**ĐỀ CƯƠNG ĐỒ ÁN TỐT NGHIỆP**

**ĐỀ TÀI: XÂY DỰNG WEBSITE BÁN QUẦN ÁO TRẺ EM CHO**

**CỬA HÀNG TEEMART**

**MỞ ĐẦU**

**1. Lý do chọn đề tài**

Hiện nay, nhiều cửa hàng và công ty kinh doanh quần áo vẫn chưa tận dụng triệt để nền tảng website để phát triển thương hiệu và mở rộng thị trường. Trong khi đó, xu hướng mua sắm trực tuyến đang ngày càng phổ biến, mang lại nhiều lợi ích như tiết kiệm thời gian, đa dạng lựa chọn và mua sắm thuận tiện mọi lúc, mọi nơi. Tuy nhiên, do chưa có sự đầu tư bài bản vào các kênh bán hàng trực tuyến, nhiều doanh nghiệp vẫn gặp khó khăn trong việc tiếp cận khách hàng tiềm năng và tối ưu hóa doanh thu.

Chính vì vậy, em quyết định chọn đề tài “Xây dựng website bán quần áo trẻ em cho cửa hàng TEEMART” nhằm nghiên cứu và đề xuất một giải pháp hiệu quả giúp các cửa hàng và công ty có thể dễ dàng quảng bá và kinh doanh sản phẩm của mình trên nền tảng online. Giải pháp này không chỉ hướng đến sự đơn giản, dễ sử dụng mà còn giúp tối ưu chi phí vận hành, rút ngắn thời gian quản lý, đồng thời nâng cao trải nghiệm mua sắm cho khách hàng. Với hệ thống này, doanh nghiệp có thể tiếp cận nhiều khách hàng hơn mà không cần đầu tư quá nhiều vào các phương thức kinh doanh truyền thống, mở ra cơ hội phát triển bền vững trong thời đại số hóa.

**2. Mục đích**

Đề tài ‘Xây dựng website bán quần áo trẻ em cho cửa hàng TEEMART’ được thực hiện với mục tiêu xây dựng một nền tảng bán hàng trực tuyến hiện đại, hiệu quả, giúp cửa hàng tiếp cận khách hàng tiềm năng và mở rộng thị trường. Cụ thể, đề tài hướng đến các mục tiêu sau:

* Tạo ra một website bán hàng chuyên nghiệp giúp cửa hàng TEEMART dễ dàng giới thiệu, trưng bày sản phẩm, nâng cao độ nhận diện thương hiệu trên nền tảng số.
* Tối ưu hóa trải nghiệm mua sắm trực tuyến bằng cách thiết kế giao diện thân thiện, dễ sử dụng, giúp khách hàng thuận tiện trong việc tìm kiếm, lựa chọn và đặt hàng.
* Hỗ trợ quản lý cửa hàng hiệu quả thông qua các tính năng như quản lý sản phẩm, đơn hàng, khách hàng, giúp giảm bớt thời gian và chi phí vận hành.
* Tích hợp các phương thức thanh toán và vận chuyển linh hoạt, tạo điều kiện thuận lợi cho cả người mua và người bán, góp phần gia tăng doanh thu.
* Ứng dụng công nghệ để nâng cao khả năng cạnh tranh, giúp cửa hàng TEEMART thích nghi với xu hướng mua sắm trực tuyến, từ đó phát triển bền vững trong thị trường kinh doanh thời trang trẻ em.

**3. Đối tượng và phạm vi nghiên cứu**

**Đối tượng nghiên cứu**

* Website bán hàng trực tuyến trong lĩnh vực thời trang, đặc biệt là quần áo trẻ em.
* Các tính năng và công nghệ hỗ trợ cho hoạt động thương mại điện tử như quản lý sản phẩm, đơn hàng, khách hàng, thanh toán và vận chuyển.
* Trải nghiệm người dùng (UX/UI) nhằm tối ưu hóa quá trình mua sắm trực tuyến.
* Các phương pháp tối ưu hóa chi phí vận hành và quản lý hệ thống hiệu quả.

**Phạm vi nghiên cứu**

* Phạm vi nội dung: Nghiên cứu và xây dựng một website bán quần áo trẻ em cho cửa hàng TEEMART với các tính năng cơ bản như hiển thị sản phẩm, giỏ hàng, thanh toán, quản lý đơn hàng và tài khoản khách hàng.
* Phạm vi công nghệ: Sử dụng các công nghệ web hiện đại như HTML, CSS, JavaScript, ASP.NET MVC kết hợp với cơ sở dữ liệu để lưu trữ và quản lý thông tin.
* Phạm vi không gian: Áp dụng cho cửa hàng TEEMART và khách hàng có nhu cầu mua sắm trực tuyến trong phạm vi nội địa, chủ yếu tập trung vào thị trường Việt Nam.
* Phạm vi thời gian: Đề tài được thực hiện trong một khoảng thời gian nhất định, từ quá trình nghiên cứu, thiết kế, lập trình đến thử nghiệm và đánh giá hệ thống.

**4. Ý nghĩa khoa học và thực tiễn của đề tài.**

**Ý nghĩa khoa học**

* Đề tài nghiên cứu các mô hình thương mại điện tử hiện đại, từ đó áp dụng các phương pháp và công nghệ lập trình phù hợp để phát triển một hệ thống bán hàng trực tuyến hiệu quả.
* Cung cấp kiến thức về thiết kế, phát triển và tối ưu hóa website thương mại điện tử, góp phần vào việc ứng dụng công nghệ thông tin trong kinh doanh.
* Ứng dụng các nguyên tắc thiết kế giao diện người dùng (UI) và trải nghiệm người dùng (UX) để nâng cao hiệu suất và tính tiện dụng của hệ thống.
* Đề tài có thể làm tài liệu tham khảo cho các nghiên cứu sau này về xây dựng hệ thống thương mại điện tử trong lĩnh vực bán lẻ.

**Ý nghĩa thực tiễn**

* Giúp cửa hàng TEEMART tiếp cận mô hình kinh doanh trực tuyến, mở rộng thị trường và gia tăng doanh thu.
* Cung cấp một nền tảng bán hàng chuyên nghiệp, giúp cửa hàng quản lý sản phẩm, đơn hàng và khách hàng một cách dễ dàng, tiết kiệm thời gian và chi phí vận hành.
* Nâng cao trải nghiệm mua sắm cho khách hàng với giao diện thân thiện, dễ sử dụng, tích hợp nhiều phương thức thanh toán và vận chuyển linh hoạt.
* Góp phần thúc đẩy xu hướng chuyển đổi số trong lĩnh vực kinh doanh quần áo, giúp doanh nghiệp thích nghi với sự phát triển của thương mại điện tử.

**MỤC LỤC**

**DANH MỤC HÌNH ẢNH**

**DANH MỤC BẢNG BIỂU**

**CHƯƠNG 1: GIỚI THIỆU TỔNG QUAN**

**1.1. Giới thiệu chung về website**

Website bán quần áo cho cửa hàng TEEMART 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 TEEMART, 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 TEEMART 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.

**1.2. Nghiên cứu hiện trạng và giải quyết vấn đề**

**1.2.1. Hiện trạng của cửa hàng**

Cửa hàng TEEMART 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, TEEMART 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.

**1.2.2. Cơ cấu tổ chức**

Cửa hàng TEEMART 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.

**1.2.3. Mô tả hoạt động các bộ phận**

Cơ cấu tổ chức của TEEMART bao gồm các bộ phận chính như:

* 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.

**1.2.4. Xác định yêu cầu hệ thống**

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 TEEMART

**1.3. Giới thiệu về công nghệ và ngôn ngữ sử dụng**

**1.3.1. 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:

* *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.

* *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.

*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.

* *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.

* *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),...

* *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.

* *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.

* *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.

**1.3.2. Giới thiệu về mô hình MVC**

MVC (Model-View-Controller) là một kiến trúc phần mềm được sử dụng rộng rãi trong phát triển ứng dụng web, giúp phân chia rõ ràng các thành phần của ứng dụng thành ba phần chính: Model (Mô hình), View (Giao diện hiển thị), và Controller (Điều khiển). Mô hình này giúp tối ưu hóa việc phát triển, quản lý và bảo trì mã nguồn, đặc biệt là với các dự án phức tạp và lớn.

• Model: Là thành phần chịu trách nhiệm quản lý dữ liệu và logic nghiệp vụ của ứng dụng. Model nhận dữ liệu từ cơ sở dữ liệu hoặc các nguồn khác và chuyển dữ liệu đến Controller khi cần. Ngoài ra, Model cũng đảm nhiệm việc xử lý các thao tác, tính toán và xác thực dữ liệu.

• View: Là thành phần hiển thị giao diện người dùng, chịu trách nhiệm hiển thị dữ liệu từ Model lên màn hình. View sẽ chỉ làm nhiệm vụ trình bày và không thực hiện bất kỳ logic nào khác, giúp tách biệt phần giao diện và xử lý dữ liệu.

• Controller: Là thành phần trung gian, đóng vai trò điều phối giữa Model và View. Khi người dùng thực hiện một hành động (như gửi biểu mẫu hoặc nhấp chuột), Controller sẽ tiếp nhận yêu cầu, tương tác với Model để lấy dữ liệu, và sau đó chọn View phù hợp để hiển thị dữ liệu đó.

Ưu điểm của mô hình MVC:

• Tách biệt rõ ràng các thành phần: Giúp dễ dàng bảo trì, mở rộng và tái sử dụng mã nguồn.

• Hỗ trợ phân công công việc: Các nhà phát triển có thể làm việc đồng thời trên các phần khác nhau của ứng dụng mà không lo bị xung đột.

• Tăng hiệu suất và tổ chức mã: MVC giúp giữ cho mã nguồn rõ ràng và có tổ chức, đồng thời giúp kiểm tra và sửa lỗi dễ dàng hơn.

MVC không chỉ áp dụng cho các ứng dụng web mà còn có thể được sử dụng trong nhiều loại ứng dụng khác. Đặc biệt, khi kết hợp với các framework như ASP.NET, mô hình MVC giúp tăng cường khả năng phát triển và mở rộng ứng dụng một cách linh hoạt và hiệu quả. Nhiều tệp tin và cấu trúc phức tạp: Mô hình MVC tạo ra nhiều tệp tin và yêu cầu tổ chức mã chặt chẽ, có thể khiến việc quản lý khó khăn trong các dự án lớn nếu không được cấu trúc tốt

**1.3.3. 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.

* *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.

* *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.
* *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:

* 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.
* Controller: Controller nhận yêu cầu, xử lý logic và tương tác với Model để lấy dữ liệu.
* Model: Nếu cần, Model sẽ truy xuất dữ liệu từ cơ sở dữ liệu hoặc các nguồn khác và trả kết quả về Controller.
* View: Controller chuyển dữ liệu tới View, nơi dữ liệu được định dạng và gửi lại kết quả (HTML) cho người dùng.
* Response: Kết quả từ View được trả về cho người dùng dưới dạng trang web.
* *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.

**1.3.4. Cơ sở dữ liệu sử MySQL**

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.

***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.

***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.

***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: PHÂN TÍCH THIẾT KẾ HỆ THỐNG**

**2.1. Khảo sát hệ thống**

**2.2. Phân tích hệ thống**

**2.2.1. Mô tả nghiệp vụ**

**2.2.2. Yêu cầu chức năng, phi chức năng**

**2.3. Xác định tác nhân và ca sử dụng của hệ thống**

**2.3.1. Các tác nhân**

**2.3.2. Các ca sử dụng**

**2.4. Biểu đồ Usecase**

**2.5. Mô tả chi tiết Usecase**

**2.6. Biểu đồ trình tự, biểu đồ lớp chi tiết các Usecase**

**2.7. Cơ sở dữ liệu**

**2.7.1. Sơ đồ quan hệ dữ liệu**

**2.7.2. Chi tiết các bảng dữ liệu**

**2.8. Thiết kế giao diện**

**CHƯƠNG 3: CÀI ĐẶT VÀ TRIỂN KHAI**

**3.1. Giới thiệu về công cụ và môi trường cài đặt**

**3.2. Kết quả một số giao diện**

* **Trang màn hình giao diện của khách hàng.**
* **Trang màn hình giao diện của người quản trị**

**3.3. Một số giao diện khác**

**3.4. Kiểm thử**

**KẾT LUẬN**

**TÀI LIỆU THAM KHẢO**