**KHOA KỸ THUẬT VÀ CÔNG NGHỆ**

BỘ MÔN CÔNG NGHỆ THÔNG TIN



**ĐỒ ÁN KẾT THÚC MÔN**

**PHÁT TRIỂN ỨNG DỤNG WEB MÃ NGUỒN MỞ**

**HỌC KỲ I, NĂM HỌC 2024 – 2025**

**XÂY DỰNG WEBSITE**

**QUẢN LÝ MƯỢN TRẢ SÁCH TẠI TRUNG TÂM HỌC LIỆU – PHÁT TRIỂN DẠY VÀ HỌC**

*Sinh viên thực hiện:*

**Nguyễn Anh Tuấn -110121123**

**Cao Duy Nhân - 110121070**

**Nguyễn Minh Hải Đăng - 110121181**

*Lớp:* **DA21TTA**

*Giáo viên hướng dẫn:*

**ThS. Phạm Thị Trúc Mai**

***Trà Vinh, tháng 01 năm 2025***

**KHOA KỸ THUẬT VÀ CÔNG NGHỆ**

BỘ MÔN CÔNG NGHỆ THÔNG TIN



**ĐỒ ÁN CHUYÊN NGHÀNH**

**PHÁT TRIỂN ỨNG DỤNG WEB MÃ NGUỒN MỞ**

**HỌC KỲ I, NĂM HỌC 2024 – 2025**

**XÂY DỰNG WEBSITE**

**QUẢN LÝ MƯỢN TRẢ SÁCH TẠI TRUNG TÂM HỌC LIỆU – PHÁT TRIỂN DẠY VÀ HỌC**

*Sinh viên thực hiện:*

**Nguyễn Anh Tuấn -110121123**

**Cao Duy Nhân - 110121070**

**Nguyễn Minh Hải Đăng - 110121181**

*Lớp:* **DA21TTA**

*Giáo viên hướng dẫn:*

**ThS. Phạm Thị Trúc Mai**

***Trà Vinh, tháng 01 năm 2025***

**NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

*Trà Vinh, ngày ….. tháng …… năm ……*

**Giáo viên hướng dẫn**

*(Ký tên và ghi rõ họ tên)*

**NHẬN XÉT CỦA THÀNH VIÊN HỘI ĐỒNG**

*Trà Vinh, ngày ….. tháng …… năm ……*

**Thành viên hội đồng**

*(Ký tên và ghi rõ họ tên)*

**LỜI CẢM ƠN**

Lời nói đầu tiên cho chúng em xin gửi lời cảm ơn và biết ơn sâu sắc đến Cô hướng dẫn Phạm Thị Trúc Mai đã giúp chúng em hoàn thành đề tài “Xây dựng Website quản lý mượn trả sách tại trung tâm học liệu – phát triển dạy và học” và chúng 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 chúng 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 chúng em chưa cao nên trong đồ án của chúng em không thể tránh khỏi những thiếu sót, chúng em rất mong nhận được sự đón góp ý kiến của các thầy, cô và bạn bè để chúng em có thể hoàn thiện đồ án này tốt hơn.

*Chúng em xin chân thành cảm ơn!*

Sinh viên

**MỤC LỤC**

[MỞ ĐẦU 11](#_Toc186686758)

[Lý do chọn đề tài 11](#_Toc186686759)

[Mục đích 11](#_Toc186686760)

[Đối tượng 12](#_Toc186686761)

[Phạm vi nghiên cứu 12](#_Toc186686762)

[CHƯƠNG 1: TỔNG QUAN 13](#_Toc186686763)

[1.1. MÔ TẢ VẤN ĐỀ 13](#_Toc186686764)

[1.1.1. Tình trạng mất đồ cá nhân phổ biến 13](#_Toc186686765)

[1.1.2. Tác động của việc mất đồ cá nhân 13](#_Toc186686766)

[1.1.3. Các giải pháp tìm đồ thất lạc hiện tại 13](#_Toc186686767)

[1.1.4. Thiếu nền tảng chung để kết nối người nhặt đồ và người đánh rơi đồ 13](#_Toc186686768)

[1.1.5. Thiếu tính năng tìm kiếm hiệu quả và chính xác 14](#_Toc186686769)

[1.1.6. Thiếu tính năng thông báo tự động 14](#_Toc186686770)

[1.1.7. Nhu cầu xây dựng nền tảng tìm đồ thất lạc hiệu quả 14](#_Toc186686771)

[1.1.8. Giải pháp sử dụng công nghệ Web để giải quyết vấn đề 14](#_Toc186686772)

[1.2. MỤC TIÊU NGHIÊN CỨU 15](#_Toc186686773)

[1.2.1. Xây dựng nền tảng kết nối người mất đồ và người nhặt được đồ 15](#_Toc186686774)

[1.2.2. Phát triển công cụ tìm kiếm và quản lý thông tin hiệu quả 15](#_Toc186686775)

[1.2.3. Tích hợp tính năng bản đồ và thông báo qua email 15](#_Toc186686776)

[1.2.4. Đảm bảo bảo mật thông tin người dùng và hiệu suất của hệ thống 16](#_Toc186686777)

[1.3. HƯỚNG TIẾP CẬN 16](#_Toc186686778)

[1.3.1. Phát triển từ nhu cầu thực tế của người dùng 16](#_Toc186686779)

[1.3.2. Xây dựng nền tảng dễ sử dụng và thân thiện với người dùng 16](#_Toc186686780)

[1.3.3. Tích hợp công nghệ hiện đại (Maps API, hệ thống email) 17](#_Toc186686781)

[1.3.4. Đảm bảo tính bảo mật và an toàn thông tin 17](#_Toc186686782)

[1.3.5. Tối ưu hóa hiệu suất và khả năng mở rộng của hệ thống 17](#_Toc186686783)

[1.4. ĐỐI TƯỢNG VÀ PHẠM VI NGHIÊN CỨU 17](#_Toc186686784)

[1.4.1. Đối tượng nghiên cứu 17](#_Toc186686785)

[1.4.2. Phạm vi nghiên cứu 18](#_Toc186686786)

[CHƯƠNG 2: NGHIÊN CỨU LÝ THUYẾT 19](#_Toc186686787)

[2.1. LÝ THUYẾT 19](#_Toc186686788)

[2.1.1. Giới thiệu về ReactJS 19](#_Toc186686789)

[2.1.2. Giới thiệu về NodeJS 25](#_Toc186686790)

[2.1.3. Giới thiệu về MySQL 31](#_Toc186686791)

[2.2. NGHIÊN CỨU TỔNG QUAN NGƯỜI DÙNG 33](#_Toc186686792)

[2.2.1. Xây dựng hệ thống tìm đồ thất lạc 33](#_Toc186686793)

[2.2.2. Tích hợp công nghệ hiện đại 34](#_Toc186686794)

[2.3. TÍNH BẢO MẬT VÀ KHẢ NĂNG ĐÁP ỨNG CỦA WEBSITE 34](#_Toc186686795)

[2.3.1. Tính khả thi 34](#_Toc186686796)

[2.3.2. Hiệu quả trong việc tìm kiếm đồ thất lạc 34](#_Toc186686797)

[2.3.3. Bảo mật và toàn vẹn dữ liệu 34](#_Toc186686798)

[2.4. QUY TRÌNH VÀ PHƯƠNG PHÁP 34](#_Toc186686799)

[2.4.1. Nghiên cứu tài liệu 34](#_Toc186686800)

[2.4.2. Thiết kế và triển khai 35](#_Toc186686801)

[2.4.3. Kiểm thử và đánh giá 35](#_Toc186686802)

[CHƯƠNG 3: HIỆN THỰC HÓA NGHIÊN CỨU 36](#_Toc186686803)

[3.1. CHUẨN BỊ NỀN TẢNG NGHIÊN CỨU 36](#_Toc186686804)

[3.1.1. Lựa chọn công nghệ 36](#_Toc186686805)

[3.1.2. Môi trường 36](#_Toc186686806)

[3.2. QUY TRÌNH THỰC HIỆN NGHIÊN CỨU 37](#_Toc186686807)

[3.2.1. Giao diện người dùng 37](#_Toc186686808)

[3.2.2. Chức năng 37](#_Toc186686809)

[CHƯƠNG 4: KẾT QUẢ NGHIÊN CỨU 38](#_Toc186686810)

[4.1. KẾT QUẢ HOÀN THÀNH DỰ ÁN 38](#_Toc186686811)

[4.1.1. Đăng nhập 38](#_Toc186686812)

[4.1.2. Đăng kí 39](#_Toc186686813)

[4.1.3. Trang chủ 40](#_Toc186686814)

[4.1.4. Đăng tin mới 41](#_Toc186686815)

[4.1.5. Chỉnh sửa bài đăng 41](#_Toc186686816)

[4.1.6. Chi tiết bài đăng 42](#_Toc186686817)

[4.1.7. Chỉnh sửa thông tin người dùng 42](#_Toc186686818)

[4.1.8. Tìm kiếm thông tin bài đăng 43](#_Toc186686819)

[4.1.9. Gửi thông báo quá email khi có bài đăng mới 43](#_Toc186686820)

[4.1.10. Giao diện của trang quản trị website 44](#_Toc186686821)

[4.1.11. Quên mật khẩu, OTP gửi đến email 45](#_Toc186686822)

[4.1. KIỂM THỬ HỆ THỐNG 46](#_Toc186686823)

[CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 48](#_Toc186686824)

[5.1. KẾT LUẬN CHUNG CHO DỰ ÁN 48](#_Toc186686825)

[5.1.1. Kết quả đạt được 48](#_Toc186686826)

[5.1.2. Hạn chế 48](#_Toc186686827)

[5.1. KẾT LUẬN CHUNG CHO DỰ ÁN 48](#_Toc186686828)

[5.2.1. Tích hợp AI: 48](#_Toc186686829)

[5.2.2. Phát triển ứng dụng di động: 48](#_Toc186686830)

[5.2.3. Hỗ trợ đa ngôn ngữ: 49](#_Toc186686831)

**DANH MỤC HÌNH ẢNH – BẢNG BIỂU**

[Hình 1: Logo ReactJS 19](#_Toc186686838)

[Hình 2: Virtual DOM 20](#_Toc186686839)

[Hình 3: JSX 21](#_Toc186686840)

[Hình 4: Component 22](#_Toc186686841)

[Hình 5: Logo NodeJS 25](#_Toc186686842)

[Hình 6: Module NodeJS 26](#_Toc186686843)

[Hình 7: Cluster NodeJS 27](#_Toc186686844)

[Hình 8: Đối tượng toàn cục NodeJS 28](#_Toc186686845)

[Hình 9: Xử lý lỗi NodeJS 28](#_Toc186686846)

[Hình 10: Logo MySQL 31](#_Toc186686847)

[Hình 11: Giao diện đăng nhập 38](#_Toc186686848)

[Hình 12: Giao diện đăng kí 39](#_Toc186686849)

[Hình 13: Trang chủ 40](#_Toc186686850)

[Hình 14: Giao diện ĐỒ NHẶT ĐƯỢC 40](#_Toc186686851)

[Hình 15: Giao diện ĐĂNG TIN MỚI 41](#_Toc186686852)

[Hình 16: Giao diện CHỈNH SỬA BÀI ĐĂNG 41](#_Toc186686853)

[Hình 17: Giao diện CHI TIẾT BÀI ĐĂNG 42](#_Toc186686854)

[Hình 18: Giao diện CHỈNH SỬA THÔNG TIN NGƯỜI DÙNG 42](#_Toc186686855)

[Hình 19: Giao diện TÌM KIẾM THÔNG TIN BÀI ĐĂNG 43](#_Toc186686856)

[Hình 20: Thông báo bài đăng mới qua email người dùng 43](#_Toc186686857)

[Hình 21: Giao diện TRANG QUẢN TRỊ 44](#_Toc186686858)

[Hình 22: Giao diện QUẢN LÝ TÀI KHOẢN NGƯỜI DÙNG 44](#_Toc186686859)

[Hình 23: Giao diện QUẢN LÝ BÀI ĐĂNG CÓ TRÊN HỆ THỐNG 45](#_Toc186686860)

[Hình 24: Giao diện QUẢN LÝ BÀI ĐĂNG ĐÃ CÓ CHỦ SỞ HỮU 45](#_Toc186686861)

[Hình 25: Nhập email yêu cầu gửi OTP 45](#_Toc186686862)

[Hình 26: OTP gửi về email 46](#_Toc186686863)

[Hình 27: Nhập OTP và đặt lại mật khẩu mới 46](#_Toc186686864)

**TÓM TẮT ĐỒ ÁN KẾT THÚC MÔN**

**Những nghiên cứu về đề tài**

Trong quá trình nghiên cứu để thực hiện đồ án, nhóm đã tiến hành nghiên cứu các vấn đề quan trọng:

* Tình trạng khó khăn trong quản lý mượn – trả sách tại các thư viện truyền thống, đặc biệt là tại Trung tâm học liệu – Phát triển dạy và học
* Các hệ thống quản lý thư viện phổ biến hiện nay, tập trung vào ưu điểm và nhược điểm của chúng để nghiên cứu đề tài, áp dụng các yếu tố phù hợp dùng cho nghiên cứu và triển khai vào dự án.
* Ứng dụng công nghệ web trong xây dựng hệ thống quản lý, bao gồm việc sử dựng Reactjs cho front-end, PHP thuần cho back-end, và MySQL để lưu trữ dữ liệu.

**Hướng tiếp cận vấn đề**

**Cách giải quyết vấn đề**

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

# MỞ ĐẦU

## Lý do chọn đề tài

Trong bối cảnh các trường Đại học hiện nay, việc quản lý chặt chẽ thư viện là nhu cầu cấp thiết nhằm đáp ứng sự phát triển không ngừng của công ngh

## Mục đích

## Đối tượng

## Phạm vi nghiên cứu

# CHƯƠNG 1: TỔNG QUAN

## 1.1. MÔ TẢ VẤN ĐỀ

## 1.2. MỤC TIÊU NGHIÊN CỨU

## 1.3. HƯỚNG TIẾP CẬN

## 1.4. ĐỐI TƯỢNG VÀ PHẠM VI NGHIÊN CỨU

### 1.4.1. Đối tượng nghiên cứu

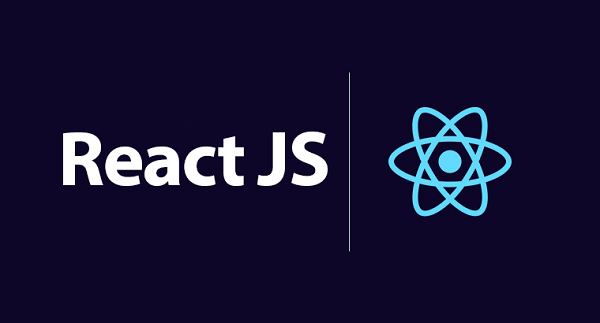
**1.4.2. Phạm vi nghiên cứu**

# 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 javasscript 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.



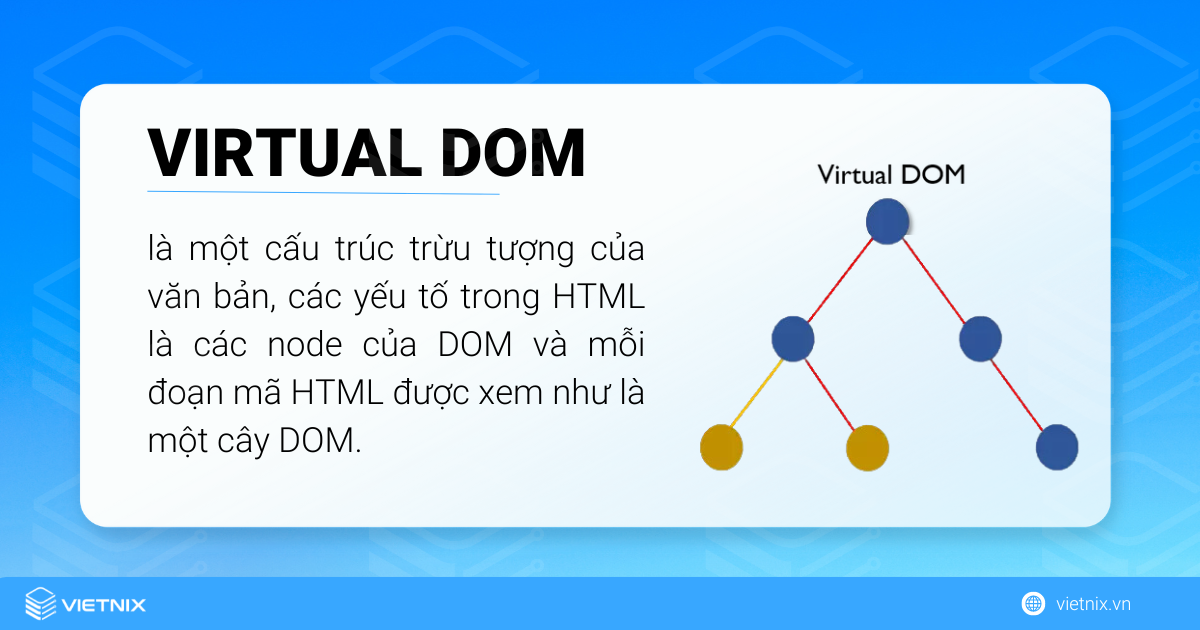
Hình 1: Logo ReactJS

#### . 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. Trươc khi đến cài đặt và cấu hình, chúng ta sẽ đi đến một số khái niệm cơ bản:

#### . Virtual DOM:



Hình 2: Virtual DOM

Document Object Model ([DOM](https://vietnix.vn/dom-la-gi/)) là một cấu trúc trừu tượng của văn bản, các yếu tố trong [HTML](https://vietnix.vn/html-la-gi/) 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.

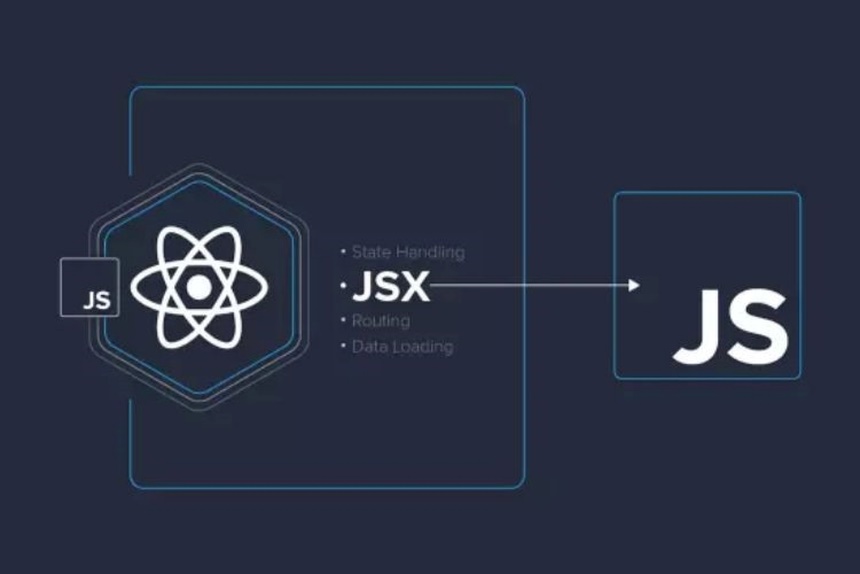
Đối với các trang web lớn, phức tạp, cụ thể là những trang web xử lý nhiều tương tác người dùng, quá trình này sẽ ảnh hưởng đến hiệu suất của trang web ở mức độ rất nghiêm trọng. Lý do là khi người dùng thực hiện một hành động yêu cầu tải lại trang thì toàn bộ cây DOM phải được tải lại, làm ảnh hưởng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

#### . JSX:



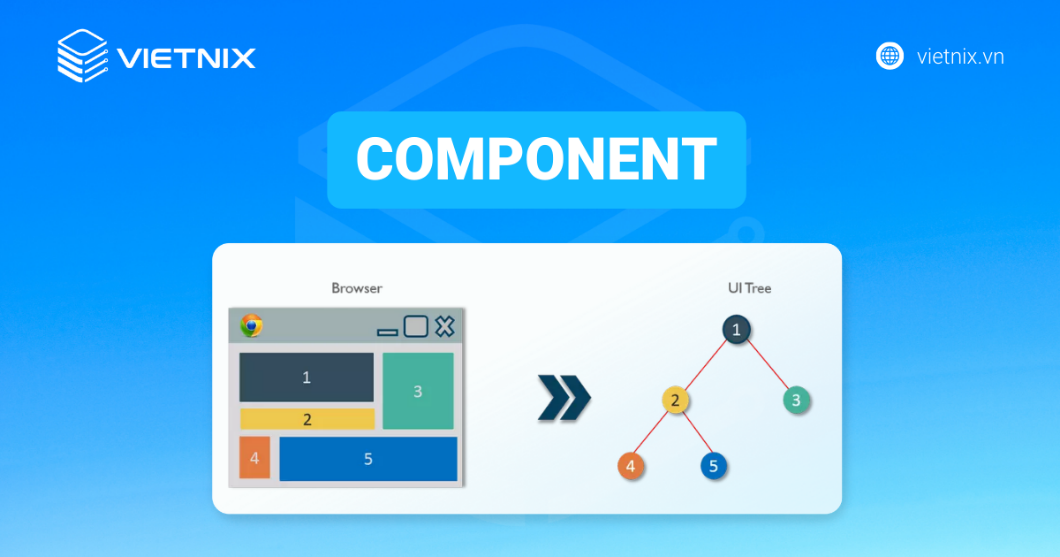
Hình 3: JSX

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](https://vietnix.vn/javascript-la-gi/), [developer](https://vietnix.vn/lap-trinh-vien-la-gi/) 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.

#### . Components:



Hình 4: Component

Với các dự án lớn, [UI](https://vietnix.vn/ui-ux-la-gi/) 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.

#### . 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 getlnitialState() 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 đò 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.

#### . 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 JXS 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](https://vietnix.vn/debug-la-gi/) 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](https://vietnix.vn/angular-la-gi/) 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 PHP

#### 2.1.2.1. PHP là gì?

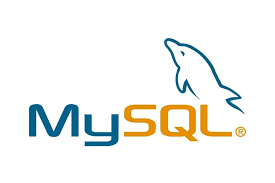
#### 2.1.2.2. Các thành phần của PHP

#### 2.1.2.3. Ưu và nhược điểm của PHP

**Ưu điểm:**

**Nhược điểm:**

### 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 việ quản lý dữ liệu onlin, 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

### 3.2.2. Chức năng

# CHƯƠNG 4: KẾT QUẢ NGHIÊN CỨU

## 4.1. KẾT QUẢ HOÀN THÀNH DỰ ÁN

## 4.1. KIỂM THỬ HỆ THỐNG

# CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## 5.1. KẾT LUẬN CHUNG CHO DỰ ÁN

### 5.1.1. Kết quả đạt được

### 5.1.2. Hạn chế

## 5.1. KẾT LUẬN CHUNG CHO DỰ ÁN