại đồ thất lạc đã nhận được sự quan tâm đặc biệt từ các chuyên gia trong lĩnh vực công nghệ thông tin, khoa học dữ liệu và thiết kế trải nghiệm người dùng.

Một số nghiên cứu trước đây tập trung vào việc sử dụng các công nghệ như mã QR, nhận diện hình ảnh, vị trí GPS để giúp tìm đồ vật thất lạc, nhưng các giải pháp này vẫn còn nhiều hạn chế. Một số hệ thống có thể yêu cầu người dùng phải đăng ký hoặc cài đặt phần mềm trước khi bị mất đồ, điều này không phải lúc nào cũng khả thi. Các hệ thống còn lại chủ yếu tập trung vào một lĩnh vực cụ thể như điện thoại di động hay ví tiền, chưa có một hệ thống tổng thể và dễ sử dụng cho nhiều loại đồ vật khác nhau. Vì vậy, đồ án này hướng tới nghiên cứu và phát triển một website có khả năng tìm kiếm đồ thất lạc hiệu quả, linh hoạt và dễ dàng sử dụng cho nhiều đối tượng người dùng khác nhau.

### Hướng tiếp cận vấn đề

Hướng tiếp cận vấn đề của đồ án là xây dựng một website hỗ trợ người dùng tìm kiếm đồ thất lạc thông qua việc đăng tải thông tin chi tiết về đồ vật bị mất và sử dụng công nghệ tìm kiếm thông minh. Website sẽ cho phép người dùng đăng ký thông tin về đồ vật mất tích, bao gồm tên đồ vật, đặc điểm, hình ảnh, thời gian và địa điểm mất. Người dùng khác có thể tìm kiếm đồ thất lạc dựa trên các tiêu chí này và nhận thông báo nếu có đồ vật trùng với mô tả đã đăng.

Đặc biệt, hệ thống sẽ sử dụng ReactJS để xây dựng giao diện người dùng (UI) vì tính năng tương tác mượt mà, dễ dàng mở rộng và tái sử dụng các thành phần giao diện. Điều này giúp phát triển nhanh chóng, dễ dàng bảo trì và tối ưu hóa trải nghiệm

người dùng, đồng thời giúp tiết kiệm thời gian trong việc triển khai các tính năng động.

### **Cách giải quyết vấn đề**

Để giải quyết vấn đề tìm đồ thất lạc, hệ thống sẽ được phát triển với các thành phần chính:

- Frontend: Giao diện người dùng được phát triển bằng ReactJS, giúp cung cấp trải nghiệm mượt mà và tương tác động. Với ReactJS, chúng tôi có thể dễ dàng xây dựng các thành phần giao diện như form đăng tải thông tin đồ vật mất, danh sách đồ vật tìm thấy, các bộ lọc tìm kiếm, và các thông báo tùy chỉnh. ReactJS giúp tối ưu hóa hiệu suất của website với khả năng cập nhật và render lại giao diện nhanh chóng mà không cần phải tải lại toàn bộ trang.
- Backend: Hệ thống backend sẽ sử dụng Node.js để xử lý các yêu cầu của người dùng, lưu trữ và quản lý cơ sở dữ liệu thông qua MongoDB hoặc MySQL. Backend sẽ xử lý các yêu cầu như đăng ký, đăng nhập, tìm kiếm đồ vật và gửi thông báo cho người dùng qua email hoặc SMS khi có đồ vật trùng khớp với thông tin đã đăng.
- Tính năng tìm kiếm thông minh: Để cải thiện khả năng tìm kiếm, hệ thống sẽ tích hợp một công cụ tìm kiếm thông minh, cho phép người dùng tìm đồ vật thất lạc qua các tiêu chí như tên, loại đồ vật, thời gian, địa điểm và hình ảnh. Đặc biệt, với tính năng so khớp hình ảnh, người dùng có thể tải lên hình ảnh của đồ vật và hệ thống sẽ tự động tìm kiếm trong cơ sở dữ liệu những đồ vật có hình ảnh tương tự.
- Thông báo: Hệ thống sẽ tự động gửi thông báo qua email hoặc SMS cho người dùng khi có đồ vật trùng khớp với mô tả đã đăng. Tính năng này giúp tăng khả năng tìm lại đồ vật của người dùng mà không cần phải kiểm tra lại toàn bộ các kết quả tìm kiếm.

### **Kết quả đạt được sau khi nghiên cứu và triển khai**

Sau khi triển khai và nghiên cứu, hệ thống website tìm đồ thất lạc sẽ hoàn thiện và hoạt động hiệu quả. Các tính năng chính của website như đăng tải thông tin đồ vật mất, tìm kiếm thông minh qua các bộ lọc và hình ảnh, cùng với hệ thống thông báo qua email và SMS đều đã được triển khai thành công. Giao diện người dùng xây dựng

bằng ReactJS mang lại trải nghiệm mượt mà, dễ sử dụng, và có thể mở rộng dễ dàng trong tương lai.

Hệ thống cũng đã chứng minh khả năng tìm kiếm đồ vật thất lạc chính xác và nhanh chóng, với tỷ lệ thành công cao trong việc kết nối người mất đồ và người tìm thấy đồ vật. Công cụ tìm kiếm hình ảnh giúp tăng độ chính xác trong việc so khớp các vật phẩm, đồng thời tạo ra cơ sở dữ liệu đồ vật mất và tìm thấy ngày càng phong phú, hệ thống đã chứng minh tiềm năng lớn trong việc giúp cộng đồng giải quyết vấn đề mất đồ, đặc biệt tại các khu vực đông người như thành phố, công viên, trung tâm mua sắm, hoặc các phương tiện công cộng. Website không chỉ giúp tiết kiệm thời gian tìm kiếm mà còn tạo ra một môi trường cộng đồng hỗ trợ lẫn nhau trong việc tìm lại đồ vật cá nhân.

## MỞ ĐẦU

### Lý do chọn đề tài

Mất đồ cá nhân là một vấn đề khá phổ biến trong đời sống hàng ngày, đặc biệt trong các môi trường công cộng như công viên, khu mua sắm, phương tiện giao thông công cộng, nơi mà các vật dụng cá nhân dễ bị thất lạc. Trong nhiều trường hợp, việc tìm lại đồ thất lạc không chỉ tốn thời gian mà còn có thể gây ra những rủi ro nghiêm trọng về bảo mật thông tin, tài sản cá nhân của người bị mất. Mặc dù hiện nay có một số giải pháp tìm kiếm đồ vật qua các ứng dụng di động hoặc dịch vụ trực tuyến, nhưng phần lớn các hệ thống này vẫn còn thiếu tính linh hoạt, chưa đầy đủ về tính năng hoặc chưa kết nối được người mất đồ và người nhặt được một cách dễ dàng.

Với mục tiêu xây dựng một nền tảng giúp kết nối người mất đồ và người nhặt được đồ vật, từ đó tạo ra một cộng đồng hỗ trợ nhau tìm lại đồ thất lạc, đề tài "Xây dựng Website Tìm Đồ Thất Lạc" được lựa chọn để nghiên cứu và triển khai. Đây là một giải pháp mang tính ứng dụng cao, giúp người dùng tiết kiệm thời gian và công sức trong việc tìm lại đồ vật bị mất.

### Mục đích

Mục đích chính của đề án là phát triển một website giúp người dùng đăng tải thông tin về đồ thất lạc hoặc đồ nhặt được, đồng thời cung cấp các chức năng tìm kiếm thông minh, giúp người dùng dễ dàng tìm lại đồ vật bị mất. Các tính năng của website bao gồm:

- Đăng tải thông tin về đồ thất lạc và đồ nhặt được, bao gồm mô tả chi tiết và hình ảnh của đồ vật.
- Tìm kiếm đồ thất lạc theo các tiêu chí như ngày mất, địa điểm, loại đồ vật, giúp người dùng nhanh chóng tìm được thông tin liên quan.
- Tích hợp hệ thống thông báo qua email khi có thông tin mới liên quan đến đồ vật thất lạc hoặc nhặt được.
- Cung cấp khu vực quản lý tài khoản người dùng, bao gồm các chức năng đăng ký, đăng nhập, chỉnh sửa thông tin cá nhân.

- Tích hợp bản đồ để giúp người dùng chỉ định vị trí mất đồ hoặc nơi tìm được đồ vật, từ đó kết nối người mất đồ và người nhặt được.

### **Đối tượng**

Đối tượng sử dụng website tìm đồ thất lạc bao gồm:

- **Người mất đồ:** Những người đã mất đồ vật cá nhân như điện thoại, ví tiền, giấy tờ, hoặc các vật dụng quan trọng khác và mong muốn tìm lại chúng.
- **Người nhặt được đồ:** Những người tình cờ tìm thấy đồ vật thất lạc và muốn trả lại cho chủ nhân của chúng.
- **Quản trị viên:** Người quản lý hệ thống, giám sát các bài đăng, xử lý các vấn đề liên quan đến dữ liệu và đảm bảo hoạt động của website luôn ổn định.

### **Phạm vi nghiên cứu**

Đề án tập trung vào việc phát triển một website tìm đồ thất lạc có các tính năng cơ bản sau:

- **Đăng tải và tìm kiếm thông tin đồ vật thất lạc:** Cho phép người dùng đăng thông tin về đồ thất lạc hoặc đồ nhặt được, bao gồm các thông tin như tên đồ vật, mô tả, hình ảnh, vị trí và ngày giờ mất.
- **Quản lý người dùng:** Cho phép người dùng đăng ký tài khoản, đăng nhập, chỉnh sửa thông tin cá nhân và theo dõi các bài đăng của mình.
- **Tính năng thông báo qua email:** Gửi thông báo cho người dùng khi có thông tin mới liên quan đến đồ thất lạc hoặc đồ nhặt được.
- **Tích hợp bản đồ:** Sử dụng API Leaflet Maps để chỉ định vị trí mất đồ hoặc tìm đồ, giúp người dùng dễ dàng tìm kiếm và kết nối.

## CHƯƠNG 1: TỔNG QUAN

### 1.1. MÔ TẢ VẤN ĐỀ

#### 1.1.1. Tình trạng mất đồ cá nhân phổ biến

Trong cuộc sống hàng ngày, mất đồ vật cá nhân là một vấn đề khá phổ biến. Các đồ vật như ví tiền, điện thoại di động, giấy tờ tùy thân, thẻ ngân hàng, hoặc các vật dụng quan trọng khác thường xuyên bị mất ở những nơi đông người như trạm xe buýt, công viên, trung tâm thương mại, hay thậm chí trong các phương tiện giao thông công cộng. Sự việc này có thể xảy ra bất kỳ lúc nào và gây ra rất nhiều phiền toái cho người bị mất đồ.

#### 1.1.2. Tác động của việc mất đồ cá nhân

Việc mất đồ không chỉ gây bất tiện về mặt vật chất mà còn có thể mang lại những hậu quả nghiêm trọng về mặt tài chính và bảo mật. Đặc biệt là khi những đồ vật bị mất có chứa thông tin cá nhân như giấy tờ tùy thân, thẻ ngân hàng, hoặc các thiết bị điện tử có chứa dữ liệu quan trọng. Ngoài ra, mất đồ còn có thể khiến người dùng cảm thấy lo lắng, mất tự tin, và đôi khi ảnh hưởng đến công việc hoặc các hoạt động hàng ngày.

#### 1.1.3. Các giải pháp tìm đồ thất lạc hiện tại

Hiện nay, có một số giải pháp tìm kiếm đồ thất lạc, chẳng hạn như nhờ bạn bè, người quen thông báo hoặc sử dụng các dịch vụ trực tuyến hỗ trợ tìm đồ. Tuy nhiên, những giải pháp này vẫn còn rất hạn chế, chủ yếu chỉ giới hạn trong việc đăng tin tìm đồ hoặc tìm kiếm đồ đã mất trên các nền tảng không được tối ưu hóa hoặc không có hệ thống kết nối rõ ràng giữa người mất đồ và người nhặt được. Bên cạnh đó, những dịch vụ này đôi khi yêu cầu người dùng phải cài đặt ứng dụng trước khi xảy ra sự cố, điều này không phải lúc nào cũng thực tế.

#### 1.1.4. Thiếu nền tảng chung để kết nối người nhặt đồ và người đánh rơi đồ

Một trong những vấn đề lớn khiến việc tìm lại đồ thất lạc trở nên khó khăn chính là thiếu một nền tảng chung để kết nối người mất đồ với người nhặt được đồ. Hiện tại, đa phần các hệ thống tìm kiếm đồ thất lạc không có cơ chế kết nối giữa các bên mà chỉ đơn giản là đăng tải thông tin mà không có sự tương tác giữa người đăng



tin và người có thể giúp tìm lại đồ vật. Điều này dẫn đến việc nhiều đồ vật bị mất không thể tìm lại, hoặc quá trình tìm kiếm kéo dài và không hiệu quả.

#### **1.1.5. Thiếu tính năng tìm kiếm hiệu quả và chính xác**

Một vấn đề khác là các công cụ tìm kiếm đồ thất lạc hiện tại chưa thực sự hiệu quả và chính xác. Các hệ thống tìm kiếm chỉ cho phép người dùng tìm theo các từ khóa cơ bản như tên đồ vật hoặc thời gian mất, mà không có khả năng lọc theo các tiêu chí cụ thể như địa điểm, loại đồ vật, hoặc thậm chí là hình ảnh. Điều này khiến người dùng khó khăn trong việc tìm lại đồ vật của mình trong một cơ sở dữ liệu không được tổ chức khoa học và dễ dàng tìm kiếm.

#### **1.1.6. Thiếu tính năng thông báo tự động**

Một vấn đề khác trong quá trình tìm đồ thất lạc là việc thiếu thông báo tự động khi có thông tin mới liên quan đến đồ vật. Người dùng thường phải tự theo dõi tình hình tìm kiếm mà không nhận được sự hỗ trợ từ hệ thống. Việc không có hệ thống thông báo tự động làm giảm khả năng người mất đồ và người nhặt được đồ có thể kết nối và trao đổi thông tin một cách kịp thời.

#### **1.1.7. Nhu cầu xây dựng nền tảng tìm đồ thất lạc hiệu quả**

Với những vấn đề trên, nhu cầu về một nền tảng hỗ trợ tìm đồ thất lạc hiệu quả ngày càng trở nên cấp thiết. Việc xây dựng một hệ thống tìm kiếm đồ thất lạc trực tuyến, kết nối người mất đồ và người nhặt được đồ thông qua một nền tảng duy nhất là rất cần thiết. Nền tảng này cần cung cấp các tính năng như đăng tin tìm đồ, tìm kiếm đồ thất lạc qua các tiêu chí rõ ràng (thời gian, địa điểm, loại đồ vật), và hệ thống thông báo tự động để người dùng có thể nắm bắt thông tin mới ngay lập tức.

#### **1.1.8. Giải pháp sử dụng công nghệ Web để giải quyết vấn đề**

Để giải quyết các vấn đề trên, việc xây dựng một website tìm đồ thất lạc là giải pháp khả thi và hiệu quả. Website này sẽ giúp người dùng đăng tải thông tin về đồ vật mất và nhặt được, đồng thời cung cấp các công cụ tìm kiếm thông minh, lọc theo các tiêu chí như ngày mất, địa điểm, loại đồ vật. Ngoài ra, tính năng tích hợp bản đồ và thông báo qua email sẽ giúp quá trình tìm đồ thất lạc trở nên dễ dàng và hiệu quả hơn. Kèm theo đó và tính năng tích hợp bản đồ và thông báo qua email sẽ giúp quá trình tìm đồ thất lạc trở nên dễ dàng và hiệu quả hơn.

## **1.2. MỤC TIÊU NGHIÊN CỨU**

Mục tiêu nghiên cứu của đề tài “Xây dựng Website tìm đồ thất lạc” bao gồm các mục tiêu chính sau:

### **1.2.1. Xây dựng nền tảng kết nối người mất đồ và người nhặt được đồ**

Mục tiêu chính của đề tài là tạo ra một website giúp kết nối người mất đồ và người nhặt được. Thông qua nền tảng này, người dùng có thể dễ dàng đăng tin về đồ vật mình mất hoặc nhặt được, giúp hai bên giao tiếp và phối hợp để trả lại đồ vật. Đây là một giải pháp tiện ích để giảm thiểu tình trạng mất đồ không tìm lại được, đặc biệt trong các khu vực công cộng đông đúc. Bằng cách tạo ra một hệ thống nơi mọi người có thể thông báo về đồ vật mình tìm thấy, website sẽ giúp tăng khả năng tìm lại đồ thất lạc và cải thiện trải nghiệm người dùng.

### **1.2.2. Phát triển công cụ tìm kiếm và quản lý thông tin hiệu quả**

Một mục tiêu quan trọng khác là phát triển tính năng tìm kiếm thông minh cho website, giúp người dùng có thể dễ dàng tìm lại đồ vật của mình dựa trên các tiêu chí rõ ràng như loại đồ, địa điểm mất đồ, thời gian mất đồ, và thậm chí là hình ảnh mô tả đồ vật. Hệ thống sẽ sử dụng các thuật toán lọc kết quả để cho phép người dùng tìm kiếm một cách chính xác và nhanh chóng, tăng tỷ lệ thành công trong việc tìm lại đồ vật. Ngoài ra, website cũng sẽ cung cấp khả năng quản lý thông tin người dùng và các bài đăng tìm đồ thất lạc một cách hiệu quả, giúp đảm bảo dữ liệu luôn được tổ chức và dễ dàng truy xuất.

### **1.2.3. Tích hợp tính năng bản đồ và thông báo qua email**

Website sẽ tích hợp bản đồ (Google Maps) để người dùng có thể chỉ định vị trí nơi đồ vật bị mất hoặc nơi đã nhặt được. Tính năng này không chỉ giúp xác định chính xác vị trí của đồ vật mà còn tạo điều kiện thuận lợi cho việc liên lạc giữa người mất đồ và người nhặt được. Hơn nữa, hệ thống thông báo qua email sẽ tự động gửi thông tin cho người dùng khi có tin tức mới về đồ thất lạc hoặc đồ nhặt được, giúp người dùng nhận được các cập nhật kịp thời mà không phải kiểm tra website thường xuyên. Điều này sẽ nâng cao hiệu quả trong quá trình tìm kiếm và trả lại đồ vật.

#### **1.2.4. Đảm bảo bảo mật thông tin người dùng và hiệu suất của hệ thống**

Bảo mật là một yếu tố quan trọng trong việc xây dựng website, đặc biệt khi hệ thống yêu cầu người dùng cung cấp thông tin cá nhân và đăng tải bài viết về đồ vật của mình. Mục tiêu nghiên cứu là đảm bảo thông tin người dùng được bảo vệ an toàn khỏi các mối nguy cơ xâm nhập, đồng thời đảm bảo hệ thống hoạt động ổn định và mượt mà, ngay cả khi số lượng người dùng tăng cao. Website sẽ được tối ưu hóa để xử lý một lượng lớn yêu cầu truy cập đồng thời mà không làm giảm hiệu suất hoặc gây gián đoạn dịch vụ, từ đó nâng cao trải nghiệm người dùng.

### **1.3. HƯỚNG TIẾP CẬN**

#### **1.3.1. Phát triển từ nhu cầu thực tế của người dùng**

Hướng tiếp cận đầu tiên là xây dựng website dựa trên nhu cầu thực tế của cộng đồng trong việc tìm lại đồ thất lạc. Đây là một vấn đề phổ biến mà nhiều người gặp phải hàng ngày, đặc biệt là trong những môi trường công cộng đông đúc. Việc tìm kiếm đồ thất lạc không chỉ giúp người mất đồ tìm lại tài sản của mình mà còn tạo ra sự gắn kết giữa các thành viên trong cộng đồng, giúp ích cho xã hội. Vì vậy, website sẽ hướng đến việc đáp ứng nhu cầu này bằng một giải pháp đơn giản, hiệu quả và dễ tiếp cận.

#### **1.3.2. Xây dựng nền tảng dễ sử dụng và thân thiện với người dùng**

Một hướng tiếp cận quan trọng là thiết kế giao diện người dùng dễ sử dụng, thân thiện và trực quan, để mọi đối tượng người dùng có thể dễ dàng tiếp cận và sử dụng website. Dù người dùng có ít hoặc không có kinh nghiệm sử dụng công nghệ, họ vẫn có thể nhanh chóng đăng bài, tìm kiếm đồ vật hoặc tương tác với những người khác. Giao diện sẽ được tối ưu hóa cho các thiết bị di động và máy tính, giúp người dùng có thể truy cập website mọi lúc, mọi nơi.

#### **1.3.3. Tích hợp công nghệ hiện đại (Maps API, hệ thống email)**

Hướng tiếp cận thứ ba là ứng dụng các công nghệ hiện đại để nâng cao hiệu quả của website. Việc tích hợp Maps API để xác định vị trí đồ thất lạc hoặc đồ nhặt được sẽ giúp người dùng có thể chỉ định chính xác khu vực mất đồ. Đồng thời, việc sử dụng hệ thống thông báo qua email giúp cập nhật thông tin mới một cách nhanh chóng và

kip thời, giúp người dùng luôn được thông báo khi có thông tin mới liên quan đến đồ vật mình tìm kiếm.

#### **1.3.4. Đảm bảo tính bảo mật và an toàn thông tin**

Một yếu tố quan trọng trong việc phát triển website là bảo mật thông tin người dùng. Để đảm bảo sự an tâm cho người sử dụng, website sẽ tích hợp các cơ chế bảo mật mạnh mẽ như mã hóa thông tin người dùng, bảo vệ dữ liệu đăng tải về đồ thất lạc. Điều này giúp người dùng có thể yên tâm khi sử dụng dịch vụ mà không lo ngại về việc bị xâm phạm thông tin cá nhân hoặc thông tin liên quan đến đồ vật của mình.

#### **1.3.5. Tối ưu hóa hiệu suất và khả năng mở rộng của hệ thống**

Hướng tiếp cận cuối cùng là tối ưu hóa hệ thống để có thể xử lý lượng người dùng lớn và nhiều dữ liệu mà không làm ảnh hưởng đến hiệu suất của website. Đảm bảo rằng website có thể duy trì tốc độ tải trang nhanh, không bị gián đoạn trong quá trình sử dụng, đặc biệt là khi lượng người dùng tăng lên. Điều này sẽ giúp website hoạt động hiệu quả và có thể mở rộng trong tương lai, phục vụ cho một cộng đồng lớn hơn.

### **1.4. ĐỐI TƯỢNG VÀ PHẠM VI NGHIÊN CỨU**

#### **1.4.1. Đối tượng nghiên cứu**

Đối tượng nghiên cứu của đề tài là người dùng trên nền tảng website tìm đồ thất lạc, bao gồm hai nhóm chính: người mất đồ và người nhặt được đồ. Các đối tượng này sẽ sử dụng website để đăng tin về đồ vật mình tìm thấy hoặc đã mất, và tìm kiếm thông tin liên quan đến đồ vật bị thất lạc. Ngoài ra, nghiên cứu cũng bao gồm việc xây dựng và phát triển các tính năng của website, như hệ thống tìm kiếm, tích hợp bản đồ, gửi thông báo qua email, và đảm bảo tính bảo mật cho người dùng.

#### **1.4.2. Phạm vi nghiên cứu**

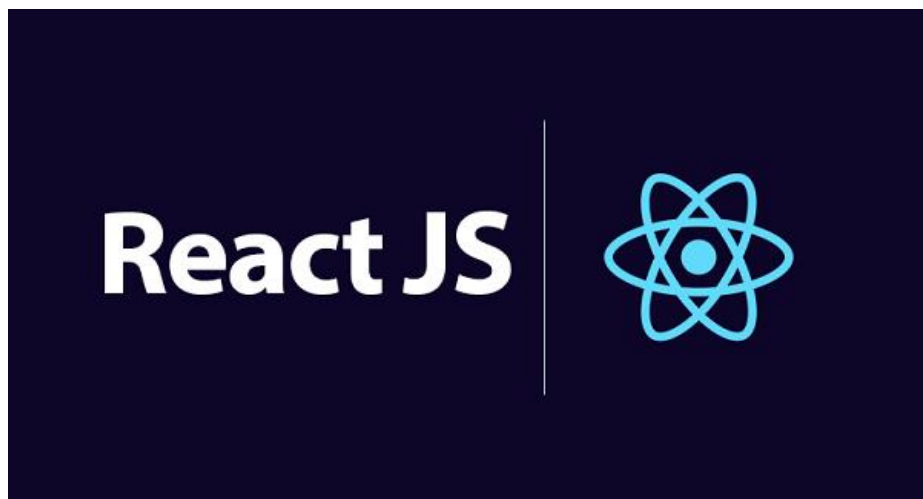
Phạm vi nghiên cứu tập trung vào việc phát triển một website có chức năng tìm đồ thất lạc và đồ nhặt được, với các công cụ tìm kiếm thông minh, tích hợp bản đồ, và hệ thống thông báo qua email. Phạm vi nghiên cứu không bao gồm việc phát triển các ứng dụng di động riêng biệt, mặc dù website sẽ được tối ưu hóa để hỗ trợ người dùng trên thiết bị di động. Ngoài ra, phạm vi nghiên cứu còn bao gồm việc thiết kế giao diện người dùng và đảm bảo tính bảo mật của dữ liệu người dùng trong suốt quá trình sử dụng dịch vụ.

## CHƯƠNG 2: NGHIÊN CỨU LÝ THUYẾT

### 2.1. LÝ THUYẾT

#### 2.1.1. Giới thiệu về ReactJS

React.js là một thư viện Javascript đang nổi lên trong những năm gần đây với xu hướng Single Page Application. Trong khi những framework khác cố gắng hướng đến một mô hình MVC hoàn thiện thì React nổi bật với sự đơn giản và dễ dàng phối hợp với những thư viện Javascript khác. Nếu như AngularJS là một Framework cho phép nhúng code javascript trong code html thông qua các attribute như ng-model, ng-repeat...thì với react là một library cho phép nhúng code html trong code javascript nhờ vào JSX, bạn có thể dễ dàng lồng các đoạn HTML vào trong JS. Tích hợp giữa javascript và HTML vào trong JSX làm cho các component dễ hiểu hơn.



*Hình 1: Logo ReactJS*

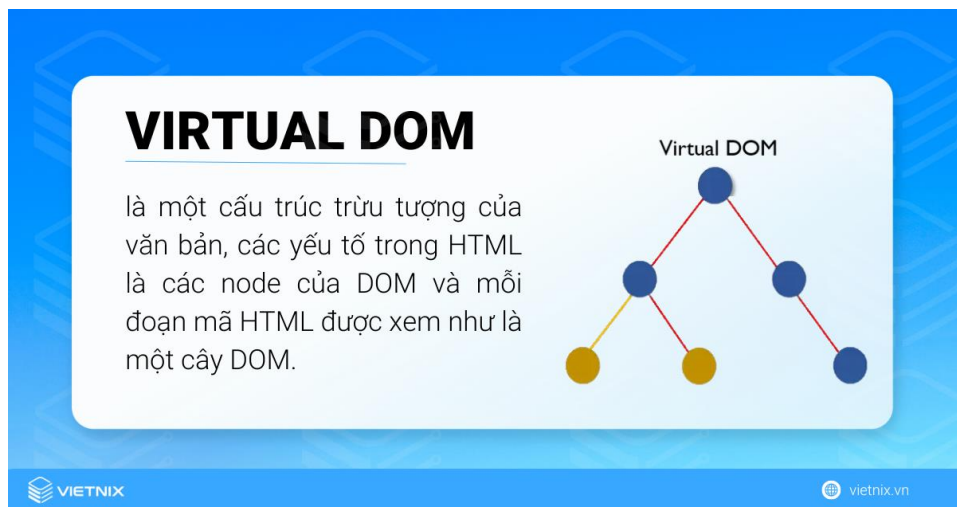
##### 2.1.1.1. ReactJS là gì ?

React là một thư viện UI phát triển tại Facebook để hỗ trợ việc xây dựng những thành phần (components) UI có tính tương tác cao, có trạng thái và có thể sử dụng lại được. React được sử dụng tại Facebook trong production, và Instagram được viết hoàn toàn trên React.

Một trong những điểm hấp dẫn của React là thư viện này không chỉ hoạt động trên phía client, mà còn được render trên server và có thể kết nối với nhau. React so sánh sự thay đổi giữa các giá trị của lần render này với lần render trước

và cập nhật ít thay đổi nhất trên DOM. Trước khi đến cài đặt và cấu hình, chúng ta sẽ đi đến một số khái niệm cơ bản:

#### 2.1.1.2. Virtual DOM:



Hình 2: Virtual DOM

Document Object Model (DOM) là một cấu trúc trừu tượng của văn bản, các yếu tố trong HTML là các node của DOM và mỗi đoạn mã HTML được xem như là một cây DOM. Cây DOM mô phỏng cấu trúc của một trang web, với mỗi phần tử HTML được biểu diễn bởi một node trong cây. Virtual DOM (DOM ảo) là một biểu diễn nhẹ nhàng của cây DOM thực sự.

Khi trạng thái của ứng dụng ReactJS thay đổi, ReactJS sẽ tạo ra một Virtual DOM mới. Sau đó, ReactJS sử dụng một thuật toán so sánh (diffing) để xác định phần tử nào của cây DOM thực sự cần cập nhật. Cuối cùng, ReactJS chỉ cập nhật những phần tử cần thiết, giúp cải thiện hiệu suất và trải nghiệm người dùng của ứng dụng.

Mặc dù DOM xử lý rất nhanh, nhưng với các ứng dụng SPA, thay đổi DOM liên tục sẽ làm chậm quá trình và gây khó khăn khi triển khai các ứng dụng lớn. Vì vậy, Virtual DOM được sử dụng để thay thế DOM thực tế.

Trong trường hợp bạn không sử dụng ReactJS (và JSX), HTML sẽ được trang web sử dụng để cập nhật lại cây DOM cho chính bản nó. Quá trình thay đổi này diễn ra tự nhiên và không yêu cầu người dùng phải tải lại trang, tuy nhiên, phương pháp này chỉ phù hợp với các trang web nhỏ, đơn giản.

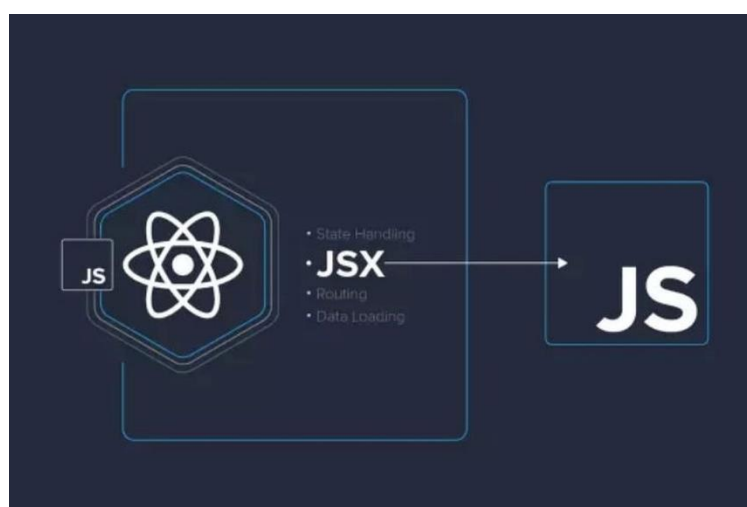
Đối với các trang web lớn, phức tạp, cụ thể là những trang web xử lý nhiều tương tác người dùng, quá trình này sẽ ảnh hưởng đến hiệu suất của trang web ở mức độ rất nghiêm trọng. Lý do là khi người dùng thực hiện một hành động yêu cầu tải lại trang thì toàn bộ cây DOM phải được tải lại, làm ảnh hưởng đáng kể hiệu suất.

Thế nhưng, vấn đề này hoàn toàn có thể giải quyết khi tiến hành render một phần tử JSX, React JS sẽ tạo ra Virtual DOM. Virtual DOM được cập nhật trước rồi đến cây DOM thực tế. React JS lúc này sẽ so sánh Virtual DOM với Virtual DOM trước đó nhằm chắc chắn chỉ có những thay đổi cần thiết được thực hiện trên cây DOM và hiển thị trên màn hình.

Qua đó, có thể thấy, việc sử dụng Virtual DOM giúp tăng tốc độ xử lý và tiết kiệm tài nguyên vì tập trung vào việc cập nhật chỉ một item duy nhất.

Khi không sử dụng ReactJS, toàn bộ cây DOM cần phải cập nhật để thông báo cho sự thay đổi mới. Ngược lại, khi sử dụng React, để xác định những gì đã thay đổi sau khi người dùng thực hiện các hành động trên, React sẽ scan qua Virtual DOM đảm bảo chỉ cập nhật sự thay đổi ở vị trí cần thiết, giúp tiết kiệm đáng kể sức mạnh tính toán cũng như thời gian tải.

### 2.1.1.3. JSX:



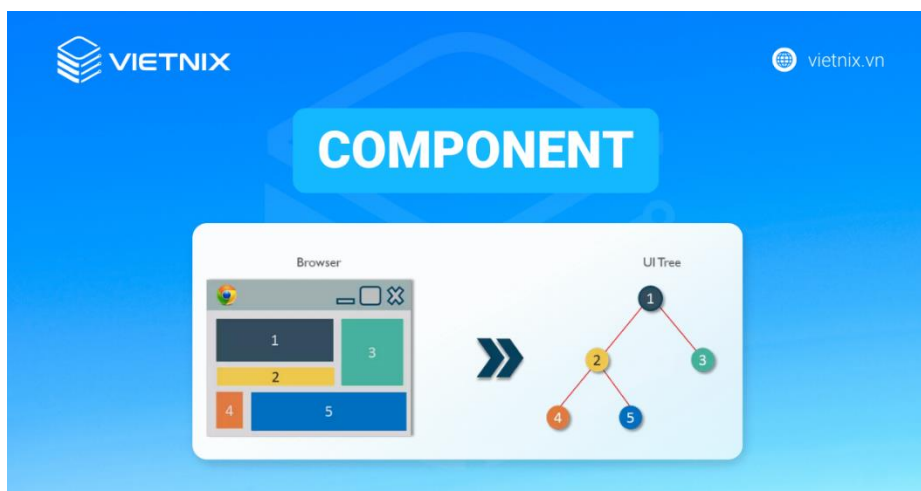
Hình 3: JSX

Yếu tố chính trong một website cơ bản thường là các HTML documents. Các documents được đọc bởi trình duyệt web để hiển thị nội dung của website trên các thiết bị điện tử tablet, máy tính, tablet và điện thoại,...

Trong quá trình này, một Document Object Model (DOM) được tạo ra, đây là một cấu trúc dữ liệu dạng cây đại diện cho cách nội dung trên website được tổ chức và hiển thị. Ngoài ra, khi sử dụng ngôn ngữ JavaScript, developer có thể thêm dynamic content vào các dự án của họ thông qua việc thay đổi cây DOM.

JSX được biết đến như là thuật ngữ viết tắt của JavaScript extension, là một React extension, giúp hỗ trợ viết đoạn mã HTML dễ dàng, có cấu trúc hơn, với cú pháp gần giống HTML. Hơn nữa, ReactJS cũng được hỗ trợ trên tất cả các trình duyệt web hiện đại, nên bạn hoàn toàn có thể sử dụng JSX cho bất kỳ trình duyệt nào bạn đang làm việc.

#### 2.1.1.4. Components:



*Hình 4: Component*

Với các dự án lớn, UI thường rất phức tạp và thường cần được phân chia thành các phần nhỏ hơn được là component. Có hai loại component chính trong React JS: Class component và Function component.

Sử dụng mô hình component sẽ giúp phát triển ứng dụng web, với các phần tử UI là những phần độc lập cho phép sử dụng lại nhiều lần trong nhiều phần khác nhau. Tiện ích này giúp code dễ hiểu, dễ bảo trì, dễ tái sử dụng nhiều lần.



### 2.1.1.5. Props & State:

Props và State đề cập đến 2 loại dữ liệu trong React. Với người mới bắt đầu, sẽ rất khó để phân biệt chúng, nhưng khi đã làm quen với code thì bạn sẽ cảm thấy dễ dàng hơn rất nhiều.

Props khác State về mặt, State chỉ thay đổi bên trong và thường mang tính private. Ngược lại, props thay đổi từ bên ngoài, được truyền từ các component có cấp lớn hơn, cụ thể:

1. Props là tham số được truyền qua truyền lại giữa các react component, sau đó chuyển cho các component tương tự như HTML attributes.
2. State lưu giữ giá trị các thuộc tính trong component cũng như được chứa trong phạm vi component đó. Khi render lại, giá trị của state thay đổi. Cài đặt State đồng nghĩa với việc cài đặt lại hàm `getInitialState()` vào component, rồi trả về yêu cầu của bạn trong state của component này. Ngoài ra, nếu muốn thay đổi state, bạn cần sử dụng `this.setState()`, sau đó truyền vào state mới.

Thông qua State và Props, React JS có thể quản lý và truyền dữ liệu của các trạng thái các thành phần UI, cập nhật giao diện người dùng, đồng thời tạo một cách tiếp cận cấu trúc và quản lý dữ liệu linh hoạt, dễ hiểu.

### 2.1.1.6. Ưu và nhược điểm của ReactJS

React JS là một thư viện JavaScript được ưa chuộng rộng rãi hiện nay nhờ khả năng xây dựng các giao diện người dùng (UI) hiệu quả và linh hoạt. Tuy nhiên, bên cạnh những ưu điểm nổi bật, React JS cũng tồn tại một số hạn chế nhất định. Tiếp theo Vietnix sẽ đi sâu phân tích các ưu nhược điểm của React JS để giúp bạn đưa ra lựa chọn phù hợp nhất cho dự án của mình.

#### Ưu điểm

- **ReactJS phù hợp với đa thể loại website:** ReactJS hỗ trợ đáng kể trong việc khởi tạo website đơn giản hơn vì bạn chỉ cần sử dụng JavaScript và HTML, các công cụ và tính năng khác đã được ReactJS cung cấp để áp dụng cho nhiều trường hợp khác nhau.

- **Dễ dàng tái sử dụng các Component:** Bạn chỉ cần tốn thời gian vào việc xây dựng ban đầu và có thể sử dụng lại trong các dự án sau này nếu bạn xây dựng các component đủ linh hoạt và tốt, có thể đáp ứng các yêu cầu của nhiều dự án khác nhau. Ngoài ra, không chỉ có ReactJS mà các framework hiện nay như Flutter cũng cho phép chúng ta thực hiện điều này.
- **Sử dụng được cho cả ứng dụng di động:** ReactJS thường được biết đến phổ biến trong việc sử dụng để phát triển các website. Nhưng nếu cần xây dựng và phát triển ứng dụng di động, React Native cũng là một lựa chọn tuyệt vời, đây là một framework khác được nghiên cứu và phát triển bởi Facebook. Với React Native, các component có thể được chia sẻ dễ dàng hoặc được tái sử dụng trong các Business Logic trong ứng dụng.
- **Tính hiệu quả cao của React JS:** Nhờ việc tạo ra các DOM ảo giúp các component tồn tại nên cải thiện hiệu suất. Không những vậy, React JS cũng giúp DOM biết những thay đổi nào phải cập nhật và tiến hành thực hiện, giúp tránh xảy ra những thao tác lãng phí trên DOM.
- **Có lợi cho SEO:** ReactJS, bản chất là một thư viện JavaScript, đã được crawl và index bởi Google Search Engine. Tuy nhiên, bạn cũng cần sử dụng một số thư viện bổ sung để có thể tối ưu hóa SEO hiệu quả nhất có thể.
- **Hỗ trợ viết code JS đơn giản hơn đáng kể:** Nội dung cú pháp Javascript mở rộng cho phép kết hợp Javascript với code HTML. Do đó, bạn hoàn toàn có thể thêm vào các đoạn HTML trong render mà không phải nối chuỗi, đây cũng đồng thời là đặc tính mới của React JS khi có thể sử dụng bộ biến đổi JSX để khởi tạo các đối tượng HTML từ các đoạn HTML.
- **Nhiều công cụ mở rộng giúp debug dễ dàng:** Sử dụng các công cụ mở rộng dành riêng cho React JS sẽ giúp debug code nhanh chóng và dễ dàng hơn, giúp tăng tốc độ release sản phẩm và tốc độ coding.

- **Test giao diện dễ dàng:** Do virtual DOM được cài hoàn toàn bằng JS nên việc sử dụng React JS rất đơn giản trong việc viết các test case giao diện.
- **Công cụ phát triển website phổ biến nhất:** Theo số liệu từ Google Trends ở Việt Nam, cụ thể là các trang web tuyển dụng hàng đầu như Topdev, Itviec,.. bạn sẽ thấy nhu cầu công việc cho vị trí React Developer là rất lớn, đi kèm mức lương hấp dẫn. Điều này cho thấy “độ phủ sóng” đáng kể của ReactJS trên thị trường Việt Nam hiện nay.

### Nhược điểm

- Do React chỉ là View Library chứ không phải một MVC framework nên chỉ có thể đáp ứng nhu cầu của tầng View. Hơn nữa, react cũng không có model và controller nên phải sử dụng các thư viện khác.
- Bạn sẽ phải cấu hình lại từ đầu nếu tích hợp react JS vào các framework MVC truyền thống
- So với các kích thước khác, React khá lớn, khoảng 35KB, tương đương với Angular khoảng 39KB. Thế nhưng, Angular là một framework hoàn chỉnh
- Người mới bắt đầu sử dụng sẽ mất nhiều thời gian để học do khá phức tạp.

### 2.1.2. Giới thiệu về NodeJS



Hình 5: Logo NodeJS

### 2.1.2.1. NodeJS là gì?

Node.js là một nền tảng mã nguồn mở được xây dựng trên V8 Javascript engine. Node.js được viết bằng c++ và Javascript. Nền tảng này được phát triển Node.js vào năm 2009 bởi Ryan Lienhart Dahl.

Mục đích của Node.js là cung cấp một môi trường chạy ứng dụng web hiệu quả và không đồng bộ. Nền tảng này sử dụng mô hình I/O không chặn, cho phép xử lý nhiều yêu cầu cùng lúc mà không bị tắc nghẽn.

### 2.1.2.2. Các thành phần của NodeJS

Module:

Module trong Node.js giống như những gói thư viện nhỏ, chứa các hàm, đối tượng và các lớp được viết sẵn để thực hiện các tác vụ cụ thể. Điều này giúp cho việc phát triển ứng dụng trở nên modun hóa và dễ dàng quản lý. Để sử dụng một module trong ứng dụng của bạn, bạn chỉ cần sử dụng hàm require().

Ở Node.JS có nhiều Module với nhiều chức năng khác nhau và đều cần thiết cho một ứng dụng web. Dưới đây là một số Module thường được sử dụng trong ứng dụng web:

Module chính	Mô tả
http	Tạo và quản lý các máy chủ HTTP để xây dựng các ứng dụng web
util	Cung cấp các hàm tiện ích hỗ trợ lập trình, như định dạng dữ liệu, kiểm tra kiểu dữ liệu...
fs	Tương tác với hệ thống file để đọc, ghi, xóa và quản lý các file
url	Phân tích cú pháp URL để trích xuất các thông tin như protocol, host, path...
querystring	Xử lý query string trong URL để lấy ra các tham số truyền vào
stream	Làm việc với các luồng dữ liệu một cách hiệu quả, giúp tối ưu hóa việc xử lý dữ liệu lớn
zlib	Nén và giải nén dữ liệu để giảm kích thước file và tăng tốc độ truyền tải

Hình 6: Module NodeJS

## Bảng điều khiển

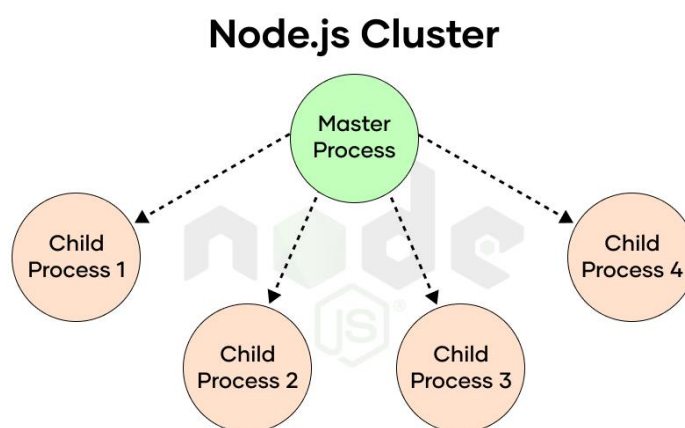
Bảng điều khiển là một module có chức năng hỗ trợ việc gỡ lỗi các ứng dụng JavaScript. Bảng điều khiển này cung cấp một giao diện tương tác để in ra các thông tin quan trọng như giá trị của biến, kết quả của biểu thức và các thông báo lỗi.

Một trong những phương pháp phổ biến để sử dụng bảng điều khiển là sử dụng hàm `console.log()`. Hàm này cho phép in ra bất kỳ giá trị nào lên màn hình console.

## Cluster

NodeJS, nổi tiếng với khả năng lập trình không đồng bộ, thường được xây dựng dựa trên mô hình đơn luồng. Tuy nhiên, để cải thiện hiệu năng và tận dụng tối đa tài nguyên CPU, Cluster là một giải pháp hoàn hảo.

Module Cluster trong NodeJS cho phép bạn tạo ra nhiều tiến trình con (worker processes), mỗi tiến trình sẽ xử lý các yêu cầu đến. Điều này giúp cân bằng tải và ngăn chặn việc một tiến trình bị quá tải. Các worker process này chia sẻ cùng một cổng máy chủ, tạo nên một cụm các tiến trình làm việc cùng nhau.



Hình 7: Cluster NodeJS

## Đối tượng toàn cục

Trong Node.JS các đối tượng toàn cục đóng vai trò vô cùng quan trọng. Đây là những biến hoặc hàm được sẵn có trong mọi module của NodeJS, cho phép các lập trình viên truy cập và sử dụng chúng trực tiếp mà không cần khai báo lại. Một số đối tượng toàn cục phổ biến ở trong bảng dưới đây:

Đối tượng toàn cục	Mô tả
<code>__dirname</code>	Trả về đường dẫn tuyệt đối đến thư mục chứa file đang thực thi.
<code>__filename</code>	Trả về đường dẫn tuyệt đối đến file đang thực thi.
<code>exports</code>	Dùng để xuất các biến hoặc hàm từ module hiện tại.
<code>module</code>	Biểu diễn module hiện tại.
<code>require</code>	Dùng để nhập các module khác vào module hiện tại.

Hình 8: Đối tượng toàn cục NodeJS

## Xử lý lỗi

Trong ứng dụng của Node.JS thường sẽ gặp 4 lỗi phổ biến như sau:

Lỗi JavaScript	Lỗi này xảy ra do cú pháp JavaScript không hợp lệ hoặc lỗi thời gian chạy. Ví dụ bao gồm EvalError, SyntaxError, RangeError, ReferenceError, TypeError và URIError.
Lỗi từ hệ thống	Lỗi này phát sinh từ các vấn đề cấp hệ thống, chẳng hạn như tệp không tồn tại hoặc đóng socket.
Lỗi từ người dùng	Lỗi này do logic hoặc đầu vào được xác định bởi người dùng gây ra.
Lỗi assertion	Lỗi này cho thấy vi phạm các điều kiện hoặc logic mong đợi.

Hình 9: Xử lý lỗi NodeJS

## Streaming (Luồng)

Luồng (Streaming) là một dòng dữ liệu liên tục được truyền từ nguồn đến đích. Trong lập trình, luồng được sử dụng để đọc hoặc ghi dữ liệu vào các thiết bị ngoại vi như file, mạng, hoặc các bộ nhớ đệm (buffer).

Có 4 loại luồng chính:

1. **Luồng đọc:** Chỉ cho phép đọc dữ liệu từ nguồn.
2. **Luồng ghi:** Chỉ cho phép ghi dữ liệu vào đích.
3. **Luồng duplex:** Cho phép vừa đọc vừa ghi dữ liệu.
4. **Luồng chuyển đổi:** Thay đổi định dạng hoặc mã hóa dữ liệu trong quá trình truyền.

### Buffer (Bộ nhớ đệm)

Buffer là một cấu trúc dữ liệu được sử dụng để lưu trữ và thao tác hiệu quả với dữ liệu nhị phân. Buffer cung cấp một cách linh hoạt để làm việc với dữ liệu thô trong bộ nhớ.

### Tên miền (Domain)

Module tên miền giúp ngăn chặn và khắc phục các lỗi tiềm ẩn trong website của bạn. Module này sử dụng hai cơ chế chính để đảm bảo hoạt động ổn định của hệ thống:

- **Liên kết nội bộ:** Bộ phát lỗi sẽ được tích hợp trực tiếp vào mã nguồn của website, giúp xác định và xử lý lỗi một cách nhanh chóng và hiệu quả.
- **Liên kết bên ngoài:** Bạn có thể thêm thủ công các bộ phát lỗi vào tên miền thông qua cấu hình hệ thống. Điều này đặc biệt hữu ích khi bạn muốn theo dõi và quản lý các lỗi từ các nguồn bên ngoài.

### DNS

Module DNS dùng để tương tác với các máy chủ Hệ thống tên miền (DNS). Module DNS cho phép các nhà phát triển hiệu quả phân giải tên miền thành địa chỉ IP.

Có hai phương pháp chính để phân giải DNS:

1. `dns.resolve()`: Phương pháp này thiết lập kết nối mạng với máy chủ DNS để thực hiện quá trình phân giải. Phù hợp cho các ứng dụng yêu cầu tra cứu DNS thời gian thực hoặc cần xử lý các truy vấn DNS phức tạp.
2. `dns.lookup()`: Phương pháp này thực hiện phân giải DNS mà không cần kết nối mạng. Phù hợp cho các trường hợp kết nối mạng bị hạn chế hoặc không đáng tin cậy, chẳng hạn như ứng dụng ngoại tuyến hoặc cơ chế lưu trữ cache.

### **Debugger (Trình gỡ lỗi)**

Node.JS có công cụ gỡ lỗi tích hợp vô cùng tiện lợi, giúp các nhà phát triển kiểm tra và khắc phục lỗi trong mã nguồn một cách hiệu quả. Mặc dù không sở hữu nhiều tính năng phức tạp như các trình gỡ lỗi chuyên dụng, nhưng công cụ này hoàn toàn đáp ứng nhu cầu kiểm tra code đơn giản trong quá trình phát triển ứng dụng.

#### **2.1.2.3. Ưu và nhược điểm của NodeJS**

##### **Ưu điểm:**

- IO hướng sự kiện không đồng bộ, cho phép xử lý nhiều yêu cầu đồng thời.
- Sử dụng JavaScript – một ngôn ngữ lập trình dễ học.
- Chia sẻ cùng code ở cả phía client và server.
- NPM(Node Package Manager) và module Node đang ngày càng phát triển mạnh mẽ.
- Cộng đồng hỗ trợ tích cực.
- Cho phép stream các file có kích thước lớn.

##### **Nhược điểm:**

- Không có khả năng mở rộng, vì vậy không thể tận dụng lợi thế mô hình đa lõi trong các phần cứng cấp server hiện nay.
- Khó thao tác với cơ sở dữ liệu quan hệ.
- Mỗi callback sẽ đi kèm với rất nhiều callback lồng nhau khác.
- Cần có kiến thức tốt về JavaScript.
- Không phù hợp với các tác vụ đòi hỏi nhiều CPU.



### 2.1.3. Giới thiệu về MySQL



*Hình 10: Logo MySQL*

#### 2.1.3.1. MySQL là gì?

MySQL là một hệ quản trị cơ sở dữ liệu quan hệ (RDBMS - Relational Database Management System) mã nguồn mở, được phát triển bởi công ty MySQL AB và hiện thuộc sở hữu của Oracle Corporation. MySQL thường được sử dụng trong các ứng dụng web và là một phần của ngăn xếp LAMP (Linux, Apache, MySQL, PHP/Perl/Python).

#### 2.1.3.2. Các đặc điểm chính của MySQL

- Mã nguồn mở: MySQL miễn phí sử dụng, nhưng cũng cung cấp các phiên bản thương mại với tính năng nâng cao.
- Kiến trúc quan hệ: Dữ liệu được lưu trữ dưới dạng bảng (tables), với các hàng (rows) và cột (columns), hỗ trợ chuẩn hóa dữ liệu.
- Hiệu suất cao: MySQL được thiết kế tối ưu hóa cho các ứng dụng có lượng truy vấn lớn và yêu cầu tốc độ xử lý nhanh.
- Đa nền tảng: MySQL có thể chạy trên nhiều hệ điều hành như Windows, Linux, macOS, và Unix.
- Hỗ trợ đa người dùng: Nhiều người dùng có thể kết nối và làm việc với cơ sở dữ liệu cùng lúc mà không gây xung đột.
- Tương thích tốt: MySQL dễ dàng tích hợp với nhiều ngôn ngữ lập trình phổ biến như PHP, Python, Java, .NET, và các framework web như Laravel, Django, Spring.

- Bảo mật cao: Cung cấp hệ thống quản lý người dùng, mật khẩu, mã hóa SSL và hỗ trợ các quyền truy cập tùy chỉnh.

### 2.1.3.3. Ưu và nhược điểm của MySQL

#### **Ưu điểm:**

- MySQL là phần mềm mã nguồn mở, có thể sử dụng miễn phí, tiết kiệm chi phí cho các cá nhân và doanh nghiệp.
- MySQL được tối ưu hóa để xử lý khối lượng dữ liệu lớn và lượng truy vấn đồng thời cao. Điều này làm cho MySQL phù hợp với các ứng dụng web và doanh nghiệp.
- Giao diện thân thiện, cú pháp SQL đơn giản, dễ học và triển khai, phù hợp cho cả người mới bắt đầu và chuyên gia.
- MySQL có thể chạy trên nhiều hệ điều hành như Windows, Linux, macOS, và Unix, giúp nó linh hoạt trong triển khai.
- Hỗ trợ nhiều ngôn ngữ lập trình phổ biến như PHP, Python, Java, .NET, và các framework như Laravel, Django, Spring.
- Cho phép lựa chọn các bộ máy lưu trữ (InnoDB, MyISAM, Memory, CSV) phù hợp với nhu cầu sử dụng cụ thể.
- Cung cấp các cơ chế bảo mật như xác thực người dùng, mã hóa dữ liệu, SSL, và quản lý quyền truy cập.
- Có cộng đồng người dùng đông đảo, tài liệu phong phú, giúp dễ dàng tìm kiếm giải pháp và hỗ trợ kỹ thuật.
- Có thể mở rộng để xử lý cơ sở dữ liệu lớn và hỗ trợ sao chép (Replication) để cải thiện hiệu suất và tính sẵn sàng.

#### **Nhược điểm:**

- Mặc dù InnoDB hỗ trợ giao dịch, nhưng so với các hệ quản trị cơ sở dữ liệu khác như PostgreSQL hay Oracle, MySQL vẫn còn hạn chế trong việc xử lý các giao dịch phức tạp.
- Một số tính năng nâng cao như hỗ trợ cho các loại dữ liệu đặc biệt (JSON, XML) và các công cụ phân tích dữ liệu chưa mạnh mẽ bằng PostgreSQL.

- Khi kích thước cơ sở dữ liệu trở nên rất lớn hoặc số lượng truy vấn đồng thời tăng đáng kể, hiệu suất MySQL có thể giảm.
- MySQL thiếu tính năng kiểm soát chi tiết và chặt chẽ như hệ thống quyền người dùng phức tạp hoặc cơ chế kiểm tra tích hợp.
- Với các hệ thống lớn, việc quản lý sao lưu và phục hồi dữ liệu có thể trở nên phức tạp.
- MySQL không hỗ trợ tốt việc xử lý dữ liệu phân tán và không có công cụ tích hợp mạnh mẽ để thực hiện các nhiệm vụ này.
- Để đạt được hiệu suất tốt nhất, các truy vấn và cấu trúc cơ sở dữ liệu cần được tối ưu hóa thủ công, đòi hỏi kiến thức chuyên sâu.
- Mặc dù MySQL là mã nguồn mở, nhưng các tính năng nâng cao trong phiên bản thương mại (Enterprise Edition) yêu cầu trả phí.

## **2.2. NGHIÊN CỨU TỔNG QUAN NGƯỜI DÙNG**

### **2.2.1. Xây dựng hệ thống tìm đồ thất lạc**

- Mục tiêu: Kết nối người mất đồ và người nhặt được đồ một cách hiệu quả.
- Yếu tố quan trọng: Dễ sử dụng, nhanh chóng, chính xác, bảo mật thông tin cá nhân.
- Lý thuyết nền tảng:
  - + Ứng dụng giao diện tương tác cao (ReactJS) để tạo trải nghiệm người dùng tốt.
  - + Xử lý backend mạnh mẽ (NodeJS) với khả năng phản hồi nhanh.
  - + Sử dụng cơ sở dữ liệu quan hệ (MySQL) để quản lý dữ liệu thông tin đồ vật và người dùng một cách toàn vẹn.

### **2.2.2. Tích hợp công nghệ hiện đại**

- Công nghệ Maps API: Hỗ trợ định vị chính xác vị trí đồ vật thất lạc, giúp người dùng dễ dàng tìm kiếm hoặc chỉ định địa điểm nhặt được đồ.
- Hệ thống thông báo tự động qua email: Cung cấp cập nhật kịp thời, tăng cơ hội tìm thấy đồ vật.

- Tìm kiếm thông minh: Tối ưu hóa các bộ lọc theo thời gian, địa điểm, và loại đồ vật, đồng thời tích hợp nhận diện hình ảnh để cải thiện hiệu suất.

## **2.3. TÍNH BẢO MẬT VÀ KHẢ NĂNG ĐÁP ỨNG CỦA WEBSITE**

### **2.3.1. Tính khả thi**

Website có thể giúp giảm thiểu thời gian và công sức trong việc tìm kiếm đồ thất lạc thông qua các công cụ tìm kiếm hiện đại và hệ thống thông báo tự động.

### **2.3.2. Hiệu quả trong việc tìm kiếm đồ thất lạc**

- Sử dụng các tiêu chí cụ thể (thời gian, địa điểm, loại đồ vật) giúp người dùng nhanh chóng tìm thấy thông tin cần thiết.
- Tích hợp công nghệ bản đồ và tìm kiếm hình ảnh sẽ tăng tỷ lệ thành công trong việc kết nối giữa người mất đồ và người nhặt được.

### **2.3.3. Bảo mật và toàn vẹn dữ liệu**

- Việc áp dụng các cơ chế bảo mật như mã hóa thông tin người dùng, xác thực tài khoản sẽ đảm bảo an toàn cho dữ liệu cá nhân.
- Cơ sở dữ liệu được thiết kế với các mối quan hệ chặt chẽ giúp duy trì tính toàn vẹn dữ liệu và tránh mất mát thông tin.

## **2.4. QUY TRÌNH VÀ PHƯƠNG PHÁP**

### **2.4.1. Nghiên cứu tài liệu**

Nghiên cứu các tài liệu liên quan đến:

- Công nghệ ReactJS, NodeJS, và MySQL.
- Các hệ thống hiện có về tìm kiếm đồ thất lạc.
- Tích hợp công nghệ Google Maps API và thông báo qua email.

### **2.4.2. Thiết kế và triển khai**

Thiết kế: Sử dụng mô hình MVC (Model-View-Controller) để xây dựng hệ thống, đảm bảo phân tách rõ ràng giữa giao diện, xử lý logic, và dữ liệu.

Triển khai:

- Xây dựng các giao diện người dùng bằng ReactJS.

- Phát triển các API backend bằng NodeJS và kết nối với MySQL.
- Tích hợp API Maps và hệ thống thông báo qua email.

#### **2.4.3. Kiểm thử và đánh giá**

- Kiểm thử chức năng: Đảm bảo các tính năng chính như đăng nhập, đăng ký, đăng bài, tìm kiếm, và thông báo hoạt động ổn định.
- Kiểm thử hiệu suất: Đánh giá tốc độ tải trang, khả năng xử lý nhiều yêu cầu đồng thời.
- Đánh giá trải nghiệm người dùng: Thu thập phản hồi từ nhóm người dùng thử nghiệm để cải thiện giao diện và tính năng.

## CHƯƠNG 3: HIỆN THỰC HÓA NGHIÊN CỨU

### 3.1. CHUẨN BỊ NỀN TẢNG NGHIÊN CỨU

#### 3.1.1. Lựa chọn công nghệ

- Front-end: Lựa chọn ReactJS vì tôi có thể sử dụng thư viện Tailwind để thiết kế giao diện dễ dàng hơn và dễ quản lý hơn so với các ngôn ngữ khác phải dùng css thiết kế rất nhiều dòng, trong khi đó thư viện Tailwind chỉ xác định phần nào cần thiết kế thì sẽ viết chung câu lệnh của phần đấy, rất tiện để dễ quản lý
- Back-end: Sử dụng NodeJS với framework ExpressJS có khả năng phản hồi nhanh, dễ dàng quản lý các mã nguồn API
- Cở sở dữ liệu: MySQL rất tiện cho việc quản lý dữ liệu online, có tính năng đảm bảo tính toàn vẹn dữ liệu và dễ dàng tích hợp với NodeJS
- Công nghệ:
  - + Tích hợp API Maps hỗ trợ định vị đồ vật tại điểm nhặt được hoặc đánh rơi.
  - + Gửi thông báo qua mail thì tích hợp thư viện Nodemailer để gửi thông báo đến tất cả cá người dùng trong hệ thống khi có bài đăng mới

#### 3.1.2. Môi trường

- Công cụ lập trình: Sử dụng Visual Studio Code (VS Code) để phát triển mã nguồn, Postman để kiểm thử API, và MySQL Workbench để quản lý cơ sở dữ liệu.
- Cài đặt thư viện và công cụ: Các thư viện chính bao gồm:
  - + Frontend: ReactJS, React Router, TailwindCSS.
  - + Backend: ExpressJS, Nodemailer, bcrypt, jsonwebtoken.
  - + Cơ sở dữ liệu: MySQL với thư viện mysql2 để kết nối từ NodeJS.
- Thiết lập cơ sở dữ liệu: Cấu trúc cơ sở dữ liệu gồm các bảng:
  - + Users: Lưu thông tin người dùng (id, name, email, password, phone, avatar).
  - + Posts: Lưu thông tin đồ thất lạc hoặc nhặt được (id, title, description, category, location, date, user\_id).

## 3.2. QUY TRÌNH THỰC HIỆN NGHIÊN CỨU

### 3.2.1. Giao diện người dùng

- Trang chính: Hiển thị danh sách bài viết về đồ thất lạc và đồ nhặt được, kèm theo bộ lọc theo loại đồ vật, thời gian và địa điểm.
- Trang đăng ký/đăng nhập: Giao diện đơn giản, cho phép người dùng tạo tài khoản hoặc đăng nhập để sử dụng các tính năng.
- Trang đăng bài: Form đăng bài cho phép người dùng nhập tiêu đề, mô tả, chọn loại đồ vật, tải lên hình ảnh và chọn vị trí trên bản đồ.
- Trang quản lý thông tin người dùng: Khu vực cho phép người dùng xem và chỉnh sửa thông tin cá nhân, cũng như theo dõi các bài viết đã đăng.
- Trang quản lý thông tin người dùng và bài đăng của người quản trị: Xem tổng số thành viên có trên hệ thống và tổng số bài đăng mà người dùng đã đăng

### 3.2.2. Chức năng

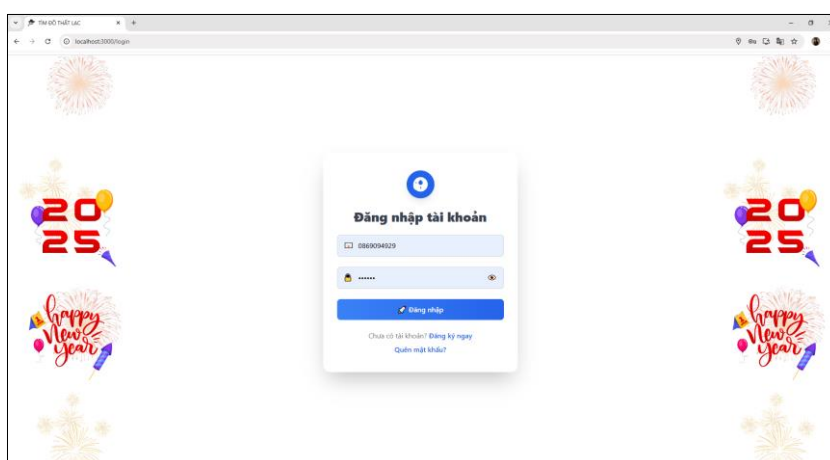
- API đăng ký và đăng nhập:
  - + Người dùng cung cấp email, mật khẩu và thông tin cá nhân để tạo tài khoản. Dữ liệu được mã hóa bằng bcrypt trước khi lưu vào cơ sở dữ liệu.
  - + Khi đăng nhập, hệ thống xác thực bằng jsonwebtoken để tạo mã thông báo truy cập (access token).
- API đăng bài viết: Hỗ trợ người dùng đăng thông tin về đồ thất lạc hoặc nhặt được. Dữ liệu bao gồm tiêu đề, mô tả, hình ảnh, và vị trí được lưu vào bảng Posts.
- API tìm kiếm: Cho phép tìm kiếm bài viết dựa trên các tiêu chí: từ khóa, loại đồ vật, thời gian và địa điểm.
- API thông báo qua email: Sử dụng Nodemailer để gửi email cho người dùng khi có bài viết mới trùng khớp với mô tả đã đăng.

## CHƯƠNG 4: KẾT QUẢ NGHIÊN CỨU

### 4.1. KẾT QUẢ HOÀN THÀNH DỰ ÁN

Dự án Website Tìm Đồ Thất Lạc đã được hoàn thiện và đáp ứng đủ các yêu cầu trong đề tài đã đăng kí. Chức năng chính của website là cung cấp nền tảng để người dùng đăng thông tin về đồ thất lạc và đồ nhặt được, giúp 2 bên người dùng kết nối với nhau dễ dàng

#### 4.1.1. Đăng nhập



Hình 11: Giao diện đăng nhập

Chức năng đăng nhập cho phép người dùng truy cập vào tài khoản cá nhân trên hệ thống. Đây là một trong những chức năng quan trọng đảm bảo tính bảo mật trong việc quản lý thông tin.

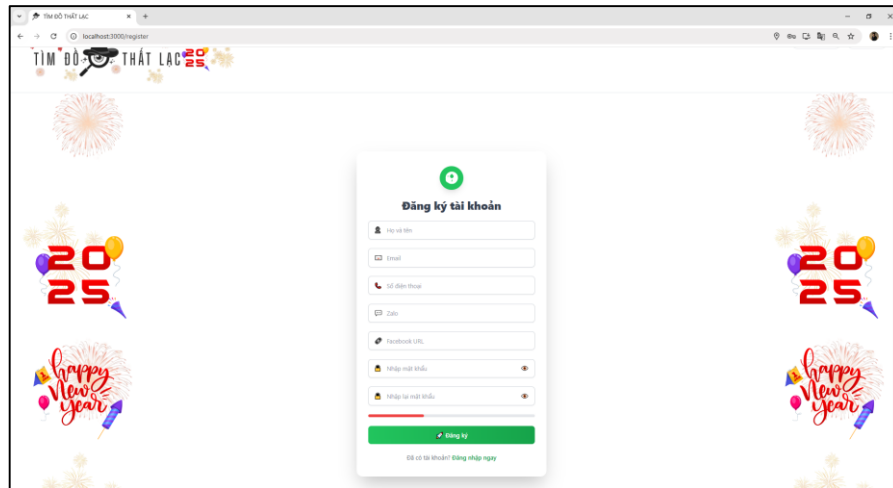
Hệ thống hỗ trợ người dùng đăng nhập bằng email hoặc số điện thoại, và nếu nhập sai thông tin thì sẽ không đăng nhập được và hệ thống sẽ thông báo lỗi chính xác, giúp cho người dùng dễ dàng nhận biết và tiến hành đăng nhập thành công.

Sau khi người dùng đăng nhập thành công thì các thông tin của người dùng sẽ được gửi đến back-end qua API với phương thức POST, back-end sẽ xác thực thông tin đăng nhập và trả về token + tên người dùng.

Phần đăng nhập này dùng chung cho cả người dùng và người quản trị, tức là nếu đăng nhập bằng tài khoản của người dùng thì sẽ chuyển hướng đến giao diện của người dùng còn nếu đăng nhập bằng tài khoản của người quản trị sẽ chuyển hướng đến trang quản trị.



### 4.1.2. Đăng kí

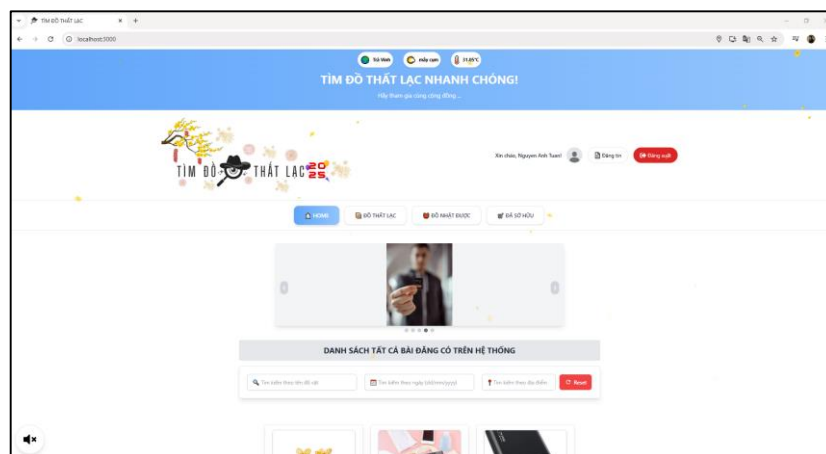


Hình 12: Giao diện đăng kí

Chức năng đăng kí tài khoản giúp người dùng tạo một tài khoản mới trên hệ thống với các thông tin cần thiết như: Họ và tên, Email, Số điện thoại, Zalo, Url Facebook, Mật khẩu và Nhập lại mật khẩu.

Sau khi người dùng nhập đầy đủ thông tin và click “Đăng kí” thì dữ liệu đó sẽ được gửi qua API đăng kí với phương thức POST. Sau khi đăng kí thành công thì người dùng có thể nhập tài khoản vừa đăng kí vào trang đăng nhập và tiếp tục sử dụng hệ thống.

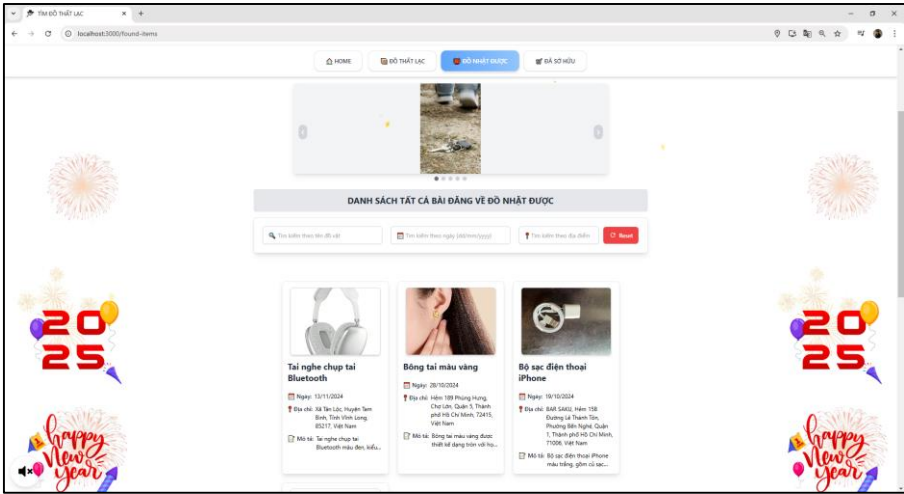
### 4.1.3. Trang chủ



Hình 13: Trang chủ

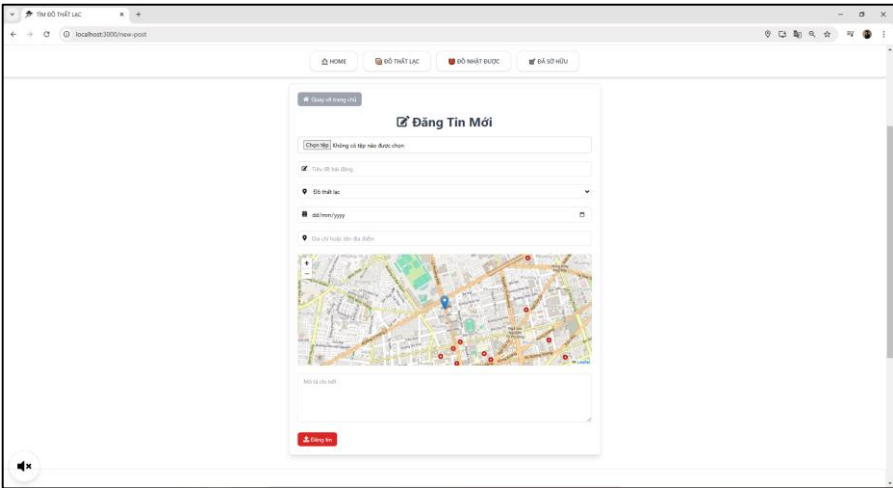
Giao diện trang chủ sẽ hiển thị các tiện ích như “Chỉnh sửa thông tin người dùng”, “Đăng tin mới”, “Chỉnh sửa bài đăng”, và ở thanh điều hướng có các button

điều hướng đến các có dữ liệu trùng khớp với tên của các button. Ví dụ: ở button “ĐỒ NHẬT ĐƯỢC” thì sẽ tải được danh sách các bài đăng về đồ nhặt được.



Hình 14: Giao diện ĐỒ NHẬT ĐƯỢC

#### 4.1.4. Đăng tin mới

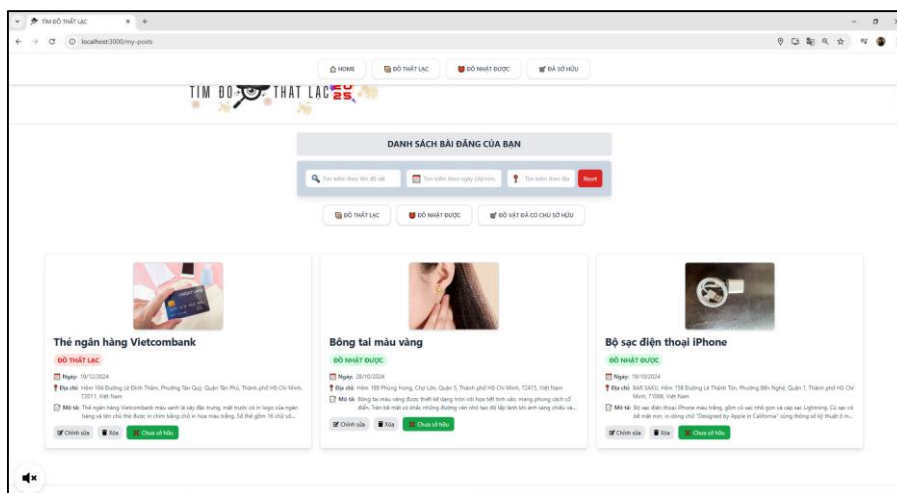


Hình 15: Giao diện ĐĂNG TIN MỚI

Chức năng đăng tin mới cho phép người dùng đăng tải một bài viết mới về đồ nhặt được hoặc đồ thất lạc lên hệ thống với các thông tin như: hình ảnh đồ vật, tên đồ vật, loại đồ nhặt được hoặc đồ thất lạc, ngày nhặt được hoặc ngày thất lạc, địa điểm cụ thể, mô tả chi tiết nhận dạng đồ vật.

Tích hợp bản đồ để người dùng có thể nhấp chuột lên bản đồ để chọn vị trí. Vị trí này được xác định bởi kinh độ, vĩ độ, và tự động lấy địa chỉ qua API Nominatim.

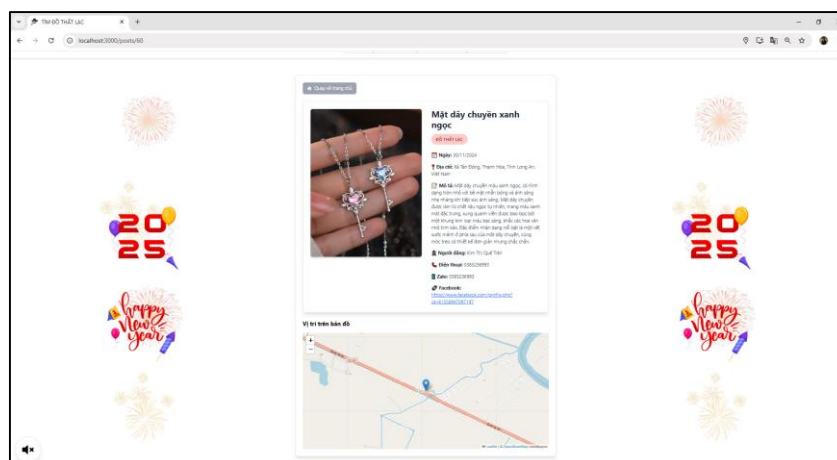
### 4.1.5. Chỉnh sửa bài đăng



Hình 16: Giao diện CHỈNH SỬA BÀI ĐĂNG

Chức năng “Chỉnh sửa bài đăng” cho phép người dùng cập nhật thông tin bài viết đã tạo trước đó hoặc xóa bài viết nếu không cần thiết hoặc đồ đã được chủ nhân tìm thấy.

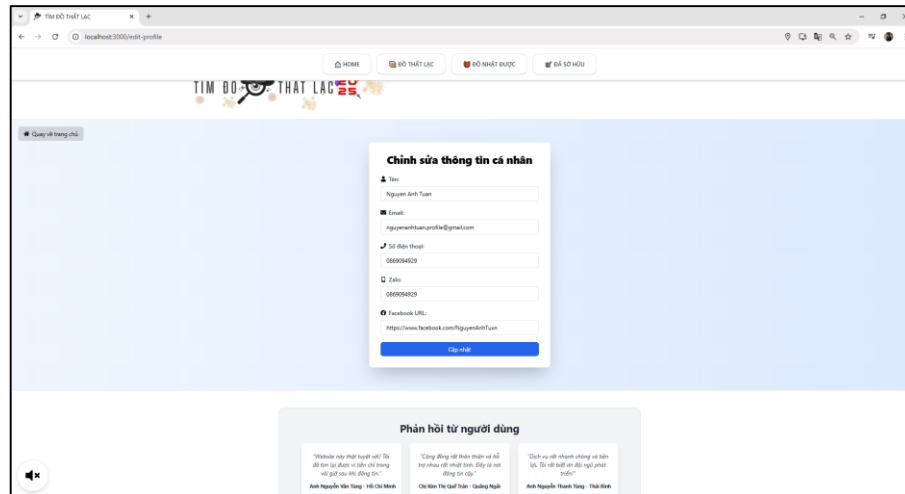
### 4.1.6. Chi tiết bài đăng



Hình 17: Giao diện CHI TIẾT BÀI ĐĂNG

Chức năng “Hiện thị thông tin chi tiết bài đăng” cho phép người dùng xem toàn bộ thông tin chi tiết của một bài đăng cụ thể với những thông tin sau: hình ảnh đồ vật, tên đồ vật, loại bài đăng, ngày, địa điểm, mô tả chi tiết nhận dạng, thông tin người đăng bài.

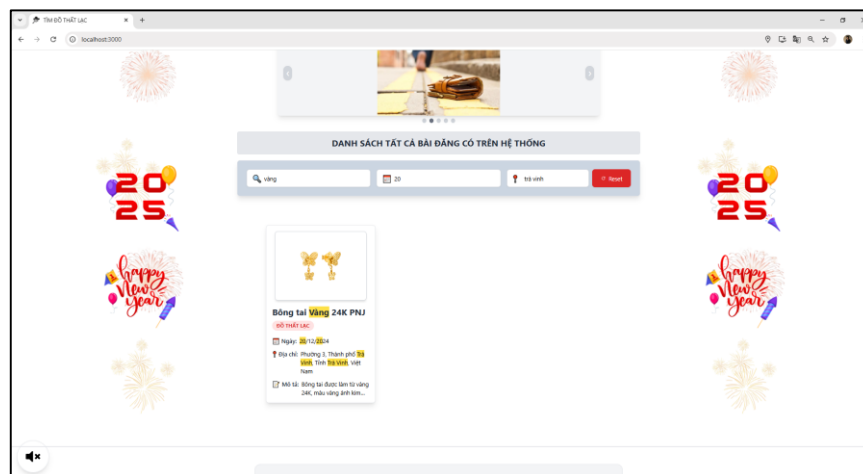
#### 4.1.7. Chỉnh sửa thông tin người dùng



Hình 18: Giao diện CHỈNH SỬA THÔNG TIN NGƯỜI DÙNG

Chức năng “Chỉnh sửa thông tin” cho phép người dùng cập nhật lại thông tin của mình, các thông tin cũng giống như các thông tin đã đăng ký trước đó.

#### 4.1.8. Tìm kiếm thông tin bài đăng



Hình 19: Giao diện TÌM KIẾM THÔNG TIN BÀI ĐĂNG

Chức năng “Tìm kiếm” cho phép người dùng tìm kiếm thông tin của bài đăng theo 3 tiêu chí “Tên đồ vật, ngày, địa điểm” nếu như trong các bài đăng có các giá trị khớp với các giá trị trong ô nhập tìm kiếm thì các giá trị khớp trong bài đăng sẽ được tô vàng cho người dùng dễ dàng nhìn thấy.

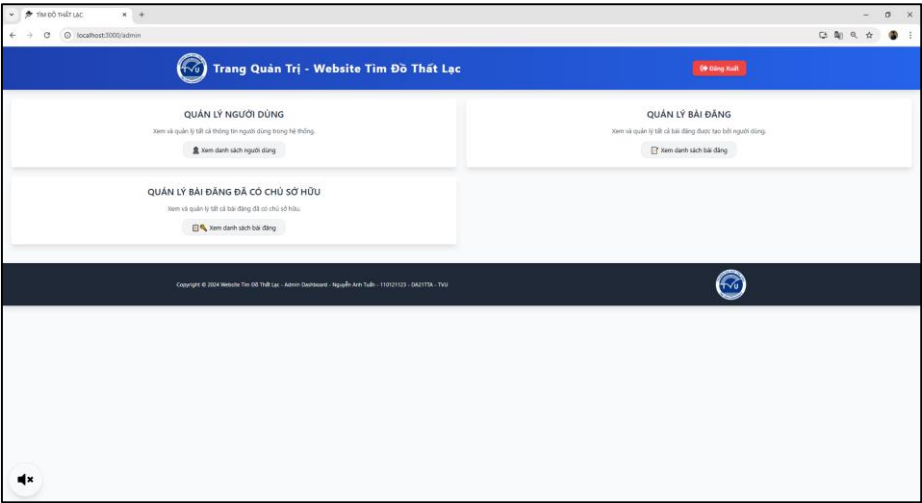
4.1.9. Gửi thông báo quá email khi có bài đăng mới



Hình 20: Thông báo bài đăng mới qua email người dùng

Khi có bài đăng mới trên hệ thống do bất kì người dùng nào đăng tải lên thì bài đăng đó sẽ được thông báo qua email của tất cả người dùng đã đăng kí tài khoản trên hệ thống, điều này giúp cho các người dùng nhận biết được đồ vật nhặt được hoặc thất lạc sớm hơn để có thể xác định được chủ nhân của người bị thất lạc đồ.

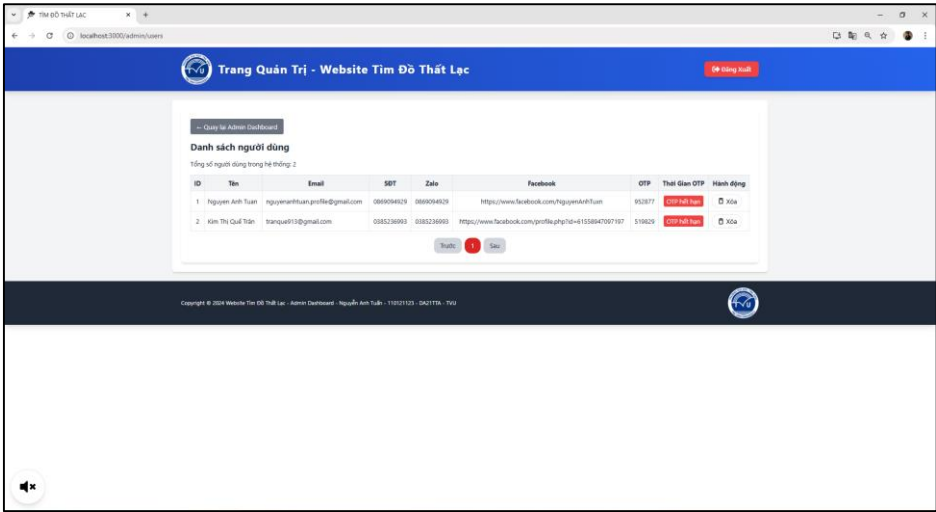
4.1.10. Giao diện của trang quản trị website



Hình 21: Giao diện TRANG QUẢN TRỊ

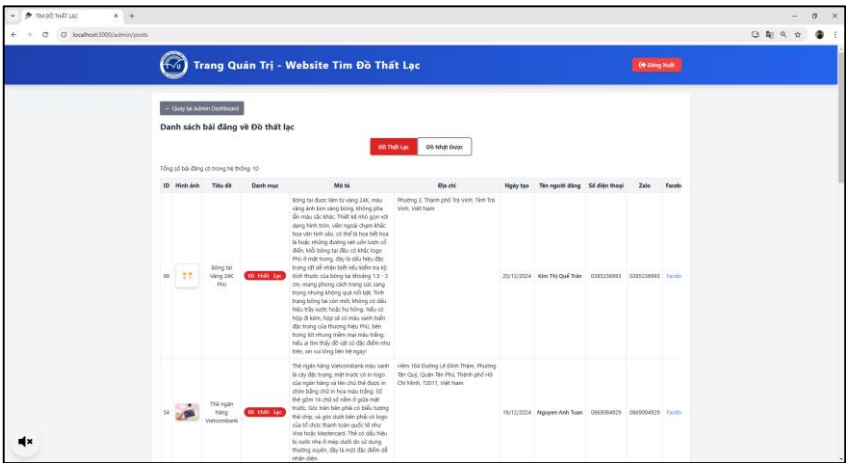
Giao diện trang quản trị gồm các chức năng chính như:

- Quản lý tài khoản người dùng: xem tổng số người dùng trên hệ thống, có quyền xóa người dùng ra khỏi hệ thống khi người dùng vi phạm.



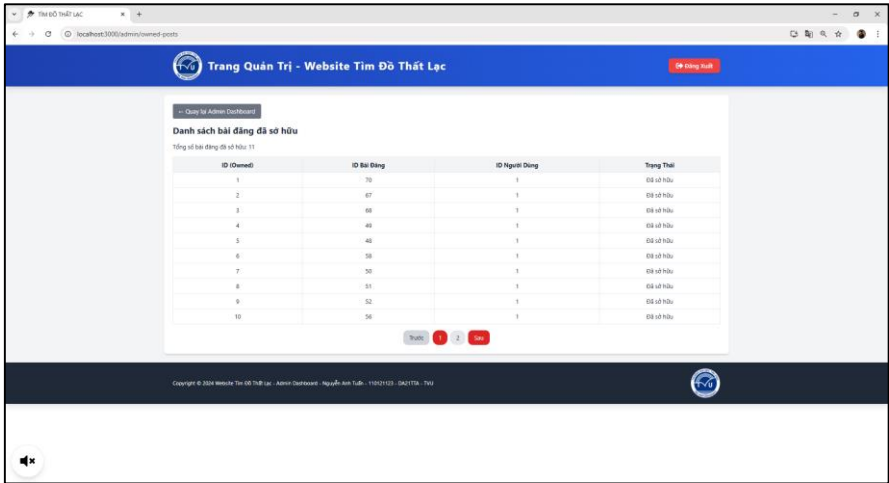
Hình 22: Giao diện QUẢN LÝ TÀI KHOẢN NGƯỜI DÙNG

- Quản lý tài bài đăng có trên hệ thống: xem tổng số bài đăng có trên hệ thống bao gồm các thông tin của bài đăng, có quyền xóa bài đăng ra khỏi hệ thống.



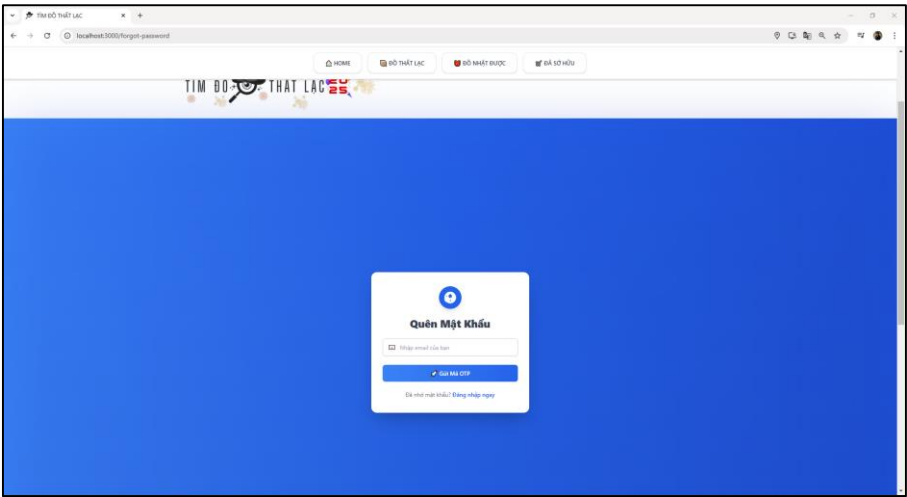
Hình 23: Giao diện QUẢN LÝ BÀI ĐĂNG CÓ TRÊN HỆ THỐNG

- Quản lý tài bài đăng có trên hệ thống: xem tổng số bài đăng đã có chủ sở hữu có trên hệ thống.



Hình 24: Giao diện QUẢN LÝ BÀI ĐĂNG ĐÃ CÓ CHỦ SỞ HỮU

4.1.11. Quên mật khẩu, OTP gửi đến email

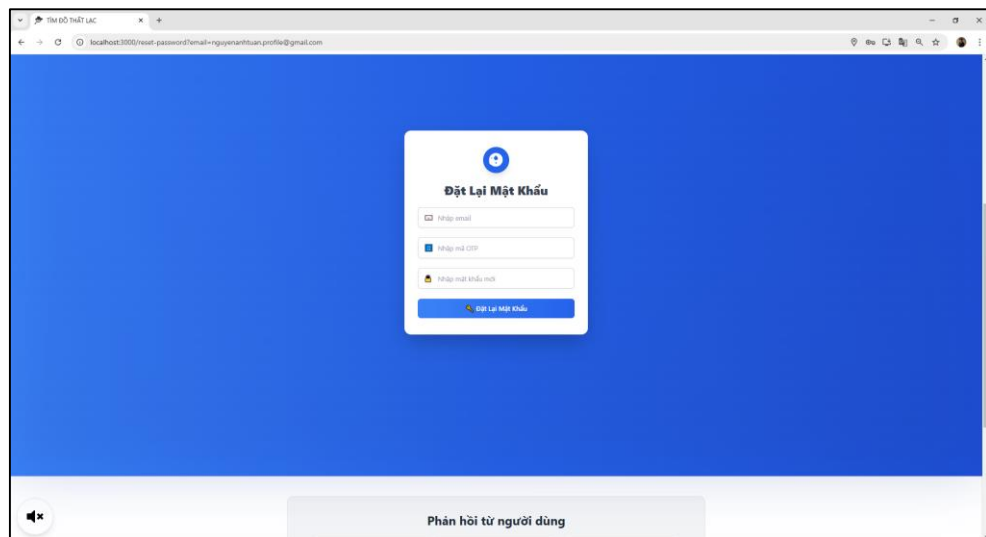


Hình 25: Nhập email yêu cầu gửi OTP

Khi quên mật khẩu người dùng có thể nhập email của mình và yêu cầu gửi mã OTP đến email để đặt lại mật khẩu mới.



Hình 26: OTP gửi về email



Hình 27: Nhập OTP và đặt lại mật khẩu mới

#### 4.1. KIỂM THỬ HỆ THỐNG

- Chức năng: tính năng đăng nhập, đăng kí, đăng bài, tìm kiếm và gửi thông báo qua email hoạt động tốt, không có lỗi.
- Hiệu suất: vì chưa deploy lên các nền tảng công khai nhưng dự án đã chạy khá tốt với một số người dùng nhất định và hệ thống đã có tới 18 bài viết mẫu được đăng tải lên trong ngày.
- Bảo mật: mật khẩu của người dùng thì được mã hóa trước khi đưa vào cơ sở dữ liệu, chỉ có người dùng mới nhớ mật khẩu của mình, còn nếu quên mật khẩu thì có chức năng đặt lại mật khẩu bằng mã OTP gửi qua email của người dùng.



## CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

### 5.1. KẾT LUẬN CHUNG CHO DỰ ÁN

Đề tài “Xây dựng Website Tìm Đồ Thất Lạc” đã hoàn thành các mục tiêu được đặt ra trước đó bao gồm xây dựng hệ thống hỗ trợ kết nối người mất đồ và người nhặt được, giao diện thân thiện và dễ sử dụng, tích hợp thêm các tính năng như tìm kiếm nhanh chóng, định vị bản đồ và thông báo qua email. Website đáp ứng tốt yêu cầu về hiệu quả trong tìm kiếm và bảo mật thông tin người dùng, chính vì thế sẽ triển khai rộng hơn để tạo nên một nền tảng cộng đồng hỗ trợ tìm kiếm đồ thất lạc.

#### 5.1.1. Kết quả đạt được

- Hoàn thiện nền tảng với các tính năng chính như đăng tin, tìm kiếm, thông báo đến người dùng cá thông tin về đồ thất lạc và quản lý người dùng.
- Hệ thống được phát triển với giao diện đơn giản, dễ sử dụng, phù hợp với tất cả mọi người.
- Phía back-end dùng để xử lý các logic nghiệp vụ của hệ thống có khả năng xử lý nhanh, bảo mật thông tin và kết nối với cơ sở dữ liệu mượt mà.

#### 5.1.2. Hạn chế

Mặc dù cũng đã có rất nhiều điều đạt được nhưng về mặt hạn chế thì hệ thống vẫn còn rất nhiều, tiếp tục cố gắng phát triển để hoàn thiện nhất hệ thống để có thể triển khai lớn hơn và rộng rãi hơn nữa.

### 5.1. HƯỚNG PHÁT TRIỂN

#### 5.2.1. Tích hợp AI:

Phát triển thêm tính năng tìm kiếm nâng cao dựa trên nhận diện hình ảnh.

#### 5.2.2. Phát triển ứng dụng di động:

Tạo ứng dụng dành cho IOS và Android, mang đến trải nghiệm tốt hơn trên thiết bị di động.

#### 5.2.3. Hỗ trợ đa ngôn ngữ:

Không chỉ riêng tiếng Việt mà sẽ phát triển thêm tính năng hỗ trợ nhiều ngôn ngữ, mở rộng khả năng sử dụng cho người dùng ở các khu vực khác nhau.

## DANH MỤC TÀI LIỆU THAM KHẢO

1. Vietnix, React JS là gì? Mọi thứ cần biết về phần mềm React JS.
2. Viblo, Giới thiệu về ReactJS - Phần I (Các khái niệm cơ bản).
3. Vietnix, NodeJS là gì? Tổng quan kiến thức về Node.JS.
4. ITViec, NodeJS là gì: Tổng quan kiến thức NodeJS và Top 5 NodeJS framework.