# BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC CẦN THƠ

TRƯỜNG CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG

Logo

Description automatically generated

LUẬN VĂN TỐT NGHIỆP ĐẠI HỌC

NGÀNH CÔNG NGHỆ THÔNG TIN

Đề tài

XÂY DỰNG WEBSITE QUẢN LÝ HOẠT ĐỘNG  
CỦA MỘT NÔNG TRẠI

BUILDING A WEBSITE  
MANAGE ACTIVITIES IN A FARM

Sinh viên: Phan Bảo Luân

Mã số: B2111851

Khóa: K47

Cần Thơ, 11/2024

# BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC CẦN THƠ

TRƯỜNG CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG

KHOA CÔNG NGHỆ THÔNG TIN

Logo

Description automatically generated

LUẬN VĂN TỐT NGHIỆP ĐẠI HỌC

NGÀNH CÔNG NGHỆ THÔNG TIN

Đề tài

XÂY DỰNG WEBSITE QUẢN LÝ HOẠT ĐỘNG  
CỦA MỘT NÔNG TRẠI

BUILDING A WEBSITE  
MANAGE ACTIVITIES IN A FARM

Người hướng dẫn Sinh viên thực hiện

ThS/TS. Trần Công Án Phan Bảo Luân

Mã số: B2111851

Khóa: K47

Cần Thơ, 11/2024

# Lời cảm ơn

Để hoàn thành Luận văn tốt nghiệp ngành Công nghệ thông tin, đầu tiên em gửi lời cảm ơn chân thành nhất đến tất cả giảng viên của Trường Công nghệ Thông tin và Truyền thông đã tận tình giảng dạy, chỉ bảo những kiến thức cơ bản đến chuyên sâu về ngành để hôm nay em có thể hoàn thành được luận văn.

Đặc biệt, em gửi đến thầy TS. Trần Công Án đã giúp đỡ em trong quá trình làm đề tài, đã cung cấp rất nhiều kiến thức, ý tưởng để em có thể hoàn thành tốt bài luận văn.

Dù hoàn thành bài luận văn nhưng sẽ khó tránh khỏi sai sót. Mong quý thầy cô tận tình chỉ điểm, góp ý chân thành.

Em chân thành cảm ơn.

Cần Thơ, ngày 1 tháng 11 năm 2024

Sinh viên thực hiện

Phan Bảo Luân

# Mục lục

# Danh mục hình ảnh

# Danh mục bảng

# Tóm lượt

# Phần giới thiệu

#### Đặt vấn đề

* 1. Vị trí quan trọng của ngành trồng trọt

Trồng trọt là một trong những ngành kinh tế chính của Việt Nam, đóng góp lớn vào GDP quốc gia và tạo công ăn việc làm cho hàng triệu nông dân. Việt Nam là nước có nhiều loại cây trồng chủ lực như lúa gạo, cà phê, trái cây nhiệt đới (như xoài, thanh long, chuối), cùng nhiều cây công nghiệp như cao su, hồ tiêu và điều. Nhờ vào sự đa dạng sinh thái và khí hậu nhiệt đới gió mùa, Việt Nam có điều kiện thuận lợi cho sản xuất nông nghiệp.

* 1. Các thách thức trong trồng trọt
* Biến đổi khí hậu: Tình trạng biến đổi khí hậu ngày càng phức tạp đã ảnh hưởng nghiêm trọng đến năng suất và chất lượng cây trồng. Thời tiết thất thường, hạn hán kéo dài, và lũ lụt diễn ra thường xuyên khiến người nông dân khó có thể đảm bảo được mùa màng ổn định.
* Quản lý thủ công kém hiệu quả: Phần lớn các hoạt động trồng trọt hiện nay vẫn dựa vào phương pháp truyền thống và quản lý thủ công. Việc ghi chép bằng giấy tờ, theo dõi thông tin sản xuất chưa khoa học, và thiếu các công cụ số để phân tích, dự báo đã làm giảm hiệu quả quản lý và khó khăn trong việc truy xuất nguồn gốc sản phẩm.
* Thiếu thông tin cập nhật về thị trường và kỹ thuật: Người nông dân còn hạn chế trong việc tiếp cận thông tin về giá cả thị trường, kỹ thuật canh tác hiện đại, và xu hướng nhu cầu từ người tiêu dùng. Điều này dẫn đến tình trạng sản xuất dư thừa hoặc thiếu hụt, ảnh hưởng trực tiếp đến thu nhập của người dân.
  1. Xu hướng chuyển đổi số trong nông nghiệp

Với sự phát triển của công nghệ, Việt Nam cũng đang dần tiếp cận các phương pháp trồng trọt hiện đại thông qua áp dụng công nghệ số vào sản xuất. Xu hướng chuyển đổi số và nông nghiệp 4.0 đang được khuyến khích, với mục tiêu áp dụng các công nghệ như IoT, dữ liệu lớn, và trí tuệ nhân tạo vào quản lý cây trồng, phân tích đất đai và dự đoán thời tiết.

* 1. Lợi ích của việc số hóa quản lý trồng trọt
* Tối ưu hóa sản xuất: Ứng dụng công nghệ trong quản lý trang trại sẽ giúp tối ưu hóa quy trình chăm sóc cây trồng, từ tưới tiêu, phân bón đến dự báo thời tiết.
* Tăng cường khả năng truy xuất nguồn gốc: Khi áp dụng công nghệ, các hoạt động trồng trọt có thể được lưu trữ một cách hệ thống, giúp việc truy xuất nguồn gốc sản phẩm dễ dàng và minh bạch hơn.
* Giảm chi phí và tăng năng suất: Các hệ thống quản lý trồng trọt tự động hoặc bán tự động sẽ giúp người nông dân tiết kiệm được chi phí, tăng năng suất và nâng cao chất lượng sản phẩm.
  1. Mục tiêu hướng tới của đề tài

Với bối cảnh trên, đề tài xây dựng website quản lý trang trại trồng trọt nhằm hỗ trợ người nông dân và các hợp tác xã nông nghiệp tối ưu hóa hoạt động sản xuất, quản lý cây trồng một cách có hệ thống, và có thể theo dõi sát sao các yếu tố ảnh hưởng đến cây trồng. Website sẽ tạo ra một giải pháp giúp nâng cao năng lực quản lý, tối ưu hoá nguồn tài nguyên, và đáp ứng yêu cầu hiện đại hóa ngành trồng trọt ở Việt Nam

#### Những nghiên cứu liên quan

* 1. Giới thiệu về FaceFarm: FaceFarm là một website quản lý trang trại hỗ trợ nông dân trong việc tổ chức, giám sát các hoạt động canh tác, và đảm bảo truy xuất nguồn gốc sản phẩm.
  2. Các chức năng chính của FaceFarm:

Quản lý đất đai bằng bản đồ

Lập kế hoạch sản xuất và theo dõi tiến độ

Quản lý sử dụng nông dược

Theo dõi thời gian làm việc của nhân công

Truy xuất nguồn gốc sản phẩm qua mã QR

* 1. Sự bổ sung của website đang thực hiện: Website của tôi đáp ứng hầu hết các chức năng chính của FaceFarm, đồng thời bổ sung thêm chức năng chat để hỗ trợ tương tác và quản lý thông tin về đất chi tiết hơn, giúp tối ưu hóa quy trình canh tác.

#### Mục tiêu đề tài

Xây dựng một hệ thống quản lý trang trại: Mục tiêu chính của đề tài là phát triển một website quản lý trang trại, giúp nông dân và các nhà quản lý dễ dàng theo dõi và điều hành các hoạt động canh tác.

Đảm bảo tích hợp các chức năng cần thiết: Website sẽ hỗ trợ các tính năng quan trọng như quản lý đất đai, lập kế hoạch sản xuất, ghi chép thông tin canh tác và theo dõi tiến độ, tương tự như hệ thống của FaceFarm.

Bổ sung các tính năng mở rộng: Hệ thống sẽ có thêm chức năng chat để tăng cường giao tiếp giữa các bên liên quan, cũng như quản lý thông tin chi tiết về đất đai, nhằm nâng cao hiệu quả quản lý và tối ưu hóa sản xuất.

Ứng dụng công nghệ số trong nông nghiệp: Nghiên cứu này hướng đến việc hiện đại hóa các quy trình nông nghiệp thông qua số hóa, từ đó cải thiện năng suất, tiết kiệm chi phí và tạo ra một mô hình quản lý hiệu quả và linh hoạt hơn.

#### Đối tượng và phạm vi nghiên cứu

Đối tượng nghiên cứu: Đối tượng của đề tài là các hệ thống quản lý trang trại và các công nghệ liên quan đến việc giám sát, ghi chép và quản lý các hoạt động canh tác trong nông nghiệp, đặc biệt là các hệ thống quản lý trang trại như FaceFarm và các giải pháp trong nước.

Phạm vi nghiên cứu:

Phạm vi nội dung: Tập trung vào việc xây dựng website quản lý trang trại với các chức năng như quản lý đất đai, nhật ký canh tác, giám sát cây trồng, quản lý nhân công và bổ sung các chức năng mở rộng như chat và quản lý thông tin chi tiết về đất.

Phạm vi địa lý: Hướng đến phục vụ các nông trại trong nước, đồng thời xem xét khả năng áp dụng cho các quy mô trang trại khác nhau.

Phạm vi công nghệ: Áp dụng các công nghệ web hiện đại, đảm bảo khả năng mở rộng và dễ sử dụng cho người dùng.

#### Phương pháp nghiên cứu

Phương pháp nghiên cứu tài liệu: Tìm hiểu các tài liệu và hệ thống quản lý trang trại hiện có để rút ra các chức năng cần thiết và các xu hướng công nghệ phù hợp, như FaceFarm và các nền tảng quốc tế khác.

Phương pháp phân tích và thiết kế hệ thống: Sử dụng phương pháp phân tích hệ thống để xác định yêu cầu và thiết kế kiến trúc tổng thể của website quản lý trang trại.

#### Nội dung nghiên cứu

Nội dung nghiến cứu để xây dựng website quản lý hoạt động của nông trại bao gồm:

1. Phân tích yêu cầu

Quản lý đất, lập được kế hoạch sản suất, theo dõi tiến độ

1. Thiết kế hệ thống
2. Cài đặt và lập trình
3. Kiểm thử và đánh giá

#### Bố cục của quyển luận văn

# Phần nội dung

### Chương I: Đặc tả yêu cầu

#### Mô tả bài toán

#### Yêu cầu bài toán đặt ra

### Chương II: Thiết kế giải pháp

#### Thiết kế cở sở dữ liệu

* 1. Quản lý dữ liệu danh mục

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên** | **Kiểu** | **Diễn giải** |
| 1 | id | string | Mã đơn vị |
| 2 | name | string | Tên đơn vị |
| 3 | description | string | Mô tả đơn vị |
| 4 | type | enum | Kiểu đơn vị |

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên** | **Kiểu** | **Diễn giải** |
| 1 | id | string | Mã đơn vị thực |
| 2 | value | float | Giá trị thực |
| 3 | unitId | string | Mã đơn vị(Unit) |

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên** | **Kiểu** | **Diễn giải** |
| 1 | id | string | Mã đơn vị nguyên |
| 2 | value | int | Giá trị nguyên |
| 3 | unitId | string | Mã đơn vị(Unit) |

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên** | **Kiểu** | **Diễn giải** |
| 1 | id | string | Mã thể loại |
| 2 | name | string | Tên thể loại |
| 3 | description | string | Mô tả thể loại |
| 4 | type | enum | Kiểu thể loại |
| 5 | slug | string | Slug |

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên** | **Kiểu** | **Diễn giải** |
| 1 | id | string | Mã công việc |
| 2 | name | string | Tên công việc |
| 3 | description | string | Mô tả công việc |
| 4 | requirement | string | Yêu cầu công việc |
| 5 | workingTime | string | Giờ làm việc |
| 6 | wage | string | Mức lương |
| 7 | slug | string | Slug |
| 8 | quantity | int | Số lượng tuyển |
| 9 | experience | enum | Kinh nghiệm làm việc |
| 10 | gender | enum | Giới tính |
| 11 | workingState | enum | Part time/ Full time |
| 12 | published | boolean | Có công khai |
| 13 | expiredAt | date | Ngày hết hạn |
| 14 | createdAt | date | Ngày tạo |
| 15 | updatedAt | date | Ngày cập nhật |

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên** | **Kiểu** | **Diễn giải** |
| 1 | id | string | Mã ứng viên |
| 2 | name | string | Tên ứng viên |
| 3 | email | string | Email ứng viên |
| 4 | phone | string | Số điện thoại ứng viên |
| 5 | address | string | Địa chỉ ứng viên |
| 6 | jobId | string | Mã công việc ứng truyển |
| 7 | status | enum | Trạng thái ứng tuyển(mới, đã xác nhận) |

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên** | **Kiểu** | **Diễn giải** |
| 1 | id | string | Mã nhân viên |
| 2 | externalId | string | Mã người dùng(bên thứ 3 cung cấp) |
| 3 | name | string | Tên nhân viên |
| 4 | email | string | Email nhân viên |
| 5 | imageUrl | string | Hình ảnh |
| 6 | phone | string | Số điện thoại |
| 7 | address | string | Địa chỉ |
| 8 | role | enum | Vai trò(superadmin, admin, farmer) |
| 9 | baseHourlyWage | float | Lương cơ bản theo giờ |
| 10 | createdAt | date | Ngày tạo |
| 11 | updatedAt | date | Ngày cập nhật |

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên** | **Kiểu** | **Diễn giải** |
| 1 | id | string | Mã đất trồng – cơ sở |
| 2 | name | string | Tên mảnh đất |
| 3 | location | string | Vị trí mảnh đất |
| 4 | orgId | string | Mã tổ chức quản lý |
| 5 | area | float | Diện tích |
| 6 | shape | string | Hình dạng mảnh đất |
| 7 | soilType | enum | Loại đất |
| 8 | note | string | Ghi chú |
| 9 | createdAt | date | Ngày tạo |
| 10 | updatedAt | date | Ngày cập nhật |

Bảng Field

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên** | **Kiểu** | **Diễn giải** |
| 1 | id | string | Mã thời tiết |
| 2 | temparatureId | string | Mã nhiệt độ(FloatUnit) |
| 3 | humidityId | string | Mã độ ẩm(IntUnit) |
| 4 | atmosphericPressureId | string | Mã áp suất không khí(FloatUnit) |
| 5 | rainfallId | string | Mã lượng mưa(IntUnit) |
| 6 | status | enum | Trạng thái thời tiết |
| 7 | fielđId | string | Mã đất trồng(Field) |
| 8 | confirmed | boolean | Đã xác nhận |
| 9 | ConfirmedAt | date | Xác nhận khi |
| 10 | ConfỉmedById | string | Xác nhận bởi(Staff) |
| 11 | pinned | boolean | Được ghim |
| 12 | note | string | Ghi chú |
| 13 | createdAt | date | Ngày tạo |
| 14 | updatedAt | date | Ngày cập nhật |

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên** | **Kiểu** | **Diễn giải** |
| 1 | id | string | Mã đất quan sát |
| 2 | ph | float | Độ PH đất |
| 3 | moistureId | string | Độ ẩm đất(IntUnit) |
| 4 | nutrientNitrogen | float | Hàm lượng nitơ |
| 5 | nutrientPhosphorus | float | Hàm lượng photpho |
| 6 | nutrientPotassium | float | Hàm lượng Kali |
| 7 | nutrientUnitId | string | Đơn vị hàm lượng(Unit) |
| 8 | fielđId | string | Mã đất trồng(Field) |
| 9 | confirmed | boolean | Đã xác nhận |
| 10 | confirmedAt | date | Xác nhận khi |
| 11 | confỉmedById | string | Xác nhận bởi(Staff) |
| 12 | pinned | boolean | Được ghim |
| 13 | note | string | Ghi chú |
| 14 | createdAt | date | Ngày tạo |
| 15 | updatedAt | date | Ngày cập nhật |

### Chương III: Cài đặt giải pháp

#### Công nghệ sử dụng

* 1. Ngôn ngữ lập trình Typescript

Trước khi tìm hiểu Typescript, trước tiên giới thiệu về một trong những ngôn ngữ lập trình phổ biến nhất thế giới Javascript.

JavaScript là một ngôn ngữ lập trình bậc cao, chủ yếu được sử dụng để phát triển các tính năng tương tác trên trang web. Được phát triển lần đầu vào năm 1995 bởi Brendan Eich tại Netscape, JavaScript ban đầu chỉ là ngôn ngữ script phía client (chạy trên trình duyệt), nhưng hiện nay nó đã phát triển mạnh mẽ và có thể chạy trên cả phía server nhờ vào môi trường như Node.js.

Một 2 đặc điểm chính của Javascript như sau:

Ngôn ngữ động (Dynamic Language): JavaScript không yêu cầu khai báo kiểu dữ liệu rõ ràng cho biến và có thể thay đổi kiểu dữ liệu trong quá trình chạy.

Không đồng bộ và đơn luồng (Asynchronous and Single-threaded): JavaScript hoạt động theo mô hình đơn luồng nhưng sử dụng cơ chế không đồng bộ (asynchronous) và vòng lặp sự kiện (event loop) để quản lý các tác vụ như xử lý sự kiện, AJAX, và các hoạt động không chặn.

Phát triển web phía client: JavaScript cho phép tạo các tính năng động trên các trang web, như xác thực biểu mẫu, tương tác người dùng, và cập nhật nội dung mà không cần tải lại trang.

Phát triển server với Node.js: Node.js mở rộng JavaScript để chạy trên server, cho phép xây dựng các API, quản lý cơ sở dữ liệu, và tạo ứng dụng real-time như chat và trò chơi trực tuyến.Một số ứng dụng cơ bản của Javascript

Phát triển ứng dụng di động: Các framework như React Native và Ionic cho phép các nhà phát triển sử dụng JavaScript để xây dựng ứng dụng di động trên cả IOS và Android.

Một số hạn chế của Javascript:

Thiếu kiểu dữ liệu tĩnh(static typing): JavaScript là ngôn ngữ không có kiểu dữ liệu tĩnh, nghĩa là bạn không xác định kiểu cho biến khi khai báo. Điều này làm cho mã JavaScript rất linh hoạt, nhưng lại dễ gây lỗi khi không nhận diện được kiểu dữ liệu của biến một cách rõ ràng.

Không phát hiện lỗi khi biên dịch: JavaScript không kiểm tra lỗi ở giai đoạn biên dịch vì đây là ngôn ngữ thông dịch, nên các lỗi chỉ được phát hiện khi chạy mã. Điều này gây khó khăn trong việc phát hiện và sửa lỗi sớm, đặc biệt là trong các dự án phức tạp.

Khó khăn trong bảo trì mã nguồn lớn: Với các dự án lớn, việc bảo trì mã JavaScript có thể khó khăn do thiếu kiểm tra kiểu dữ liệu và kiểm soát chặt chẽ về cấu trúc.

* + 1. Typescript là gì?

TypeScript là một ngôn ngữ lập trình mã nguồn mở là một phần mở rộng của JavaScript với các tính năng bổ sung về kiểu dữ liệu tĩnh (static typing).

TypeScript giúp tăng cường sự an toàn và đáng tin cậy cho mã JavaScript, giúp các nhà phát triển phát hiện lỗi ngay trong quá trình viết mã, thay vì phải chờ đến lúc chạy. Điều này đặc biệt hữu ích khi làm việc với các dự án lớn, nơi mà các vấn đề về lỗi có thể phức tạp và khó tìm.

* + 1. Đặc điểm chính của typescript

Hỗ trợ kiểu dữ liệu tĩnh (Static Typing): TypeScript cho phép định nghĩa kiểu cho biến, hàm, và đối tượng, giúp phát hiện lỗi trong quá trình viết mã, cải thiện khả năng tự hoàn thiện mã và khả năng đọc hiểu.

Tương thích với JavaScript: TypeScript được thiết kế để tương thích ngược hoàn toàn với JavaScript. Điều này có nghĩa là mọi mã JavaScript hợp lệ đều là mã TypeScript hợp lệ.

Hỗ trợ các tính năng hiện đại của JavaScript: TypeScript hỗ trợ các tính năng mới nhất của JavaScript, bao gồm các tiêu chuẩn ES6, ES7 và các phiên bản mới hơn. Trình biên dịch TypeScript chuyển mã TypeScript thành mã JavaScript tương thích với các trình duyệt khác nhau.

Chính vì những lợi thế so với Javascript, trong đề tài này sẽ sử dụng ngôn ngữ lập trình Typescript để giải quyết vấn đề đã đặt ra ở phần đặt tả yêu cầu.

* 1. NextJS
     1. NextJS là gì?

Next.js là một framework mạnh mẽ được xây dựng dựa trên React, giúp phát triển các ứng dụng web với khả năng render phía server (SSR - Server-Side Rendering) và các tính năng tối ưu hóa khác. Được phát triển bởi Vercel, Next.js hướng đến việc đơn giản hóa quá trình phát triển web và cải thiện hiệu năng cũng như trải nghiệm người dùng.

* + 1. Các tính năng nỗi bật:

Hybrid Rendering Modes (SSR, CSR)

Server-Side Rendering (SSR): Next.js cho phép render trang trên server tại thời điểm yêu cầu, sau đó gửi HTML đã render về cho client. Điều này giúp cải thiện SEO và giảm thời gian tải trang ban đầu.

Client-Side Rendering (CSR): Next.js vẫn hỗ trợ render phía client cho các phần của ứng dụng khi cần, kết hợp tốt với các chế độ render khác để tạo trải nghiệm mượt mà.

File-based Routing

Next.js sử dụng cách tổ chức file trong thư mục /pages để tự động tạo các route cho ứng dụng, giúp giảm công sức cấu hình. Mỗi file trong thư mục này sẽ đại diện cho một route.

Dynamic Routing: Next.js hỗ trợ route động, cho phép bạn tạo các route dựa trên các tham số trong URL. Ví dụ, file [id].js trong thư mục /pages sẽ tương ứng với route /posts/123.

API Routes

Next.js cho phép tạo API route ngay trong thư mục /pages/api, giúp xây dựng các API backend mà không cần cài đặt server riêng. Điều này giúp phát triển các ứng dụng full-stack nhanh chóng và dễ dàng hơn.

Typescript Support

Next.js hỗ trợ TypeScript ngay từ khi khởi tạo dự án, giúp phát triển mã nguồn an toàn và dễ bảo trì hơn với kiểu dữ liệu tĩnh.

Tóm lại Next.js là một framework mạnh mẽ cho các ứng dụng React, cung cấp nhiều tính năng hữu ích như SSR, CSR, routing linh hoạt, API routes. Các tính năng này giúp Next.js trở thành một lựa chọn lý tưởng cho các ứng dụng web hiệu năng cao, dễ mở rộng, là một framework tốt để phát triển website giải quyết vấn đề đã đặt ra.

* 1. Postgresql

PostgreSQL là một hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) mã nguồn mở, mạnh mẽ và phổ biến, được sử dụng để quản lý dữ liệu trong các ứng dụng và hệ thống có yêu cầu về độ tin cậy, hiệu suất và tính mở rộng. PostgreSQL được biết đến với tính linh hoạt và khả năng hỗ trợ nhiều tính năng tiên tiến trong việc xử lý và quản lý cơ sở dữ liệu.

Một số đặc điểm của Postgresql:

Hệ quản trị cơ sở dữ liệu quan hệ

PostgreSQL hỗ trợ mô hình cơ sở dữ liệu quan hệ, nơi dữ liệu được lưu trữ trong các bảng có mối quan hệ với nhau thông qua các khóa chính và khóa ngoại, giúp dễ dàng tổ chức và quản lý dữ liệu.

Tính tuân thủ ACID (Atomicity, Consistency, Isolation, Durability)

PostgreSQL hoàn toàn tuân thủ các nguyên lý ACID, đảm bảo tính toàn vẹn dữ liệu và khả năng phục hồi sau sự cố. Điều này giúp bảo vệ dữ liệu khỏi các lỗi phần mềm hoặc phần cứng.

. Hỗ trợ JSON và các kiểu dữ liệu không quan hệ

PostgreSQL là một trong những cơ sở dữ liệu quan hệ đầu tiên cung cấp hỗ trợ mạnh mẽ cho dữ liệu JSON, giúp dễ dàng lưu trữ và truy vấn dữ liệu không cấu trúc mà không cần phải chuyển đổi sang các cơ sở dữ liệu NoSQL.

Hỗ trợ tính năng như full-text search

PostgreSQL hỗ trợ tìm kiếm văn bản toàn bộ (full-text search), giúp tìm kiếm nhanh chóng trong các văn bản lớn mà không cần sử dụng công cụ bên ngoài.

So sánh Mysql và Postgresql

MySQL và PostgreSQL đều là hai hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) phổ biến và mã nguồn mở, nhưng chúng có những điểm mạnh và tính năng khác nhau phù hợp với các trường hợp sử dụng khác nhau.

Hiệu suất

MySQL thường được cho là nhanh hơn trong các truy vấn đọc đơn giản và khi sử dụng trong các ứng dụng có khối lượng truy vấn cao nhưng ít phức tạp. MySQL sử dụng các chỉ mục B-tree và có khả năng tối ưu hóa cho các thao tác đọc dữ liệu với số lượng lớn.

PostgreSQL được biết đến với khả năng xử lý các truy vấn phức tạp, đặc biệt là các truy vấn với nhiều phép toán join, subquery hoặc các hàm phức tạp. Nó có một hệ thống lập kế hoạch truy vấn mạnh mẽ giúp tối ưu hóa hiệu suất cho các truy vấn phức tạp.

Quản lý dữ liệu

MySQL đơn giản và dễ sử dụng, thích hợp với các ứng dụng yêu cầu quản lý dữ liệu cơ bản và có khối lượng truy vấn lớn nhưng không quá phức tạp.

PostgreSQL hỗ trợ nhiều loại dữ liệu nâng cao như hình học (geometric data types), mảng, dữ liệu JSON/JSONB, và XML, giúp xử lý dữ liệu phức tạp và không quan hệ dễ dàng hơn.

Lựa chọn sử dụng

Mysql Thường được lựa chọn cho các ứng dụng web đơn giản, hệ thống yêu cầu khối lượng lớn các truy vấn đọc (ví dụ: blog, website thương mại điện tử, và các ứng dụng có khối lượng dữ liệu vừa phải).

Postgresql Thường được chọn cho các ứng dụng yêu cầu tính toàn vẹn cao, các truy vấn phức tạp, hoặc các hệ thống có yêu cầu về dữ liệu lớn, phân tán hoặc dữ liệu không quan hệ (ví dụ: phân tích dữ liệu, hệ thống GIS, ứng dụng tài chính, v.v.).

Trong đề tài này sẽ sử dụng postgresql làm cơ sở dữ liệu cho website

* 1. Tailwind

Tailwind CSS là một framework CSS tiện ích (utility-first CSS framework) mã nguồn mở được sử dụng để xây dựng giao diện người dùng nhanh chóng và dễ dàng. Khác với các framework CSS truyền thống như Bootstrap hoặc Foundation, Tailwind cung cấp các lớp tiện ích có thể tái sử dụng để tạo kiểu cho các phần tử HTML mà không cần phải viết CSS tùy chỉnh. Điều này giúp giảm bớt việc viết mã CSS riêng biệt và tạo ra các giao diện linh hoạt, có thể thay đổi nhanh chóng.

Môt số đặc điểm

Utility-First Design

Tailwind sử dụng phong cách utility-first (tiện ích trước), nghĩa là bạn sử dụng các lớp CSS tiện ích đơn giản để áp dụng kiểu dáng trực tiếp vào các phần tử HTML thay vì viết các lớp tùy chỉnh.

Responsive Design

Tailwind cung cấp các lớp tiện ích cho responsive design (thiết kế đáp ứng), cho phép bạn dễ dàng tạo giao diện web thân thiện với mọi thiết bị. Các lớp tiện ích như sm:, md:, lg:, xl: giúp áp dụng các kiểu dáng khác nhau cho các kích thước màn hình khác nhau.

Kết luận

Tailwind CSS là một công cụ mạnh mẽ, giúp đơn giản hóa việc xây dựng giao diện người dùng với các lớp tiện ích dễ sử dụng và rất dễ tùy chỉnh. Đây là một lựa chọn lý tưởng cho các nhà phát triển web muốn tạo giao diện nhanh chóng, dễ bảo trì và có thể mở rộng cho các dự án lớn.

* 1. ShadcnUI

Shadcn UI là một thư viện giao diện người dùng (UI library) hiện đại được xây dựng cho các ứng dụng React. Shadcn UI giúp bạn xây dựng các giao diện người dùng đẹp mắt, dễ sử dụng và tối ưu hiệu suất bằng cách sử dụng các thành phần giao diện người dùng tùy chỉnh (customizable components). Thư viện này được phát triển với mục tiêu tạo ra những giao diện thân thiện với người dùng và dễ dàng tích hợp vào các dự án web hiện đại.

. Component-Based Design

Shadcn UI cung cấp một bộ các component UI sẵn có, giúp bạn dễ dàng tạo ra các giao diện người dùng như nút bấm (buttons), hộp thoại (modals), biểu mẫu (forms), bảng (tables), danh sách (lists), v.v.

Các thành phần này có thể dễ dàng tùy chỉnh để phù hợp với nhu cầu của bạn thông qua việc sử dụng props hoặc các tính năng mở rộng (extensibility).

2. Được Thiết Kế Theo Chế Độ Dark Mode và Light Mode

Shadcn UI hỗ trợ chế độ dark mode và light mode một cách mượt mà, giúp ứng dụng của bạn có thể dễ dàng chuyển đổi giữa hai chế độ giao diện này dựa trên sở thích của người dùng hoặc cài đặt hệ thống.

3. Tích Hợp Với Tailwind CSS

Một trong những điểm đặc biệt của Shadcn UI là nó được xây dựng để hoạt động mượt mà với Tailwind CSS, giúp bạn sử dụng các lớp tiện ích của Tailwind để tùy chỉnh giao diện của các thành phần một cách dễ dàng mà không cần phải viết CSS riêng biệt.

Việc tích hợp này giúp việc phát triển giao diện nhanh chóng và linh hoạt hơn.

Shadcn UI là một thư viện giao diện người dùng mạnh mẽ, dễ sử dụng và dễ tùy chỉnh, giúp bạn xây dựng giao diện web nhanh chóng và hiệu quả. Với khả năng tích hợp mượt mà với Tailwind CSS và tập trung vào trải nghiệm người dùng, Shadcn UI là một lựa chọn tuyệt vời cho các dự án React hiện đại.

* 1. Clerk

Clerk là một dịch vụ cung cấp các giải pháp xác thực và quản lý người dùng cho các ứng dụng web và di động. Clerk giúp bạn dễ dàng triển khai hệ thống đăng nhập, đăng ký người dùng, và quản lý phiên làm việc (session) mà không cần phải tự xây dựng các giải pháp phức tạp. Clerk tích hợp các tính năng bảo mật mạnh mẽ và giao diện người dùng sẵn có để giúp bạn nhanh chóng triển khai và quản lý người dùng trong ứng dụng của mình.

Xác Thực Người Dùng (Authentication)

Clerk hỗ trợ nhiều phương thức xác thực, bao gồm email/password, social login (Google, GitHub, Twitter, v.v.), và passwordless authentication (xác thực không mật khẩu qua liên kết gửi email hoặc SMS).

Quản Lý Người Dùng (User Management)

Clerk cung cấp một giao diện quản trị mạnh mẽ cho phép bạn dễ dàng quản lý người dùng trong ứng dụng của mình, bao gồm thông tin cá nhân, trạng thái tài khoản và các cài đặt bảo mật.

Các API của Clerk cũng cho phép bạn tạo, cập nhật và xóa người dùng từ phía server của mình.

Đăng Ký và Đăng Nhập Xã Hội (Social Login)

Clerk hỗ trợ đăng nhập qua các nền tảng xã hội phổ biến như Google, GitHub, Twitter, Apple và nhiều nền tảng khác. Điều này giúp người dùng có thể dễ dàng đăng nhập vào ứng dụng mà không cần phải tạo tài khoản mới.

Quá trình đăng nhập xã hội được quản lý tự động bởi Clerk, giúp bạn tiết kiệm thời gian và công sức khi triển khai tính năng này.

Tổ chức(Organization):

Clerk hỗ trợ tổ chức(nhóm các người dùng) để quản lý người dùng và quyền truy cập của họ vào tài nguyên của ứng dụng. Với tổ chức, có thể chỉ định vai trò và quyền cụ thể của từng thành viên trong tổ chức giúp thuận lợi cho việc quản lý, điều phối nhóm và tạo điều kiện cho quan hệ đối tác.

Clerk là một giải pháp xác thực người dùng và quản lý phiên làm việc mạnh mẽ, linh hoạt và dễ sử dụng, giúp tiết kiệm thời gian và công sức cho các nhà phát triển khi triển khai tính năng xác thực và bảo mật trong ứng dụng

* 1. Uploadthing

UploadThing là một dịch vụ chuyên cung cấp giải pháp tải lên và lưu trữ tệp cho các ứng dụng web và di động. Dịch vụ này giúp các nhà phát triển dễ dàng tích hợp tính năng tải tệp vào ứng dụng của họ mà không cần phải xây dựng hệ thống tải tệp từ đầu. UploadThing cung cấp một API đơn giản và dễ sử dụng để tải lên và quản lý các tệp, đồng thời đảm bảo tính bảo mật, khả năng mở rộng và hiệu suất cao.

* 1. SocketIO

Socket.IO là một thư viện JavaScript giúp bạn xây dựng các ứng dụng thời gian thực (real-time) trên nền tảng web. Nó cung cấp một giao diện đơn giản để giao tiếp giữa client (trình duyệt) và server thông qua WebSocket và các phương thức truyền thông khác, hỗ trợ kết nối hai chiều (bi-directional communication) giữa client và server.

Socket.IO giúp xử lý các tác vụ như gửi và nhận dữ liệu theo thời gian thực, tương tác ngay lập tức giữa người dùng (chẳng hạn như trong các ứng dụng chat, game trực tuyến, thông báo thời gian thực, v.v.), mà không cần phải tải lại trang hoặc thực hiện các yêu cầu HTTP truyền thống.

* 1. Prisma

Prisma là một thư viện ORM (Object Relational Mapping) mã nguồn mở cho Node.js và TypeScript, giúp quản lý cơ sở dữ liệu trong các ứng dụng web và backend. Prisma cung cấp một công cụ mạnh mẽ để làm việc với cơ sở dữ liệu, giúp các nhà phát triển dễ dàng tương tác với các cơ sở dữ liệu quan hệ như PostgreSQL, MySQL, SQLite, SQL Server, và MongoDB (dự kiến trong tương lai).

Prisma giúp bạn xử lý các thao tác cơ bản với cơ sở dữ liệu như tạo, đọc, cập nhật, và xóa (CRUD) dữ liệu một cách dễ dàng, bảo mật, và hiệu quả, đồng thời cung cấp các tính năng mạnh mẽ để tối ưu hóa các truy vấn và cải thiện hiệu suất.

* 1. Nodemailer

Nodemailer là một thư viện mã nguồn mở dành cho Node.js, giúp gửi email từ các ứng dụng web hoặc server. Nó hỗ trợ gửi email thông qua các dịch vụ SMTP (Simple Mail Transfer Protocol) như Gmail, SendGrid, Amazon SES, hoặc bất kỳ dịch vụ SMTP nào khác. Nodemailer là một trong những thư viện phổ biến nhất để gửi email trong các ứng dụng Node.js nhờ vào tính đơn giản, dễ sử dụng và khả năng tùy chỉnh cao.

* 1. NextInlt

next-intl là một thư viện dành cho Next.js giúp hỗ trợ quản lý và xử lý đa ngôn ngữ (internationalization, i18n) trong các ứng dụng web. Nó giúp bạn dễ dàng thêm tính năng đa ngôn ngữ vào các ứng dụng Next.js mà không cần phải viết mã phức tạp hoặc sử dụng các công cụ bên ngoài.

* 1. Mergent

Mergent một dịch vụ được thiết kế để giúp bạn thực thi các tác vụ định kỳ và tác vụ nền (background jobs) một cách dễ dàng và hiệu quả mà không cần phải quản lý hạ tầng máy chủ. Dịch vụ này phù hợp cho các ứng dụng serverless hoặc những môi trường mà bạn không muốn tự quản lý các tác vụ định kỳ, như gửi email hàng loạt, cập nhật dữ liệu tự động, hay các công việc xử lý dài hạn khác.

### Chương IV: Đánh giá kiểm thử

# Phần kết luận

#### Kết quả đạt được

#### Hướng phát triển

# Tài liệu tham khảo

# Phụ lục