TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI

KHOA CÔNG NGHỆ THÔNG TIN

Diagram

Description automatically generated with low confidence

BÁO CÁO ĐỒ ÁN CHUYÊN NGÀNH CNTT

MÃ HỌC PHẦN: IT6122001

Đề tài: Xây dựng website bán quần áo cho cửa hàng TrueMart

|  |
| --- |
|  |

GVHD: TS. Phạm Văn Hiệp

Nhóm sinh viên thực hiện:

1. Hoàng Văn Lâm - 2021607668
2. Chu Quốc Khánh - 2021604521
3. Nguyễn Thanh Lâm - 2021607814
4. Phan Văn Khánh - 2021607067

Mã lớp: 2024IT6122001

Khoá: 16 Nhóm: 15

Hà Nội – Năm 2024

**CHƯƠNG 1: GIỚI THIỆU TỔNG QUAN**

* 1. **Lý do chọn đề tài**

Do hiện nay các cửa hàng, công ty về các loại quần áo chưa được phát triển và quảng bá nhiều trên nền tảng website online để kinh doanh và nhu cầu mua sắm online đang rất được ưa chuộng.

Vì vậy em đã chọn làm đề tài này nhằm đưa ra giải pháp tiện lợi, nhanh gọn trong việc kinh doanh các sản phẩm của các cửa hàng hoặc công ty có thể quảng bá và bán các loại sản phẩm của mình trên mạng online một cách đơn giản mà không cần tốn quá nhiều thời gian chi phí của khách hàng cũng như của chủ các cửa hàng.

* 1. **Giới thiệu chung về hệ thống website bán quần áo cho cửa hàng TrueMart**

Website bán quần áo cho cửa hàng TrueMart là một nền tảng thương mại điện tử được thiết kế nhằm phục vụ nhu cầu mua sắm trực tuyến của khách hàng. Với xu hướng ngày càng gia tăng của việc mua sắm trực tuyến, việc xây dựng một website chuyên nghiệp và dễ sử dụng là cần thiết để nâng cao khả năng cạnh tranh và mở rộng thị trường cho công ty.

**Chức năng chính của website bao gồm:**

* **Giới thiệu sản phẩm:** Cung cấp thông tin chi tiết về các sản phẩm quần áo, bao gồm hình ảnh, mô tả, giá cả và kích thước. Khách hàng có thể dễ dàng tìm kiếm và lựa chọn sản phẩm mình yêu thích.
* **Quản lý giỏ hàng:** Khách hàng có thể thêm các sản phẩm vào giỏ hàng, xem lại danh sách sản phẩm đã chọn và điều chỉnh số lượng trước khi tiến hành thanh toán.
* **Thanh toán trực tuyến:** Cung cấp các phương thức thanh toán linh hoạt và an toàn như thẻ tín dụng, thanh toán khi nhận hàng.
* **Quản lý đơn hàng:** Cho phép khách hàng theo dõi tình trạng đơn hàng của mình, từ khi đặt hàng đến khi giao hàng.

**Thiết kế giao diện:** Website sẽ được thiết kế với giao diện thân thiện, dễ dàng điều hướng và tương thích với các thiết bị thông minh. Màu sắc và phong cách thiết kế sẽ phản ánh thương hiệu của cửa hàng TrueMart, tạo cảm giác gần gũi và dễ tiếp cận cho khách hàng.

**Tầm quan trọng của website:** Việc phát triển website không chỉ giúp cửa hàng TrueMart nâng cao khả năng tiếp cận với khách hàng mà còn cải thiện hình ảnh thương hiệu và tăng cường sự chuyên nghiệp trong hoạt động kinh doanh. Đồng thời, website cũng là công cụ hữu ích để thu thập thông tin và phản hồi từ khách hàng, giúp công ty cải thiện chất lượng sản phẩm và dịch vụ.

**1.3 Nghiên cứu hiện trạng và giải quyết vấn đề**

***1.3.1. Hiện trạng của cửa hàng.***

Cửa hàng TrueMart là một cửa hàng bán quần áo thời trang với nhiều mặt hàng đa dạng, từ quần áo nam, nữ đến phụ kiện thời trang. Hiện tại, TrueMart chủ yếu bán hàng trực tiếp tại cửa hàng và thông qua mạng xã hội. Tuy nhiên, cửa hàng chưa có website chính thức để mở rộng kênh bán hàng trực tuyến, gia tăng sự hiện diện thương hiệu và tối ưu trải nghiệm mua sắm của khách hàng.

**Các vấn đề hiện tại:**

* Chưa có nền tảng bán hàng trực tuyến chuyên nghiệp.
* Khó quản lý hàng tồn kho, đơn hàng và dữ liệu khách hàng một cách hiệu quả.
* Hạn chế khả năng tiếp cận khách hàng ở các khu vực xa.
* Không thể tối ưu hóa trải nghiệm khách hàng khi mua sắm online.

***1.3.2. Cơ cấu tổ chức.***

Cửa hàng TrueMart có cơ cấu tổ chức nhỏ gọn với các bộ phận chính như sau:

* **Quản lý cửa hàng:** Đảm nhận quản lý chung về hoạt động, kinh doanh và phát triển.
* **Bộ phận bán hàng:** Gồm nhân viên bán hàng tại cửa hàng và hỗ trợ bán hàng qua các nền tảng mạng xã hội.
* **Bộ phận kho hàng:** Đảm bảo việc nhập hàng, kiểm kê và quản lý tồn kho.
* **Bộ phận marketing:** Chịu trách nhiệm quảng cáo và truyền thông trên các kênh mạng xã hội.
* **Bộ phận chăm sóc khách hàng:** Hỗ trợ giải đáp thắc mắc, xử lý đơn hàng và bảo hành sản phẩm.

***1.3.3. Mô tả hoạt động của các bộ phận.***

* **Bán hàng:** Tiếp nhận và xử lý đơn hàng của khách qua mạng xã hội và tại cửa hàng, đồng thời tư vấn sản phẩm cho khách hàng.
* **Kho hàng:** Kiểm tra số lượng hàng tồn kho, theo dõi đơn hàng và điều phối việc giao nhận hàng hóa.
* **Marketing:** Lên kế hoạch và thực hiện các chiến dịch quảng cáo qua Facebook, Instagram, và TikTok để tăng cường độ nhận diện thương hiệu.
* **Chăm sóc khách hàng:** Tiếp nhận các câu hỏi và phản hồi từ khách hàng, hỗ trợ xử lý đổi trả, khiếu nại và theo dõi phản hồi sau mua hàng.

***1.3.4. Xác định yêu cầu.***

Yêu cầu chính cho website bán quần áo:

* **Giao diện người dùng thân thiện:** Dễ sử dụng trên cả máy tính và thiết bị di động, có chức năng lọc sản phẩm theo loại, kích thước, giá cả, màu sắc.
* **Chức năng giỏ hàng:** Khách hàng có thể thêm sản phẩm vào giỏ hàng, xem lại và điều chỉnh trước khi tiến hành thanh toán.
* **Thanh toán trực tuyến:** Tích hợp các phương thức thanh toán qua thẻ tín dụng, ví điện tử (Momo, ZaloPay) và chuyển khoản ngân hàng.
* **Quản lý đơn hàng:** Hệ thống cho phép quản lý và theo dõi đơn hàng, kiểm tra lịch sử giao dịch của khách hàng.
* **Tối ưu SEO:** Đảm bảo website được tối ưu hóa cho các công cụ tìm kiếm để tăng lưu lượng truy cập.
* **Tính năng đăng ký và quản lý tài khoản khách hàng:** Khách hàng có thể tạo tài khoản, theo dõi đơn hàng, lịch sử mua sắm và nhận thông báo về chương trình khuyến mãi.
* **Tích hợp quản lý tồn kho:** Website cần có hệ thống để quản lý hàng tồn kho theo thời gian thực.

**Mục tiêu:**

* Tạo ra một kênh bán hàng trực tuyến hiệu quả, nâng cao khả năng tiếp cận khách hàng và gia tăng doanh thu cho cửa hàng TrueMart

**1.4 Giới thiệu về công nghệ, ngôn ngữ sử dụng**

**1.4.1. Giới thiệu về ASP.NET**

ASP.NET là một framework do Microsoft phát triển, giúp xây dựng các ứng dụng web và dịch vụ web mạnh mẽ. Đây là một phần của .NET framework và cho phép lập trình viên phát triển các ứng dụng dựa trên web một cách dễ dàng và hiệu quả. Dưới đây là các khái niệm và tính năng cơ bản của ASP.NET:

**1.4.1.1. Cấu trúc MVC (Model-View-Controller)**

ASP.NET hỗ trợ mô hình MVC, là một kiến trúc phân tách rõ ràng giữa ba thành phần:

* Model: Đại diện cho dữ liệu và logic xử lý dữ liệu.
* View: Hiển thị giao diện người dùng và tương tác với người dùng.
* Controller: Xử lý yêu cầu từ người dùng, điều khiển dữ liệu và trả về phản hồi (kết quả).

Kiến trúc này giúp việc quản lý, bảo trì và mở rộng ứng dụng trở nên dễ dàng hơn.

**1.4.1.2. Razor Pages**

Đây là một cách tiếp cận mới trong ASP.NET, giúp tạo các trang web theo cấu trúc "page-based". Razor Pages là một mô hình đơn giản hơn so với MVC và phù hợp với các ứng dụng web nhỏ hoặc có tính năng không quá phức tạp.

**1.4.1.3. ASP.NET Web API**

ASP.NET cũng cung cấp các công cụ để xây dựng API dịch vụ web (RESTful), cho phép ứng dụng tương tác với các hệ thống khác thông qua HTTP. Web API dễ dàng tích hợp với nhiều ứng dụng khác nhau, từ mobile apps đến desktop apps.

**1.4.1.4. SignalR**

SignalR là một thư viện trong ASP.NET giúp xây dựng các ứng dụng thời gian thực như chat, thông báo, hay các ứng dụng yêu cầu cập nhật dữ liệu liên tục mà không cần người dùng phải làm mới trang.

**1.4.1.5. Bảo mật**

ASP.NET có các cơ chế bảo mật mạnh mẽ như xác thực (Authentication) và phân quyền (Authorization), giúp bảo vệ ứng dụng khỏi các tấn công như SQL Injection, Cross-Site Scripting (XSS), Cross-Site Request Forgery (CSRF),...

**1.4.1.6. Hiệu năng cao**

ASP.NET Core, phiên bản mới của ASP.NET, đã được tối ưu hóa để có hiệu năng vượt trội hơn so với ASP.NET cũ. Nhờ đó, các ứng dụng web có thể xử lý lượng lớn người dùng với tốc độ nhanh hơn.

**1.4.1.7. Đa nền tảng**

ASP.NET Core cho phép phát triển ứng dụng trên nhiều nền tảng khác nhau như Windows, macOS và Linux, tạo điều kiện thuận lợi cho việc triển khai ứng dụng trên các môi trường khác nhau.

**1.4.1.8. Quản lý trạng thái (State Management)**

ASP.NET cung cấp nhiều cơ chế quản lý trạng thái, như ViewState, Session, Cache, và Cookies, giúp lưu trữ và theo dõi thông tin của người dùng khi truy cập vào ứng dụng.

* + 1. **Giới thiệu về mô hình MVC**

Mô hình MVC (Model-View-Controller) là một kiến trúc phần mềm giúp tách biệt các phần của ứng dụng web.

* Model: Chịu trách nhiệm xử lý dữ liệu, quản lý logic liên quan đến dữ liệu và kết nối với cơ sở dữ liệu. Trong ứng dụng sử dụng Node.js, Mongoose thường được dùng để tương tác với cơ sở dữ liệu MongoDB.
* View: Là phần giao diện người dùng, hiển thị thông tin và nhận input từ người dùng. Pug (trước đây là Jade) được sử dụng làm engine để render các view động dựa trên dữ liệu từ Model.
* Controller: Xử lý các yêu cầu từ người dùng, gọi đến Model để lấy dữ liệu cần thiết và chuyển chúng đến View. Express.js đóng vai trò là bộ xử lý request và định tuyến giữa các phần của ứng dụng.
  + 1. **Tổng quan về ASP.NET MVC**

ASP.NET MVC là một framework giúp phát triển ứng dụng web bằng cách áp dụng mô hình kiến trúc **Model-View-Controller** (MVC). Đây là một trong những mô hình phổ biến nhất để tách biệt logic xử lý và giao diện của ứng dụng, giúp tăng cường tính linh hoạt, dễ bảo trì và mở rộng trong quá trình phát triển. ASP.NET MVC được Microsoft phát triển như là một phần mở rộng của ASP.NET.

**1.4.3.1. Cấu trúc Model-View-Controller (MVC):**

Mô hình MVC phân chia ứng dụng thành ba thành phần chính:

* **Model**: Quản lý dữ liệu và logic kinh doanh. Model chịu trách nhiệm giao tiếp với cơ sở dữ liệu hoặc các dịch vụ bên ngoài và xử lý các dữ liệu được yêu cầu bởi Controller hoặc View.
* **View**: Chịu trách nhiệm hiển thị giao diện cho người dùng. View nhận dữ liệu từ Model thông qua Controller và định dạng chúng thành HTML để hiển thị.
* **Controller**: Điều khiển luồng hoạt động của ứng dụng. Controller xử lý các yêu cầu từ người dùng, tương tác với Model và trả về dữ liệu thông qua View.

Sự phân chia này giúp cho ứng dụng dễ dàng phát triển, kiểm thử và bảo trì bởi các thành phần hoạt động độc lập và không bị phụ thuộc lẫn nhau.

**1.4.3.2. Tính năng của ASP.NET MVC**

ASP.NET MVC cung cấp nhiều tính năng nổi bật giúp hỗ trợ phát triển ứng dụng web một cách hiệu quả:

* **Routing**: Cho phép định nghĩa các URL thân thiện với người dùng. ASP.NET MVC sử dụng một hệ thống routing linh hoạt, giúp chuyển đổi các URL thành các yêu cầu cho controller cụ thể.
* **Data Binding và Model Validation**: ASP.NET MVC hỗ trợ việc liên kết dữ liệu từ các form trong View với các model trong Controller một cách tự động. Ngoài ra, nó cung cấp các cơ chế để xác thực dữ liệu (Data Annotation), đảm bảo dữ liệu đầu vào của người dùng luôn đúng và hợp lệ.
* **Templating và Scaffolding**: ASP.NET MVC hỗ trợ công cụ Scaffolding giúp tự động tạo ra mã cho các phần CRUD (Create, Read, Update, Delete) của ứng dụng dựa trên Model. Tính năng này giúp tiết kiệm thời gian khi phát triển các ứng dụng đơn giản.
* **View Engine (Razor)**: ASP.NET MVC sử dụng Razor như là một công cụ hiển thị View mạnh mẽ, cho phép chèn mã C# trực tiếp vào các trang HTML thông qua cú pháp ngắn gọn @. Điều này giúp giảm thiểu lỗi và tăng tính linh hoạt trong việc xử lý dữ liệu và giao diện người dùng.

**1.4.3.3. Quá trình xử lý yêu cầu (Request Processing Flow) trong ASP.NET MVC**

Khi một yêu cầu được gửi từ trình duyệt đến ứng dụng ASP.NET MVC, nó trải qua các bước sau:

1. **Request**: Người dùng truy cập một URL, yêu cầu này sẽ được hệ thống routing trong ASP.NET MVC phân tích và chuyển hướng đến controller tương ứng.
2. **Controller**: Controller nhận yêu cầu, xử lý logic và tương tác với Model để lấy dữ liệu.
3. **Model**: Nếu cần, Model sẽ truy xuất dữ liệu từ cơ sở dữ liệu hoặc các nguồn khác và trả kết quả về Controller.
4. **View**: Controller chuyển dữ liệu tới View, nơi dữ liệu được định dạng và gửi lại kết quả (HTML) cho người dùng.
5. **Response**: Kết quả từ View được trả về cho người dùng dưới dạng trang web.

**1.4.3.4. Ưu điểm của ASP.NET MVC**

* **Tách biệt rõ ràng giữa các thành phần**: Việc tách riêng logic xử lý (Model), giao diện (View) và điều khiển (Controller) giúp dễ bảo trì, mở rộng và kiểm thử ứng dụng.
* **Hỗ trợ kiểm thử**: Với cấu trúc rõ ràng, ASP.NET MVC hỗ trợ mạnh mẽ việc viết các unit test để kiểm thử từng phần của ứng dụng.
* **Kiểm soát tốt hơn về HTML, CSS, và JavaScript**: ASP.NET MVC cho phép lập trình viên có toàn quyền kiểm soát việc tạo HTML, thay vì dựa hoàn toàn vào các công cụ như WebForms, giúp tối ưu hóa SEO và trải nghiệm người dùng.
* **Công cụ Scaffolding và Razor Engine**: Giúp giảm thời gian phát triển bằng cách tự động tạo mã và cung cấp một cách viết mã đơn giản hơn.
  + 1. **Cơ sở dữ liệu sử dụng**

**1.4.4.1. Giới thiệu chung**

MySQL là một trong những hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) phổ biến nhất hiện nay. Được phát triển vào đầu những năm 1990, MySQL được sử dụng rộng rãi trong các ứng dụng web, từ các trang blog cá nhân đến các ứng dụng thương mại điện tử lớn. MySQL nổi bật với hiệu suất cao, tính ổn định và khả năng mở rộng, giúp đáp ứng nhu cầu lưu trữ và truy xuất dữ liệu của nhiều loại ứng dụng khác nhau.

**1.4.4.2. Các đặc điểm nổi bật của MySQL**

* **Miễn phí và mã nguồn mở**: MySQL là một phần mềm mã nguồn mở, cho phép người dùng tự do sử dụng, chỉnh sửa và phân phối. Điều này giúp giảm chi phí cho các doanh nghiệp và cá nhân khi triển khai các ứng dụng cần cơ sở dữ liệu.
* **Hiệu suất cao**: MySQL được tối ưu hóa để xử lý một khối lượng lớn dữ liệu và hỗ trợ nhiều kết nối đồng thời. Điều này giúp cải thiện tốc độ truy vấn và giảm thời gian phản hồi.
* **Khả năng mở rộng**: MySQL có khả năng mở rộng linh hoạt, cho phép người dùng mở rộng dung lượng lưu trữ và tăng cường hiệu suất khi cần thiết. Nó hỗ trợ các cấu hình từ máy chủ đơn giản đến các cụm máy chủ phức tạp.
* **Hỗ trợ ACID**: MySQL đảm bảo tính nhất quán và độ tin cậy của dữ liệu thông qua các giao dịch ACID (Atomicity, Consistency, Isolation, Durability). Điều này có nghĩa là các giao dịch được thực hiện đầy đủ hoặc không thực hiện chút nào, đảm bảo rằng dữ liệu luôn ở trạng thái hợp lệ.
* **Tính bảo mật**: MySQL cung cấp nhiều cơ chế bảo mật, bao gồm xác thực người dùng, phân quyền truy cập và mã hóa dữ liệu, giúp bảo vệ thông tin nhạy cảm trong cơ sở dữ liệu.

**1.4.4.3. Ứng dụng của MySQL**

MySQL được sử dụng trong nhiều lĩnh vực và ứng dụng, bao gồm:

* **Web Applications**: MySQL là lựa chọn phổ biến cho các ứng dụng web, từ hệ thống quản lý nội dung (CMS) đến các trang thương mại điện tử.
* **Hệ thống quản lý**: Nhiều hệ thống quản lý thông tin, như hệ thống quản lý khách hàng (CRM) và hệ thống quản lý tài sản, sử dụng MySQL để lưu trữ và quản lý dữ liệu.
* **Dự án mã nguồn mở**: Nhiều dự án mã nguồn mở, như WordPress, Joomla và Drupal, sử dụng MySQL làm cơ sở dữ liệu mặc định.

**1.4.4.4. Lợi ích của việc sử dụng MySQL**

* **Chi phí thấp**: Là một giải pháp mã nguồn mở, MySQL giúp giảm chi phí triển khai cho doanh nghiệp, đặc biệt là cho các dự án nhỏ và vừa.
* **Dễ sử dụng**: MySQL cung cấp một giao diện người dùng thân thiện và nhiều công cụ hỗ trợ, giúp người dùng dễ dàng thiết lập, quản lý và truy vấn dữ liệu.
* **Cộng đồng hỗ trợ**: Với một cộng đồng người dùng lớn, MySQL có nhiều tài liệu, diễn đàn và hướng dẫn hỗ trợ, giúp người dùng dễ dàng tìm kiếm thông tin và giải pháp cho các vấn đề gặp phải.

MySQL là một giải pháp cơ sở dữ liệu mạnh mẽ, linh hoạt và dễ sử dụng, phù hợp cho nhiều loại ứng dụng khác nhau. Với hiệu suất cao, tính ổn định và khả năng mở rộng, MySQL đã chứng tỏ mình là một trong những lựa chọn hàng đầu cho việc quản lý dữ liệu trong môi trường phát triển hiện đại. Đối với các doanh nghiệp và lập trình viên, việc lựa chọn MySQL cho các dự án của mình sẽ mang lại nhiều lợi ích và giá trị gia tăng.

**CHƯƠNG 2: KẾT QUẢ NGHIÊN CỨU**

2.1 Phân tích hệ thống

* Mô tả nghiệp vụ
* Yêu cầu chức năng, phi chức năng

2.2 Xác định tác nhân và ca sử dụng (Usecase)

Các tác nhân

Usecase tổng quan, phân rã

2.3 Biểu đồ Usecase

2.4 Mô tả chi tiết Usecase

Đăng nhập, đăng ký, tìm kiếm, quản lý giỏ hàng, thanh toán

2.5 Thiết kế hệ thống

* Biểu đồ lớp, biểu đồ tuần tự
* Sơ đồ dữ liệu quan hệ
* Chi tiết bảng dữ liệu

2.6 Giao diện hệ thống

* Front-end: Trang chủ, đăng ký, danh sách sản phẩm, giỏ hàng, kiểm tra đơn hàng
* Back-end: Quản lý danh mục, sản phẩm, tài khoản, đơn hàng

**CHƯƠNG 3: KẾT LUẬN VÀ BÀI HỌC KINH NGHIỆM**

3.1 Kiến thức và kỹ năng đã học

3.2 Bài học kinh nghiệm

3.3 Tính khả thi của đề tài

Thuận lợi và khó khăn

**TÀI LIỆU THAM KHẢO**