**BỘ LAO ĐỘNG THƯƠNG BINH & XÃ HỘI**

**TRƯỜNG CAO ĐĂNG CÔNG NGHỆ THÔNG TIN TP.HCM**

**KHOA CÔNG NGHỆ THÔNG TIN – ĐIỆN TỬ**

****

**BÁO CÁO ĐỀ TÀI**

**MÔN HỌC: ỨNG DỤNG WEB**

**ĐỀ TÀI: XÂY DỰNG HỆ THỐNG WEBSITE THƯƠNG MẠI ĐIỆN TỬ BABY CHILL**

**Giảng viên hướng dẫn: Trương Bá Vĩnh**

**Sinh viên thực hiện: Huỳnh Văn Tuấn MSSV: 501230558**

**Nguyễn Thanh Tùng MSSV: 514230566**

**Trương Minh Hiếu MSSV: 501230575**

**Tên nhóm học phần: 501\_UDW\_HK2.2425\_CD23CT.N5**

**Học kỳ: 2 Năm học: 2024 – 2025**

**TP. Hồ Chí Minh, tháng 02 năm 2025**

**BỘ LAO ĐỘNG THƯƠNG BINH & XÃ HỘI**

**TRƯỜNG CAO ĐĂNG CÔNG NGHỆ THÔNG TIN TP.HCM**

**KHOA CÔNG NGHỆ THÔNG TIN – ĐIỆN TỬ**

****

**BÁO CÁO ĐỀ TÀI**

**MÔN HỌC: ỨNG DỤNG WEB**

**ĐỀ TÀI: XÂY DỰNG HỆ THỐNG WEBSITE THƯƠNG MẠI ĐIỆN TỬ BABY CHILL**

**Giảng viên hướng dẫn: Trương Bá Vĩnh**

**Sinh viên thực hiện: Huỳnh Văn Tuấn MSSV: 501230558**

**Nguyễn Thanh Tùng MSSV: 514230566**

**Trương Minh Hiếu MSSV: 501230575**

**Tên nhóm học phần: 501\_UDW\_HK2.2425\_CD23CT.N5**

**Học kỳ: 2 Năm học: 2024 – 2025**

**TP. Hồ Chí Minh, tháng 02 năm 2025**

LỜI MỞ ĐẦU

Trong thời đại công nghệ số phát triển mạnh mẽ, thương mại điện tử ngày càng trở thành xu hướng phổ biến, mang lại nhiều tiện ích cho cả doanh nghiệp và người tiêu dùng. Đặc biệt, trong lĩnh vực kinh doanh các sản phẩm dành cho trẻ sơ sinh và trẻ nhỏ, nhu cầu mua sắm trực tuyến ngày càng tăng cao do tính tiện lợi, đa dạng sản phẩm và khả năng tiếp cận dễ dàng.

Với mong muốn mang đến một giải pháp hiện đại, hỗ trợ các cửa hàng kinh doanh đồ dùng cho trẻ sơ sinh tiếp cận khách hàng tốt hơn, chúng tôi thực hiện đồ án **"Xây dựng hệ thống website thương mại điện tử Baby Chill"**. Hệ thống này không chỉ giúp người tiêu dùng dễ dàng tìm kiếm và mua sắm các sản phẩm chất lượng mà còn hỗ trợ cửa hàng quản lý bán hàng, đơn hàng, khách hàng một cách hiệu quả.

Website được thiết kế với giao diện thân thiện, dễ sử dụng, tích hợp các tính năng quan trọng như: danh mục sản phẩm phong phú, giỏ hàng, thanh toán trực tuyến, theo dõi đơn hàng và hệ thống quản lý dành cho người bán. Ngoài ra, hệ thống còn hướng đến việc tối ưu trải nghiệm người dùng và bảo mật thông tin giao dịch, đảm bảo sự thuận tiện và an toàn khi mua sắm trực tuyến.

**MỤC LỤC**

[CHƯƠNG 1. TỔNG QUAN ĐỀ TÀI 1](#_Toc190819167)

[1.1. Giới thiệu – Lý do chọn đề tài 1](#_Toc190819168)

[1.2. Mục tiêu của đề tài 1](#_Toc190819169)

[1.2.1. Mục tiêu tổng quát 1](#_Toc190819170)

[1.2.2. Mục tiêu cụ thể 1](#_Toc190819171)

[1.3. Phạm vi nghiên cứu 2](#_Toc190819172)

[1.4. Các công nghệ sử dụng 2](#_Toc190819173)

[1.4.1. Tổng quan về Express.js 2](#_Toc190819174)

[1.4.2. Tổng quan về React.js 2](#_Toc190819175)

[1.4.3. Tổng quan về MongoDB 2](#_Toc190819176)

[1.4.4. Tổng quan thư viện Recharts 3](#_Toc190819177)

[1.4.5. Tổng quan thư viện axios 3](#_Toc190819178)

[1.4.6. Tổng quan thư viện jwt-decode 3](#_Toc190819179)

[CHƯƠNG 2. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG 4](#_Toc190819180)

[2.1. Các tác nhân hệ thống 4](#_Toc190819181)

[2.2. Sơ đồ usecase 4](#_Toc190819182)

[CHƯƠNG 3. THIẾT KẾ CƠ SỞ DỮ LIỆU 5](#_Toc190819184)

[3.1. Tổng quan cơ sở dữ liệu 5](#_Toc190819185)

[CHƯƠNG 4. GIAO DIỆN NGƯỜI DUNG VÀ CHỨC NĂNG HỆ THỐNG 6](#_Toc190819187)

[4.1. Giao diện người dùng phía khách hàng và các chức năng 6](#_Toc190819188)

[4.2. Giao diện người dùng phía quản trị viên và các chức năng 12](#_Toc190819199)

[CHƯƠNG 5. CẤU TRÚC THƯ MỤC VÀ CODE 15](#_Toc190819206)

[5.1. Cấu trúc thư mục backend 15](#_Toc190819207)

[5.2. Cấu trúc thư mục frontend 18](#_Toc190819209)

[HƯỚNG DẪN XEM ĐỒ ÁN 21](#_Toc190819211)

[KẾT LUẬN 22](#_Toc190819212)

[DANH MỤC THAM KHẢO 23](#_Toc190819213)

**DANH MỤC HÌNH**

[Hình 2.1. Sơ đồ usecase tổng quan 4](#_Toc190819183)

[Hình 3.1. Sơ đồ ERD tổng quan cơ sở dữ liệu 5](#_Toc190819186)

[Hình 4.1. Giao diện người dùng phía khách hàng 6](#_Toc190819189)

[Hình 4.2. Giao diên người dùng chức năng đăng nhập tài khoản 6](#_Toc190819190)

[Hình 4.3. Giao diện người dùng chức năng đăng ký tài khoản 7 Hình 4.4. Giao diện người dùng chức năng tìm kiếm sản phẩm 7](#_Toc190819191)

[Hình 4.5. Giao diện người dùng chức năng xem chi tiết sản phẩm 8](#_Toc190819192)

[Hình 4.6. Giao diện người dùng chức năng thanh toán 8](#_Toc190819193)

[Hình 4.7. Giao diện người dùng chức năng thanh toán 9](#_Toc190819194)

[Hình 4.8. Chức năng thông báo đơn hàng qua mail 9](#_Toc190819195)

[Hình 4.9. Giao diện người dùng chức năng chỉnh sửa thông tin cá nhân 10](#_Toc190819196)

[Hình 4.10. Giao diện người dùng chức năng xem lịch sử đơn hàng 10](#_Toc190819197)

[Hình 4.12. Giao diện người dùng trang quản lý và chức năng xuất báo cáo sang file excel 12](#_Toc190819200)

[Hình 4.13. Giao diện người dùng chức năng quản lý đơn hàng 12](#_Toc190819201)

[Hình 4.14. Giao diện người dùng chức năng quản lý sản phẩm 13](#_Toc190819202)

[Hình 4.15. Giao diện người dùng chức năng quản lý tài khoản 13](#_Toc190819203)

[Hình 4.16. Giao diện người dùng chức năng quản lý bài viết 14](#_Toc190819204)

[Hình 4.17. Giao diện người dùng chức năng tạo bài viết 14](#_Toc190819205)

[Hình 5.1. Cấu trúc thư mục backend 15](#_Toc190819208)

[Hình 5.2. Cấu trúc thư mục frontend 18](#_Toc190819210)

**DANH SÁCH TỪ VIẾT TẮT**

|  |  |
| --- | --- |
| F&B | Food and Beverage |
| SaaS | Software as a Service |
| DI | Dependency Injection |
| CLI | Command Line Interface |
| API | Application Programming Interface |
| SSR | Server-Side Rendering |
| SSG | Static Site Generation |
| SEO | Search Engine Optimization |
| ACID | Atomicity, Consistency, Isolation, Durability |
| JSON | JavaScript Object Notation |
|  |  |

CHƯƠNG 1. TỔNG QUAN ĐỀ TÀI

* 1. Giới thiệu – Lý do chọn đề tài

Hiện nay, nhu cầu mua sắm trực tuyến đang ngày càng gia tăng, vấn đề hàng giả hàng, hàng kém chất lượng tràn lan trên thị trường gây ra nhiều mối lo ngại cho sức khỏe người tiêu dùng, đặc biệt trong lĩnh vực tiêu dùng cho mẹ và bé. Các bậc phụ huynh luôn tìm kiếm những sản phẩm chất lượng, an toàn và thuận tiện trong việc mua sắm để chăm sóc con cái tốt hơn. Tuy nhiên, không phải cửa hàng nào cũng có hệ thống bán hàng trực tuyến chuyên nghiệp, dễ sử dụng và đáp ứng đầy đủ nhu cầu khách hàng.

Thực tế cho thấy, các doanh nghiệp lớn như **Con Cưng, Bibomart, KidsPlaza** đã thành công trong việc áp dụng thương mại điện tử để mở rộng thị trường, nâng cao trải nghiệm khách hàng. Điều này chứng tỏ sự cần thiết của một hệ thống website thương mại điện tử hiện đại, giúp các cửa hàng kinh doanh đồ dùng trẻ sơ sinh tiếp cận khách hàng nhanh chóng hơn, tối ưu quá trình quản lý và nâng cao doanh thu.

Với mong muốn ứng dụng công nghệ vào thực tiễn và góp phần giải quyết bài toán trên, chúng tôi quyết định thực hiện đề tài **“Xây dựng hệ thống website thương mại điện tử Baby Chill”**.

* 1. Mục tiêu của đề tài
     1. Mục tiêu tổng quát

Mục tiêu chung của đề tài là xây dựng và phát triển hệ thống website thương mại điện tử Baby Chill dành cho các cửa hàng bán đồ cho trẻ sơ sinh. Website này sẽ giúp nâng cao hiệu quả kinh doanh, tối ưu quy trình bán hàng và cung cấp cho người tiêu dùng một nền tảng mua sắm trực tuyến an toàn, tiện lợi và dễ sử dụng.

* + 1. Mục tiêu cụ thể

Phát triển website thương mại điện tử hoàn chỉnh**:** Xây dựng giao diện người dùng thân thiện, dễ sử dụng và tương thích với các thiết bị di động. Tích hợp các tính năng cơ bản như danh mục sản phẩm, giỏ hàng, thanh toán trực tuyến, và theo dõi đơn hàng.

Tạo hệ thống quản lý dành cho cửa hàng**:** Phát triển hệ thống quản lý sản phẩm, đơn hàng, khách hàng và các báo cáo kinh doanh. Tạo các công cụ để cửa hàng dễ dàng cập nhật, chỉnh sửa thông tin sản phẩm và quản lý kho hàng.

* 1. Phạm vi nghiên cứu

Phạm vi nghiên cứu của đồ án tập trung vào phát triển nền tảng thương mại điện tử cho cửa hàng bán đồ dùng cho em bé, bao gồm các tính năng như danh mục sản phẩm, giỏ hàng, thanh toán trực tuyến và các tính năng quản lý cho cửa hàng như: quản lý đơn hàng, quản lý người dùng, theo dõi và lập báo cáo. Công nghệ sử dụng gồm **Node.js** (Express.js), **React.js** và **MongoDB**. Phạm vi không bao gồm marketing hay vận hành website sau triển khai.

* 1. Các công nghệ sử dụng
     1. Tổng quan về Express.js

Express.js là một framework cho Node.js mã nguồn mở, đơn giản nhưng mạnh mẽ, giúp phát triển các ứng dụng web và API nhanh chóng và dễ dàng. Với kiến trúc tối giản và khả năng mở rộng cao, Express.js hỗ trợ các tính năng như routing, middleware và các cơ chế xử lý yêu cầu HTTP linh hoạt. Nhờ vào việc không có cấu trúc bắt buộc, Express.js cho phép các nhà phát triển tự do tùy chỉnh và tối ưu hóa ứng dụng. Đây là lựa chọn phổ biến để xây dựng các ứng dụng web, đặc biệt trong các dự án yêu cầu hiệu suất cao, nhanh chóng và dễ bảo trì.

* + 1. Tổng quan về React.js

React.js là một thư viện JavaScript mã nguồn mở được phát triển bởi Facebook, được sử dụng chủ yếu để xây dựng giao diện người dùng (UI) cho các ứng dụng web đơn trang (SPA). React.js nổi bật với khả năng quản lý DOM ảo (Virtual DOM), giúp tăng hiệu suất và giảm thời gian tải trang. Thư viện này hỗ trợ việc tạo ra các component tái sử dụng, giúp mã nguồn dễ dàng bảo trì và mở rộng. React.js được sử dụng rộng rãi trong các dự án web hiện đại, đặc biệt là các ứng dụng thương mại điện tử, mạng xã hội, và các hệ thống có yêu cầu cao về trải nghiệm người dùng.

* + 1. Tổng quan về MongoDB

MongoDB là một cơ sở dữ liệu NoSQL mã nguồn mở, lưu trữ dữ liệu dưới dạng tài liệu JSON thay vì bảng như trong các cơ sở dữ liệu quan hệ SQL. MongoDB hỗ trợ khả năng mở rộng linh hoạt và hiệu suất cao, đặc biệt trong các ứng dụng yêu cầu xử lý khối lượng lớn dữ liệu phi cấu trúc hoặc bán cấu trúc. Với tính năng tự động phân chia dữ liệu (sharding), MongoDB cho phép ứng dụng mở rộng dễ dàng khi cần xử lý lượng truy vấn lớn hoặc lưu trữ dữ liệu trên nhiều máy chủ. MongoDB thường được sử dụng trong các ứng dụng web, dịch vụ đám mây và các hệ thống có yêu cầu về dữ liệu không gian và thời gian.

* + 1. Tổng quan thư viện Recharts

Recharts là một thư viện biểu đồ React mạnh mẽ, giúp bạn tạo ra các biểu đồ tương tác và tùy chỉnh ngay trong ứng dụng React của mình. Với cách tiếp cận khai báo, Recharts cho phép bạn dễ dàng kết hợp các thành phần biểu đồ để tạo ra nhiều loại biểu đồ khác nhau, từ biểu đồ đường, cột đến biểu đồ tròn và nhiều hơn nữa. Nhờ được xây dựng trên React, Recharts mang lại hiệu suất tốt và khả năng tùy biến cao, lý tưởng để trực quan hóa dữ liệu trong các ứng dụng web hiện đại.

* + 1. Tổng quan thư viện axios

Axios là một thư viện HTTP client mạnh mẽ cho JavaScript, hoạt động trên cả trình duyệt và Node.js, nổi bật với việc sử dụng Promises giúp code bất đồng bộ dễ quản lý hơn. Thư viện này cung cấp nhiều tính năng như interceptors để xử lý request và response, tự động chuyển đổi JSON, bảo vệ chống XSRF, và khả năng hủy request. Với sự linh hoạt và dễ sử dụng, Axios là lựa chọn phổ biến để thực hiện các yêu cầu HTTP và tương tác với API trong phát triển web.

* + 1. Tổng quan thư viện jwt-decode

Thư viện jwt-decode là một thư viện JavaScript nhỏ gọn và hữu ích, được thiết kế để giải mã JSON Web Tokens (JWTs) một cách nhanh chóng và dễ dàng. Điểm quan trọng cần nhớ là nó chỉ thực hiện việc giải mã token, chứ không xác minh chữ ký của token. Thường được sử dụng ở phía client (trình duyệt), jwt-decode cho phép bạn truy cập và sử dụng thông tin được mã hóa bên trong JWT, chẳng hạn như thông tin người dùng hoặc vai trò, sau khi token đã được xác thực và trả về từ server. Điểm mạnh của thư viện nằm ở sự nhỏ gọn, tốc độ, và tính dễ sử dụng, khiến nó trở thành lựa chọn lý tưởng cho các ứng dụng client-side cần đọc thông tin từ JWT mà không cần thực hiện xác thực. Hãy nhớ rằng, jwt-decode không nên được sử dụng để xác thực JWT, việc này luôn phải được thực hiện ở phía server để đảm bảo an toàn.

CHƯƠNG 2. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

2.1. Các tác nhân hệ thống

* Khách hàng:
  + Tạo tài khoản và đăng nhập: Đăng ký, đăng nhập vào hệ thống.
  + Duyệt sản phẩm và mua sắm: Tìm kiếm, xem sản phẩm, thêm vào giỏ hàng và thanh toán.
  + Theo dõi đơn hàng: Kiểm tra tình trạng đơn hàng và lịch sử giao dịch.
  + Đánh giá sản phẩm: Cung cấp phản hồi về sản phẩm đã mua.
* Quản trị viên
  + Quản lý người dùng: Tạo, chỉnh sửa, xóa tài khoản khách hàng.
  + Quản lý sản phẩm: Thêm, sửa, xóa sản phẩm trong hệ thống.
  + Quản lý đơn hàng: Theo dõi và cập nhật trạng thái đơn hàng.
  + Xem báo cáo: Truy cập báo cáo doanh thu, đơn hàng và hoạt động hệ thống.

2.2. Sơ đồ usecase

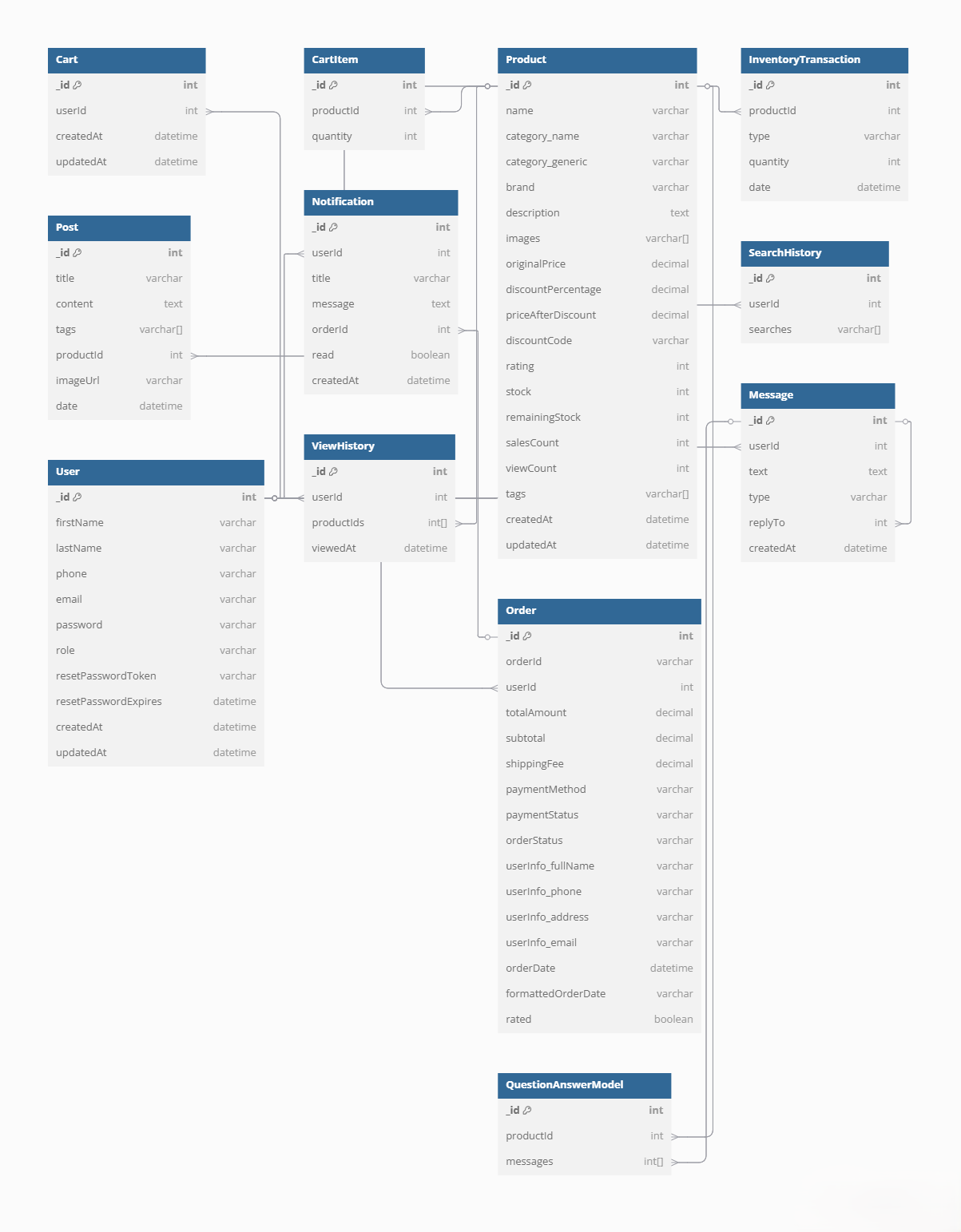
Ảnh có chứa văn bản, biểu đồ, hàng, Phông chữ

Nội dung do AI tạo ra có thể không chính xác.

*Hình 2.1. Sơ đồ usecase tổng quan*

CHƯƠNG 3. THIẾT KẾ CƠ SỞ DỮ LIỆU

3.1. Tổng quan cơ sở dữ liệu



*Hình 3.1. Sơ đồ ERD tổng quan cơ sở dữ liệu*

CHƯƠNG 4. GIAO DIỆN NGƯỜI DUNG VÀ CHỨC NĂNG HỆ THỐNG

4.1. Giao diện người dùng phía khách hàng và các chức năng

Ảnh có chứa văn bản, trang phục, ảnh chụp màn hình, người

Nội dung do AI tạo ra có thể không chính xác.

#### Hình 4.1. Giao diện người dùng phía khách hàng

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Trang web

Nội dung do AI tạo ra có thể không chính xác.*Hình 4.2. Giao diên người dùng chức năng đăng nhập tài khoản*

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Biểu tượng máy tính

Nội dung do AI tạo ra có thể không chính xác.*Hình 4.3. Giao diện người dùng chức năng đăng ký tài khoản*Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Trang web

Nội dung do AI tạo ra có thể không chính xác.*Hình 4.4. Giao diện người dùng chức năng tìm kiếm sản phẩm*

Ảnh có chứa văn bản, ảnh chụp màn hình, Trang web, phần mềm

Nội dung do AI tạo ra có thể không chính xác.*Hình 4.5. Giao diện người dùng chức năng xem chi tiết sản phẩm*

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Trang web

Nội dung do AI tạo ra có thể không chính xác.*Hình 4.6. Giao diện người dùng chức năng thanh toán*

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Biểu tượng máy tính

Nội dung do AI tạo ra có thể không chính xác.*Hình 4.7. Giao diện người dùng chức năng thanh toán*

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Biểu tượng máy tính

Nội dung do AI tạo ra có thể không chính xác.*Hình 4.8. Chức năng thông báo đơn hàng qua mail*

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Trang web

Nội dung do AI tạo ra có thể không chính xác.*Hình 4.9. Giao diện người dùng chức năng chỉnh sửa thông tin cá nhân*

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Biểu tượng máy tính

Nội dung do AI tạo ra có thể không chính xác.*Hình 4.10. Giao diện người dùng chức năng xem lịch sử đơn hàng*

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Trang web

Nội dung do AI tạo ra có thể không chính xác.*Hình 4.11. Giao diện người dùng chức năng giỏ hàng*

4.2. Giao diện người dùng phía quản trị viên và các chức năng

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Biểu tượng máy tính

Nội dung do AI tạo ra có thể không chính xác.

#### Hình 4.12. Giao diện người dùng trang quản lý và chức năng xuất báo cáo sang file excel

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Trang web

Nội dung do AI tạo ra có thể không chính xác.*Hình 4.13. Giao diện người dùng chức năng quản lý đơn hàng*

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Biểu tượng máy tính

Nội dung do AI tạo ra có thể không chính xác.

#### Hình 4.14. Giao diện người dùng chức năng quản lý sản phẩm

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Biểu tượng máy tính

Nội dung do AI tạo ra có thể không chính xác.*Hình 4.15. Giao diện người dùng chức năng quản lý tài khoản*

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Trang web

Nội dung do AI tạo ra có thể không chính xác.

#### Hình 4.16. Giao diện người dùng chức năng quản lý bài viết

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Biểu tượng máy tính

Nội dung do AI tạo ra có thể không chính xác.

#### Hình 4.17. Giao diện người dùng chức năng tạo bài viết

CHƯƠNG 5. CẤU TRÚC THƯ MỤC VÀ CODE

5.1. Cấu trúc thư mục backend

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Phần mềm đa phương tiện

Nội dung do AI tạo ra có thể không chính xác.

#### Hình 5.1. Cấu trúc thư mục backend

Backend sử dụng Node.js và Express.js được tổ chức theo một cấu rõ ràng, hỗ trợ việc phát triển và bảo trì ứng dụng hiệu quả. Các thành phần chính trong cấu trúc thư mục bao gồm:

1. **config**:

Thư mục này chứa các tệp cấu hình ứng dụng, bao gồm các thông tin cần thiết để kết nối với cơ sở dữ liệu, cấu hình các môi trường khác nhau (ví dụ: môi trường phát triển và môi trường sản xuất), và các tham số bảo mật như API keys và thông tin xác thực.

1. **constants**:

Thư mục này chứa các hằng số được sử dụng trong toàn bộ dự án. Các giá trị này có thể là các mã lỗi HTTP, thông báo lỗi mặc định, hoặc các cấu hình và thiết lập tĩnh giúp đảm bảo tính nhất quán trong toàn bộ ứng dụng.

1. **controllers**:

Các bộ điều khiển (controllers) trong thư mục này chịu trách nhiệm xử lý logic nghiệp vụ của ứng dụng. Mỗi controller xử lý các yêu cầu HTTP (GET, POST, PUT, DELETE) từ phía client, tương tác với các mô hình dữ liệu (models) và trả về kết quả phù hợp.

1. **data**:

Thư mục này lưu trữ các tệp dữ liệu mẫu hoặc các tệp hỗ trợ cho ứng dụng, chẳng hạn như tệp JSON, CSV hoặc các dữ liệu khác cần thiết trong quá trình phát triển hoặc kiểm thử.

1. **middleware**:

Middleware là các hàm được thực thi trong quá trình xử lý yêu cầu HTTP trước khi đến các bộ điều khiển. Thư mục này chứa các đoạn mã trung gian như xác thực người dùng, ghi log, xử lý lỗi, và các tác vụ khác nhằm cải thiện hiệu quả và bảo mật của ứng dụng.

1. **models**:

Các mô hình (models) trong thư mục này định nghĩa cấu trúc dữ liệu và các thao tác tương tác với cơ sở dữ liệu. Đối với các ứng dụng sử dụng cơ sở dữ liệu NoSQL như MongoDB, các mô hình có thể được định nghĩa bằng cách sử dụng Mongoose để xác định schema và các phương thức truy vấn.

1. **node\_modules**:

Đây là thư mục chứa tất cả các thư viện và gói phụ thuộc của dự án, được cài đặt thông qua Node Package Manager (NPM). Các gói này cung cấp các chức năng và công cụ cần thiết để phát triển ứng dụng, bao gồm Express.js và các thư viện hỗ trợ khác.

1. **routes**:

Thư mục này chứa các định nghĩa tuyến đường (routes) của Express.js, nơi các URL và phương thức HTTP được ánh xạ đến các controller tương ứng. Các route này giúp điều hướng các yêu cầu từ client đến các bộ điều khiển xử lý logic cụ thể.

1. **utils**:

Thư mục này bao gồm các hàm tiện ích và công cụ hỗ trợ, có thể là các chức năng xử lý chuỗi, thời gian, hoặc các tác vụ khác giúp tăng cường hiệu quả của ứng dụng.

1. **.env**:

Tệp này chứa các biến môi trường, giúp cấu hình các tham số nhạy cảm như thông tin kết nối cơ sở dữ liệu hoặc khóa API mà không cần phải mã hóa trực tiếp trong mã nguồn.

1. **exportData.js**:

Đây là tệp JavaScript dùng để xuất dữ liệu từ cơ sở dữ liệu hoặc các nguồn khác. Nó có thể thực hiện các tác vụ như chuyển dữ liệu ra tệp JSON hoặc CSV để sử dụng trong các ứng dụng hoặc báo cáo khác.

1. **package-lock.json**:

Tệp này được tạo tự động khi cài đặt các gói thông qua NPM, giúp bảo đảm các phiên bản của các gói phụ thuộc được sử dụng trong suốt quá trình phát triển và triển khai, từ đó đảm bảo tính ổn định của ứng dụng.

1. **package.json**:

Tệp này là tệp cấu hình chính của dự án, chứa thông tin về các gói phụ thuộc và các script cần thiết để xây dựng và triển khai ứng dụng. Nó cũng lưu trữ các metadata của dự án, như tên, phiên bản và các thông tin liên quan khác.

1. **server.js**:

Đây là tệp khởi động chính của ứng dụng backend, nơi cấu hình Express server và các middleware được thiết lập. Tệp này cũng quản lý quá trình khởi tạo và khởi động server, giúp ứng dụng hoạt động trên các cổng và môi trường xác định.

5.2. Cấu trúc thư mục frontend

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Phần mềm đa phương tiện

Nội dung do AI tạo ra có thể không chính xác.

#### Hình 5.2. Cấu trúc thư mục frontend

Frontend sử dụng React với Create React App (CRA) được tổ chức theo một cấu trúc chuẩn, giúp dễ dàng quản lý và phát triển ứng dụng. Cấu trúc thư mục này bao gồm các thành phần chính sau:

1. **node\_modules**:

Thư mục này chứa tất cả các gói phụ thuộc được cài đặt thông qua Node Package Manager (NPM). Đây là nơi chứa các thư viện cần thiết cho dự án, bao gồm React và các thư viện hỗ trợ khác.

1. **public**:

Thư mục này chứa các tệp tĩnh phục vụ cho ứng dụng, chẳng hạn như các tệp HTML, favicon, và các tệp tài nguyên khác như hình ảnh hoặc font chữ. Tệp index.html trong thư mục này là điểm khởi đầu cho ứng dụng React.

1. **src**:

Đây là thư mục chính chứa mã nguồn của ứng dụng React. Toàn bộ các thành phần, logic và giao diện người dùng đều được phát triển trong thư mục này.

* + **components**: Thư mục này chứa các thành phần React (components) tái sử dụng được trong toàn bộ ứng dụng. Mỗi thành phần có thể là một UI độc lập hoặc một phần của UI lớn hơn.
  + **hooks**: Chứa các custom hooks của React, giúp tái sử dụng các logic trạng thái và hiệu ứng trong các thành phần khác nhau của ứng dụng.
  + **pages**: Chứa các trang của ứng dụng, nơi mỗi tệp đại diện cho một trang riêng biệt trong ứng dụng. Các trang này được xây dựng từ các thành phần con.
  + **redux**: Nếu ứng dụng sử dụng Redux để quản lý trạng thái toàn cục, thư mục này chứa các tệp liên quan đến cấu hình Redux, như reducers, actions và store.
  + **styles**: Chứa các tệp CSS hoặc các tệp cấu hình giao diện, bao gồm các class và style để thiết kế giao diện người dùng. Các tệp này có thể sử dụng CSS Modules hoặc các phương pháp tổ chức khác.
  + **utils**: Thư mục này chứa các hàm tiện ích (utility functions) giúp tái sử dụng các đoạn mã trong toàn bộ ứng dụng, chẳng hạn như xử lý chuỗi, định dạng ngày tháng, v.v.

1. **App.js**:

Đây là thành phần chính của ứng dụng React, nơi quản lý toàn bộ cấu trúc của ứng dụng và các route (nếu có) để điều hướng giữa các trang.

1. **index.js**:

Tệp này là điểm vào của ứng dụng, nơi React được khởi tạo và gắn vào tệp HTML chính. Nó cũng liên kết các thành phần gốc của ứng dụng vào DOM.

1. **.gitignore**:

Tệp này xác định những tệp và thư mục nào sẽ bị bỏ qua khi sử dụng Git để kiểm soát phiên bản. Thường bao gồm node\_modules và các tệp môi trường hoặc build.

1. **package-lock.json**:

Tệp này được tạo tự động khi cài đặt các gói phụ thuộc thông qua NPM. Nó giúp đảm bảo rằng tất cả các thành viên trong nhóm phát triển và môi trường triển khai đều sử dụng các phiên bản gói phụ thuộc giống nhau.

1. **package.json**:

Tệp cấu hình chính của dự án, chứa thông tin về các gói phụ thuộc, các script (ví dụ: npm start, npm test) và các metadata khác của dự án.

1. **postcss.config.js**:

Tệp cấu hình cho PostCSS, một công cụ dùng để xử lý CSS, chẳng hạn như tối ưu hóa hoặc thêm các tính năng mới vào CSS.

1. **README.md**:

Tệp này chứa tài liệu mô tả về dự án, các bước hướng dẫn cài đặt, cấu hình và sử dụng, cũng như các thông tin quan trọng khác cho các nhà phát triển khác.

HƯỚNG DẪN XEM ĐỒ ÁN

Để chạy dự án cần thực hiện các bước sau đây:

1. **Truy Cập Kho Lưu Trữ GitHub**:

* Mở kho lưu trữ dự án từ đường link sau: <https://github.com/miavn9x/Do_An_Du_Thi_Web_Desige>.

1. **Tải Mã Nguồn Về Máy Tính**:

* Bạn có thể tải mã nguồn về máy tính của mình bằng cách nhấn vào nút **Code** và chọn **Download ZIP**. Sau đó, giải nén tệp ZIP và truy cập vào thư mục chứa mã nguồn.
* Nếu bạn sử dụng Git, có thể clone dự án bằng lệnh sau:

git clone https://github.com/miavn9x/Do\_An\_Du\_Thi\_Web\_Desige.git

1. **Cài Đặt và Chạy Dự Án Backend**:

* Trước tiên, chuyển đến thư mục backend:
  + cd backend
* Cài đặt các phụ thuộc của backend bằng lệnh:
  + npm install
* Sau khi cài đặt xong, khởi chạy backend với lệnh:
  + npm start
* Backend sẽ chạy và có thể truy cập tại địa chỉ mặc định: http://localhost:5001.

1. **Cài Đặt và Chạy Dự Án Frontend**:

* Tiếp theo, chuyển đến thư mục frontend:
  + cd frontend
* Cài đặt các phụ thuộc của frontend bằng lệnh:
  + npm install
* Sau khi cài đặt xong, bạn có thể khởi chạy ứng dụng frontend với lệnh:
  + npm start
* Giao diện frontend sẽ được mở ra trong trình duyệt tại địa chỉ: <http://localhost:3000>.

1. **Demo dự án lên internet:**<https://do-an-du-thi-web-desige.onrender.com/>

KẾT LUẬN

Đồ án "Xây dựng hệ thống website thương mại điện tử Baby chill", chúng tôi đã hoàn thành việc thiết kế và triển khai một hệ thống bán hàng trực tuyến phục vụ nhu cầu mua sắm các sản phẩm cho trẻ em. Các chức năng cơ bản của website như quản lý sản phẩm, giỏ hàng, thanh toán và AI hộ trợ chăm sóc khách hàng đã được xây dựng và tích hợp thành công, đảm bảo hiệu quả sử dụng và đáp ứng mục tiêu của dự án.

Tuy nhiên, để nâng cao chất lượng và khả năng mở rộng của hệ thống trong tương lai, chúng tôi đề xuất một số hướng phát triển quan trọng. Đầu tiên, việc xây dựng một trang quản trị viên chuyên biệt sẽ giúp người quản trị dễ dàng theo dõi và quản lý dữ liệu như sản phẩm, đơn hàng, khách hàng, và các giao dịch một cách trực quan và hiệu quả. Thứ hai, tích hợp trí tuệ nhân tạo vào hệ thống có thể cải thiện trải nghiệm người dùng, với các tính năng như gợi ý sản phẩm dựa trên thói quen mua sắm của khách hàng. Cuối cùng, việc áp dụng các thuật toán cá nhân hóa sẽ tạo ra những trải nghiệm mua sắm độc đáo, phù hợp với từng khách hàng, từ đó gia tăng sự hài lòng và trung thành.

Đồng thời, tối ưu hóa hiệu suất và bảo mật hệ thống cũng sẽ là yếu tố quan trọng giúp duy trì hoạt động ổn định và an toàn trong tương lai. Những đề xuất trên không chỉ giúp website Baby Mart nâng cao chất lượng dịch vụ mà còn mở ra cơ hội phát triển bền vững trong môi trường thương mại điện tử ngày càng cạnh tranh.

DANH MỤC THAM KHẢO

1. **Express.js Documentation (n.d.)** Express.js: Fast, unopinionated, minimalist web framework for Node.js. Available at: <https://expressjs.com/> (Accessed: 16 February 2025).
2. **React.js Documentation (n.d.)** React: A JavaScript library for building user interfaces. Available at: <https://react.dev/learn> (Accessed: 16 February 2025).
3. **MongoDB Documentation (n.d.)** MongoDB: The database for modern applications. Available at: <https://www.mongodb.com/docs/> (Accessed: 16 February 2025).
4. **AI botpress :** <https://www.botpress.com/docs/home>