|  |  |
| --- | --- |
|  |  |
| Lê Minh Quang CÔNG NGHỆ THÔNG TIN | **BỘ CÔNG THƯƠNG**  **TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**  **---------------------------------------** |
|  |
| **ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC**  **NGÀNH CÔNG NGHỆ THÔNG TIN** |
|  |
| **XÂY DỰNG WEBSITE MẠNG XÃ HỘI SỬ DỤNG**  **ASP.NET CORE WEB API VÀ React.JS** |
|  |
| **CBHD:** ThS. Vũ Duy Giang  **Sinh viên:** Lê Minh Quang  **Mã số sinh viên:** 2020605574 |
|  |
| **Hà Nội – Năm 2024** |

LỜI NÓI ĐẦU

*Em xin bày tỏ lòng biết ơn sâu sắc đến Trường Đại học Công nghiệp Hà Nội và Thầy giáo ThS. Vũ Duy Giang đã dành thời gian và kiến thức quý báu để hướng dẫn và hỗ trợ em trong quá trình thực hiện đồ án tốt nghiệp ngành Công nghệ Thông tin.*

*Em không thể không đề cập đến sự cống hiến và sự hỗ trợ chân thành của Thầy giáo Giang trong suốt quãng thời gian chúng tôi làm việc với ông. Sự kiên nhẫn, kiến thức sâu rộng và sự chỉ dẫn tận tình của Thầy giáo Giang đã giúp em vượt qua những thách thức trong quá trình nghiên cứu và thực hiện đồ án.*

*Em cũng muốn bày tỏ lòng biết ơn đến Ban giám hiệu và các giảng viên của Trường Đại học Công nghiệp Hà Nội đã cung cấp môi trường học tập và nghiên cứu chuyên nghiệp, giúp em phát triển kiến thức và kỹ năng trong suốt thời gian học tập tại trường.*

*Em xin chân thành cảm ơn!*

MỤC LỤC

[LỜI NÓI ĐẦU i](#_Toc167606925)

[DANH MỤC HÌNH ẢNH iii](#_Toc167606926)

[DANH MỤC BẢNG BIỂU v](#_Toc167606927)

[MỞ ĐẦU 1](#_Toc167606928)

[1. Lý do chọn đề tài 1](#_Toc167606929)

[2. Mục tiêu đề tài 1](#_Toc167606930)

[3. Đối tượng nghiên cứu 2](#_Toc167606931)

[4. Tính cấp thiết của đề tài 2](#_Toc167606932)

[Chương 1. CƠ SỞ LÝ THUYẾT 3](#_Toc167606933)

[1.1. Các khái niệm cơ bản 3](#_Toc167606934)

[Chương 2. CÔNG NGHỆ SỬ DỤNG ASP.NET VÀ REACTJS 8](#_Toc167606935)

[2.1. Công nghệ sử dụng 8](#_Toc167606936)

[2.2. Công cụ 16](#_Toc167606937)

[Chương 3. XÂY DỰNG WEBSITE MẠNG XÃ HỘI 18](#_Toc167606938)

[3.1. Nghiên cứu, khảo sát 18](#_Toc167606939)

[3.2. Phân tích nghiệp vụ 19](#_Toc167606940)

[3.3. Đặc tả yêu cầu 21](#_Toc167606941)

[3.4. Kết quả đạt được 45](#_Toc167606942)

[3.5. Kiểm thử 48](#_Toc167606943)

[KẾT LUẬN 54](#_Toc167606944)

[1. Những kết quả sau quá trình nghiên cứu 54](#_Toc167606945)

[2. Kiến nghị về những nghiên cứu tiếp theo. 55](#_Toc167606946)

[TÀI LIỆU THAM KHẢO 56](#_Toc167606947)

DANH MỤC HÌNH ẢNH

[Hình 2.1. ReactJS Framework 8](#_Toc167564334)

[Hình 2.2. Ảnh thư viện SimplePeer 10](#_Toc167564335)

[Hình 2.3. Mô hình Clean Architecture 14](#_Toc167564336)

[Hình 2.4. Mô hình chi tiết Clean Architecture 15](#_Toc167564337)

[Hình 3.1. Biểu đồ use case tổng quát 21](#_Toc167564338)

[Hình 3.2. Biểu đồ tuần tự của use case đăng nhập 23](#_Toc167564339)

[Hình 3.3. Biểu đồ trình tự use case đăng ký 25](#_Toc167564340)

[Hình 3.4. Biểu đồ trình tự use case quản lí thông tin cá nhân 27](#_Toc167564341)

[Hình 3.5. Biểu đồ trình tự quản lý bài viết 30](#_Toc167564342)

[Hình 3.6. Biểu đồ trình tự quản lí bình luận 33](#_Toc167564343)

[Hình 3.7. Biểu đồ trình tự quản lí tin nhắn 36](#_Toc167564344)

[Hình 3.8. Biểu đồ trình tự quản lí yêu cầu kết bạn 39](#_Toc167564345)

[Hình 3.9. Biểu đồ trình tự quản lí thông báo 42](#_Toc167564346)

[Hình 3.10. Biểu đồ Entity Relationship 43](#_Toc167564347)

[Hình 3.11. Hình các bảng trong CSDL 44](#_Toc167564348)

[Hình 3.12. Màn hình đăng nhập 45](#_Toc167564348)

[Hình 3.13. Màn hình đăng ký 45](#_Toc167564348)

[Hình 3.14. Màn hình trang chủ 46](#_Toc167564348)

[Hình 3.15. Màn hình trang cá nhân 46](#_Toc167564348)

[Hình 3.16. Màn hình hội thoại 47](#_Toc167564348)

[Hình 3.17. Màn hình quản lý thông báo 47](#_Toc167564348)

[Hình 3.18. Màn hình quản lý kết bạn 48](#_Toc167564348)

DANH MỤC BẢNG BIỂU

[Bảng 3.1. Bảng mô tả use case đăng nhập 22](#_Toc167564334)

[Bảng 3.2. Bảng mô tả use case đăng ký 24](#_Toc167564335)

[Bảng 3.3. Bảng mô tả use case quản lý thông tin cá nhân 26](#_Toc167564336)

[Bảng 3.4. Bảng mô tả use case bài viết 28](#_Toc167564337)

[Bảng 3.5. Bảng mô tả use case quản lý bình luận 31](#_Toc167564338)

[Bảng 3.6. Bảng mô tả use case quản lý tin nhắn 34](#_Toc167564339)

[Bảng 3.7. Bảng mô tả use case quản lý yêu cầu kết bạn 37](#_Toc167564340)

[Bảng 3.8. Bảng mô tả use case quản lý thông báo 40](#_Toc167564341)

[Hình 3.9. Thông tin kiểm thử 48](#_Toc167564342)

[Hình 3.10. Nội dung kiểm thử 48](#_Toc167564343)

[Hình 3.11. Kiểm thử chức năng hệ thống 52](#_Toc167564344)

[Hình 3.12. Mô tả kiểm thử giao diện 52](#_Toc167564344)

**DANH MỤC CÁC THUẬT NGỮ, KÝ HIỆU VÀ CÁC CHỮ VIẾT TẮT**

|  |  |  |
| --- | --- | --- |
| STT | Ký hiệu | Ý nghĩa |
| 1 | MXH | Mạng xã hội |
|  |  |  |
|  |  |  |

MỞ ĐẦU

1. Lý do chọn đề tài

Trong thời buổi công nghệ thông tin và truyền thông ngày càng cải tiến và phát triển một cách thần tốc. Mạng xã hội đã trở thành một thứ không còn xa lạ gì với mọi người. Mạng xã hội (MXH) là những nền tảng trực tuyến cho phép người dùng tương tác, chia sẻ thông tin và kết nối với nhau. Ngày nay, MXH đã trở thành một phần không thể thiếu trong cuộc sống của nhiều người, mang đến vô số lợi ích và tác động to lớn đến xã hội. Lý do chính đằng sau sự ra đời của MXH xuất phát từ nhu cầu kết nối và chia sẻ của con người. Từ xa xưa, con người đã luôn tìm kiếm những cách thức để liên lạc, trao đổi thông tin và duy trì mối quan hệ với nhau. MXH ra đời như một giải pháp hoàn hảo cho nhu cầu này, giúp xóa nhòa rào cản địa lý, thời gian và kết nối mọi người trên toàn thế giới. Bên cạnh đó, MXH cũng được xây dựng với mục đích giải trí, cung cấp thông tin và hỗ trợ kinh doanh. Các nền tảng MXH đa dạng mang đến cho người dùng vô số nội dung giải trí hấp dẫn, cập nhật tin tức nhanh chóng và tạo cơ hội quảng bá thương hiệu, sản phẩm hiệu quả. Chính vì thế, em đã quyết định xây dựng một trang web về mạng xã hội nhắm vào nhóm đối tượng yêu thích phim ảnh. Website sẽ cung cấp những tính năng của một trang web mạng xã hội và cung cấp những kho tàng phim ảnh đồ sộ dể người dùng thỏa thích lựa chọn..Tuy nhiên, sản phẩm này của em về chuyên môn vẫn chưa được hoàn thiện và có thể thiếu sót một số thông tin nhất định. Nhưng em sẽ cố gắng hoàn thiện ứng dụng một cách chỉnh chu và chuyên nghiệp nhất có thể.

1. Mục tiêu đề tài

Đề tài: **Xây dựng website mạng xã hội** đáp ứng được những mục tiêu:

* Xây dựng giao diện người dùng với một số chức năng cơ bản : cho phép người dùng quản lý thông tin cá nhân, tạo bài viết, bình luận, nhắn tin tương tác với những người dùng khác.
* Website còn cung cấp một kho tàng phim ảnh để cho người dùng yêu điện ảnh có thể thỏa thích lựa chọn.
* Phân tích thiết kế Database.

1. Đối tượng nghiên cứu

* Website mạng xã hội
* Framework ReactJS
* ASP.Net Core
* CSDL : MariaDB

1. Tính cấp thiết của đề tài

Với sự bùng nổ mạnh mẽ của công nghệ thông tin và sự phát triển của mạng Internet. Nhu cầu trao đổi thông tin cũng như giải trí đã trở nên cần thiết hơn bao giờ hết trong xã hội hiện.

Để đáp ứng được những nhu cầu này. Việc xây dựng một website mạng xã hội là cần thiết hơn bao giờ hết.

Để xây đáp ứng yêu cầu của học phần đề ra, cũng như thấy được sự quan trọng của vấn đề này và niềm đam mê thiết kế ra các trang web em đã chọn đề tài “Xây dựng website mạng xã hội.”.

## CƠ SỞ LÝ THUYẾT

### Các khái niệm cơ bản

##### Website là gì ?

Web là tên thường gọi của World Wide Web (mạng toàn cầu), một tập hợp con của Internet bao gồm các trang có thể được truy cập bằng trình duyệt Web.

Các trang web được định dạng bằng ngôn ngữ gọi là Ngôn ngữ đánh dấu siêu văn bản (Hypertext Markup Language - HTML). Ngôn ngữ này cho phép người dùng nhấp qua các trang trên Web thông qua các liên kết. Web sử dụng giao thức HTTP để truyền dữ liệu và chia sẻ thông tin. Các trình duyệt như Internet Explorer, Google Chrome hoặc Mozilla Firefox hoạt động như một công cụ để người dùng có thể truy cập các tài liệu Web hoặc các trang Web được kết nối thông qua các liên kết.

Web chỉ là một trong những cách chia sẻ thông tin qua Internet bên cạnh những thứ khác bao gồm email, nhắn tin tức thời và Giao thức truyền tệp (FTP).

Hiểu một cách ngắn gọn thì Web là mạng.

Còn Site là địa điểm. Ví dụ worksite có nghĩa là nơi làm việc, chỉ một địa điểm, địa chỉ cụ thể.

Như vậy Website = Web + Site, tức một địa chỉ cụ thể trong mạng toàn cầu hay còn được gọi là trang mạng.

Điều đó có nghĩa là website phải đảm bảo được các yếu tố: Nằm trong mạng toàn cầu, có thể truy cập bằng các trình duyệt web, sử dụng giao thức HTTP để truyền dữ liệu và chia sẻ thông tin và quan trọng là phải có một địa chỉ cụ thể.

Website thường chứa các nội dung văn bản, hình ảnh, video và rất nhiều định dạng nội dung khác, được lưu trữ trên máy chủ.

Cấu tạo và hoạt động của website :

* Website thường chứa nhiều webpage hay còn thường được gọi là trang con. Tất cả được lưu trữ dưới định dạng html hoặc xhtml (Extensible HyperText Markup Language - mở rộng của html). Chúng sẽ được lưu trên các máy chủ (web server).
* Khi người dùng muốn truy cập các thông tin từ website cần sử dụng các trình duyệt web để truy cập vào địa chỉ của website, đọc các file lưu trữ dưới định dạng html hoặc xhtml và hiển thị dưới dạng trực quan để dễ dàng tiếp nhận nội dung, thao tác.

Một website muốn hoạt động cần có các thành phần:

* Source Code (mã nguồn): Để các nội dung có thể hiển thị, thao tác, tương tác, tự động tối ưu trên từng loạt thiết bị…
* Web hosting (Lưu trữ web): Nếu website là một ngôi nhà thì hosting chính là miếng đất để xây ngôi nhà đó. Nó lưu trữ Source Code, hình ảnh, video, nội dung… của website.
* Tên miền (domain): Khi bạn đã có miếng đất và ngôi nhà, bạn cần có địa chỉ để những người khác có thể tìm đến và truy cập. Vì vậy bạn cần có một tên miền (domain). Mỗi tên miền là duy nhất trên toàn thế giới để đảo bảo mọi người đều có thể đến chính xác nhà của bạn.

##### Website mạng xã hội:

**Website mạng xã hội** (MXH) là những nền tảng trực tuyến cho phép người dùng **tương tác, chia sẻ thông tin và kết nối với nhau**. Chúng khác với các website thông thường ở những điểm sau:

* **Tính tương tác:** Website MXH chú trọng vào **tương tác trực tiếp** giữa người dùng. Người dùng có thể đăng tải nội dung, bình luận, chia sẻ, thích, theo dõi và nhắn tin cho nhau.
* **Tính cộng đồng:** Website MXH tạo ra **môi trường cộng đồng** nơi người dùng có thể kết nối với những người cùng sở thích, đam mê hoặc mục tiêu.
* **Nội dung đa dạng:** Website MXH cung cấp **nhiều loại nội dung** khác nhau như văn bản, hình ảnh, video, âm thanh, livestream...
* **Tính kết nối:** Website MXH giúp người dùng **kết nối** với bạn bè, gia đình, đồng nghiệp và những người mới từ khắp nơi trên thế giới.
* **Tính cập nhật:** Website MXH thường xuyên **cập nhật** những nội dung mới nhất, giúp người dùng nắm bắt tin tức và xu hướng một cách nhanh chóng.

Một số ví dụ về website MXH phổ biến:

* Facebook
* Instagram
* Twitter
* Zalo
* YouTube
* TikTok

Lợi ích của MXH:

* Kết nối con người: MXH giúp kết nối mọi người từ khắp nơi trên thế giới, thu hẹp khoảng cách địa lý và tạo điều kiện cho giao lưu, chia sẻ thông tin.
* Truy cập thông tin: MXH là nguồn cung cấp thông tin nhanh chóng và phong phú, giúp người dùng cập nhật tin tức, sự kiện và kiến thức mới nhất.
* Giải trí: MXH mang đến nhiều hình thức giải trí đa dạng như video, âm nhạc, game, giúp người dùng thư giãn và giảm stress.
* Kinh doanh: MXH là kênh marketing hiệu quả để quảng bá thương hiệu, sản phẩm và dịch vụ, thu hút khách hàng và thúc đẩy doanh số bán hàng.
* Hỗ trợ học tập: MXH giúp chia sẻ tài liệu, kết nối với giáo viên và bạn học, tạo môi trường học tập trực tuyến hiệu quả.

Thách thức của MXH:

* Tin giả: MXH là môi trường tiềm ẩn nhiều tin giả, thông tin sai lệch và gây hoang mang cho người dùng.
* Nội dung độc hại: MXH có thể chứa nội dung bạo lực, khiêu dâm, ảnh hưởng tiêu cực đến sức khỏe tinh thần và đạo đức, đặc biệt là trẻ em.
* Nghiện mạng xã hội: Việc sử dụng MXH quá mức có thể dẫn đến nghiện, gây ảnh hưởng đến sức khỏe, học tập và công việc.
* Sâu rộng và khó kiểm soát: MXH là môi trường rộng lớn và khó kiểm soát, tiềm ẩn nguy cơ lừa đảo, quấy rối và các hành vi vi phạm pháp luật.

**Kết luận chương**

Sau khi tìm hiểu về các khái niệm website và website mạng xã hội, ta hiểu rõ hơn về cách thức hoạt động và tầm quan trọng của chúng trong xã hội hiện nay. Website, với vai trò là nền tảng cung cấp thông tin và dịch vụ, đã trở thành công cụ không thể thiếu trong đời sống hiện nay. Đặc biệt, website mạng xã hội, với các chức năng như kết nội mọi người, giải trí,.. mang đến vô số lợi ích và tác động to lớn đến mọi khía cạnh của xã hội. Tuy nhiên, bên cạnh những lợi ích, MXH cũng tiềm ẩn nhiều nguy cơ và thách thức cần được quan tâm giải quyết. Tóm lại, Việc sử dụng MXH một cách thông minh và có trách nhiệm sẽ giúp chúng ta tận dụng tối đa lợi ích và hạn chế những tác hại tiềm ẩn sẽ đem lại cho xã hội vô vàn lợi ích.

## CÔNG NGHỆ SỬ DỤNG ASP.NET VÀ REACTJS

### Công nghệ sử dụng

#### Xây dựng giao diện người dùng bằng ReactJS

##### FrameWork ReactJS

 ReactJS là một framework mã nguồn mở của JavaScript được sử dụng để phát triển các giao diện web tương tác. Nó là một trong những framework nổi tiếng được sử dụng để đơn giản hóa việc phát triển web. ReactJS tập trung vào view layer. Nó có thể dễ dàng tích hợp vào các dự án lớn để phát triển front-end mà không gặp bất kỳ sự cố nào.

A blue and white logo

Description automatically generated

Hình 2.1. ReactJS Framework

Trong ReactJS, có một số tính năng quan trọng bao gồm:

* **Mô hình thành phần:** ReactJS sử dụng mô hình thành phần để chia nhỏ giao diện người dùng thành các phần nhỏ, độc lập và có thể tái sử dụng. Điều này giúp cho việc tổ chức mã dễ dàng hơn, đồng thời giúp cải thiện khả năng bảo trì và tái sử dụng mã.
* **DOM ảo:** ReactJS sử dụng DOM ảo để tối ưu hóa hiệu suất của ứng dụng. DOM ảo là một mô hình biểu diễn nội bộ của DOM thực. Khi có thay đổi được thực hiện đối với giao diện người dùng, ReactJS sẽ so sánh DOM ảo với DOM thực và chỉ cập nhật các phần DOM đã thay đổi. Điều này giúp giảm thiểu số lượng thao tác DOM cần thiết, dẫn đến hiệu suất ứng dụng nhanh hơn.
* **JSX:** ReactJS sử dụng JSX, một cú pháp mở rộng JavaScript, để mô tả các thành phần giao diện người dùng. JSX cho phép bạn viết mã HTML trực tiếp trong mã JavaScript của mình, giúp cho việc viết mã giao diện người dùng dễ dàng và trực quan hơn.
* **Trình quản lý trạng thái:** ReactJS cung cấp một hệ thống quản lý trạng thái tích hợp để giúp bạn quản lý trạng thái của ứng dụng của mình. Hệ thống quản lý trạng thái này giúp đảm bảo rằng dữ liệu ứng dụng của bạn luôn nhất quán và cập nhật.
* **Khả năng mở rộng:** ReactJS là một thư viện có thể mở rộng cao, có nghĩa là bạn có thể sử dụng nó để xây dựng các ứng dụng web từ đơn giản đến phức tạp. Có rất nhiều thư viện và công cụ của bên thứ ba có sẵn cho ReactJS, giúp bạn dễ dàng thêm tính năng vào ứng dụng của mình.
* **Cộng đồng lớn:** ReactJS có một cộng đồng lớn và phát triển nhanh chóng gồm các nhà phát triển. Điều này có nghĩa là bạn có thể dễ dàng tìm thấy sự trợ giúp và hỗ trợ khi bạn cần.

##### Simple Peer:

**Simple Peer** là một thư viện JavaScript mã nguồn mở giúp đơn giản hóa quá trình xây dựng các ứng dụng WebRTC ngang hàng (peer-to-peer). Nó cung cấp một API cấp cao giúp bạn dễ dàng kết nối hai trình duyệt web với nhau và truyền tải dữ liệu âm thanh, video và dữ liệu giữa chúng.

A blue and white logo

Description automatically generated

Hình 2.2. Ảnh thư viện SimplePeer

Ưu điểm của Simple Peer:

* Dễ sử dụng: Simple Peer cung cấp một API đơn giản và dễ hiểu, giúp bạn dễ dàng bắt đầu sử dụng WebRTC.
* Nhẹ: Simple Peer là một thư viện nhỏ và nhẹ, giúp giảm thiểu kích thước gói tin của bạn.
* Tương thích: Simple Peer tương thích với tất cả các trình duyệt web hiện đại.
* Mã nguồn mở: Simple Peer là mã nguồn mở, vì vậy bạn có thể kiểm tra mã và đóng góp cho dự án.

Simple Peer là một công cụ mạnh mẽ để xây dựng các ứng dụng WebRTC ngang hàng. Nó dễ sử dụng, nhẹ và tương thích với tất cả các trình duyệt web hiện đại.

Ngoài ra, Simple Peer còn có một số tính năng nâng cao khác, bao gồm:

* Hỗ trợ ICE: ICE (Interactive Connectivity Establishment) là một giao thức giúp thiết lập kết nối giữa hai trình duyệt web qua mạng NAT.
* Hỗ trợ STUN/TURN: STUN và TURN là các máy chủ giúp thiết lập kết nối giữa hai trình duyệt web khi chúng không nằm trên mạng cục bộ.
* Hỗ trợ mã hóa: Simple Peer hỗ trợ mã hóa SRTP và DTLS để bảo vệ dữ liệu của bạn khỏi bị nghe lén.

#### Xây dựng API bằng ASP.NET core

##### Framework ASP.NET core

ASP.NET là một nền tảng web được sử dụng để tạo các trang web, ứng dụng và dịch vụ web. Đây là sản phẩm tích hợp của các ngôn ngữ HTML, CSS và JavaScript. ASP.NET được ra đời vào năm 2002 và đến nay đã được nâng cấp thành nhiều phiên bản khác nhau.

ASP.NET hoạt động trên HTTP, sử dụng các lệnh và chính sách của HTTP để trình duyệt có thể trở thành giao tiếp song phương của máy chủ.

Lợi ích của ASP.NET Core:

* Mã nguồn mở: ASP.NET Core là mã nguồn mở, miễn phí cho bất kỳ ai sử dụng. Điều này mang lại cho các nhà phát triển sự linh hoạt và khả năng kiểm soát cao hơn đối với các ứng dụng của họ.
* Đa nền tảng: ASP.NET Core có thể chạy trên Windows, macOS và Linux. Điều này cho phép các nhà phát triển xây dựng các ứng dụng có thể triển khai trên nhiều môi trường khác nhau.
* Hiệu suất cao: ASP.NET Core được thiết kế để mang lại hiệu suất cao và có thể xử lý tải trọng lớn.
* Dễ sử dụng: ASP.NET Core được thiết kế để dễ sử dụng và học tập. Nó cung cấp nhiều tính năng và công cụ để giúp các nhà phát triển xây dựng các ứng dụng web nhanh chóng và dễ dàng.
* Có thể mở rộng: ASP.NET Core có thể mở rộng và có thể được sử dụng để xây dựng các ứng dụng web từ đơn giản đến phức tạp.

Các tính năng chính của ASP.NET Core:

* Model-View-Controller (MVC): ASP.NET Core sử dụng mô hình MVC để phân tách các mối quan tâm của ứng dụng web thành ba lớp: mô hình, giao diện và bộ điều khiển.
* Razor Pages: Razor Pages là một cách tiếp cận mới để xây dựng các trang web ASP.NET Core. Razor Pages đơn giản hơn MVC và có thể được sử dụng để xây dựng các ứng dụng web nhanh chóng và dễ dàng.
* Web API: ASP.NET Core Web API là một khung API RESTful cho phép các nhà phát triển xây dựng các dịch vụ web.
* SignalR: SignalR là một thư viện cho phép các nhà phát triển xây dựng các ứng dụng web thời gian thực.
* Dependency Injection: ASP.NET Core hỗ trợ Dependency Injection, giúp các nhà phát triển viết mã sạch hơn và dễ bảo trì hơn.

Sử dụng ASP.NET Core:

ASP.NET Core có thể được sử dụng để xây dựng nhiều loại ứng dụng web khác nhau, bao gồm:

* Ứng dụng web: ASP.NET Core có thể được sử dụng để xây dựng các ứng dụng web từ đơn giản đến phức tạp.
* Dịch vụ web: ASP.NET Core Web API có thể được sử dụng để xây dựng các dịch vụ web RESTful.
* Ứng dụng web thời gian thực: SignalR có thể được sử dụng để xây dựng các ứng dụng web thời gian thực.
* Ứng dụng di động: ASP.NET Core có thể được sử dụng để xây dựng các ứng dụng di động back-end.

##### Xây dựng theo chuẩn Restful API

Restful **API** là một tiêu chuẩn dùng trong việc thiết kế các API cho các ứng dụng web để quản lý các resource. Restful là một trong những kiểu thiết kế API được sử dụng phổ biến ngày nay để cho các ứng dụng (web, mobile…) khác nhau giao tiếp với nhau.

Chức năng quan trọng nhất của **REST** là quy định cách sử dụng các HTTP method (như GET, POST, PUT, DELETE…) và cách định dạng các URL cho ứng dụng web để quản các resource. **RESTful** không quy định logic code ứng dụng và không giới hạn bởi ngôn ngữ lập trình ứng dụng, bất kỳ ngôn ngữ hoặc framework nào cũng có thể sử dụng để thiết kế một **RESTful API.**

REST hoạt động chủ yếu dựa vào giao thức HTTP. Các hoạt động cơ bản nêu trên sẽ sử dụng những phương thức HTTP riêng.

* GET (SELECT): Trả về một Resource hoặc một danh sách Resource.
* POST (CREATE): Tạo mới một Resource.
* PUT (UPDATE): Cập nhật thông tin cho Resource.
* DELETE (DELETE): Xoá một Resource.

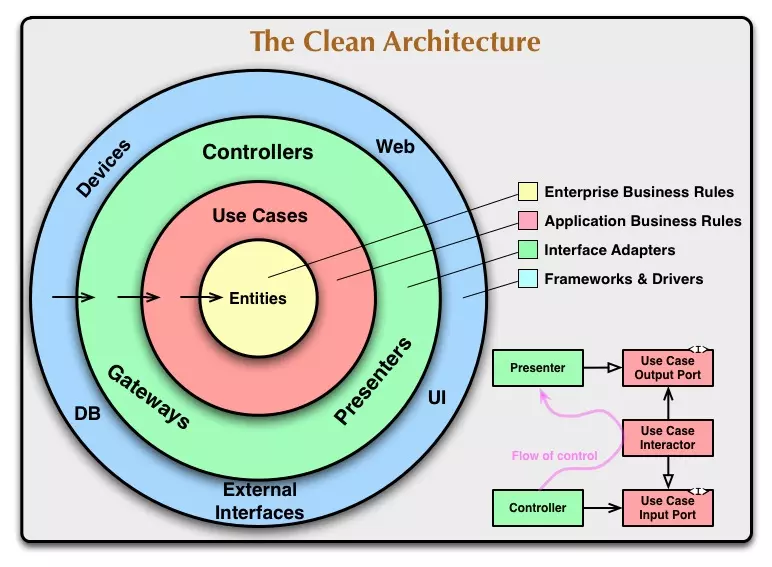
##### Mô hình xây dựng : Mô hình đa tầng

Kiến Trúc đa tầng (Clean Architecture) là một mô hình thiết kế phần mềm được giới thiệu bởi Robert C. Martin (còn được gọi là Uncle Bob). Mục tiêu của kiến trúc này là tạo ra các hệ thống dễ bảo trì, dễ mở rộng và dễ kiểm tra, đồng thời tách biệt rõ ràng giữa các mối quan tâm khác nhau của ứng dụng.

Nguyên lý cơ bản :

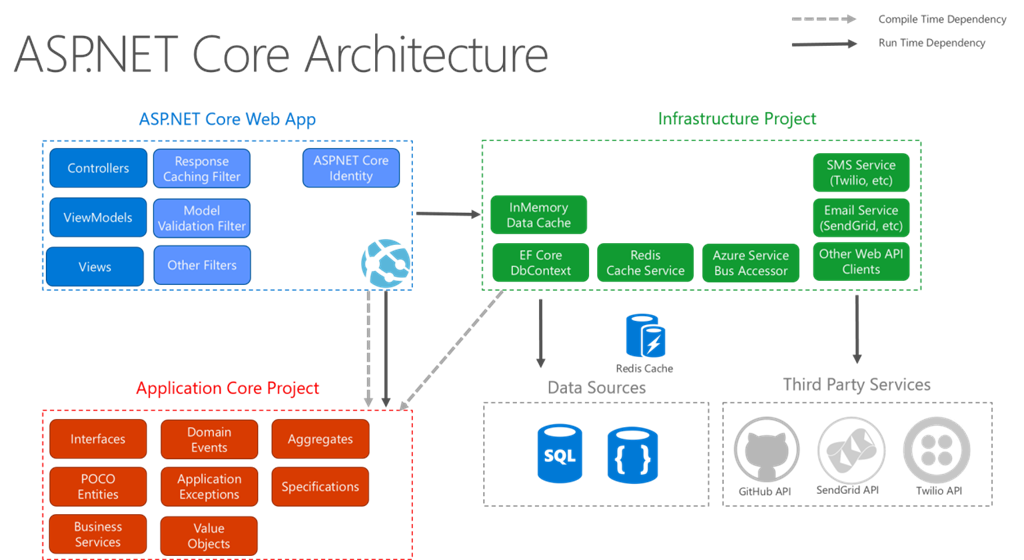
Kiến Trúc đa tầng dựa trên một số nguyên lý cơ bản, trong đó nổi bật nhất là Nguyên Tắc Đảo Ngược Phụ Thuộc (Dependency Inversion Principle - DIP) và Thiết Kế Hướng Miền (Domain-Driven Design - DDD). Các nguyên lý này nhấn mạnh rằng:

* Các mô-đun cấp cao không nên phụ thuộc vào các mô-đun cấp thấp. Cả hai nên phụ thuộc vào các giao diện trừu tượng.
* Các chi tiết cụ thể (cơ sở dữ liệu, giao diện người dùng, thư viện) nên phụ thuộc vào các trừu tượng chứ không phải ngược lại



Hình 2.3. Mô hình Clean Architecture

Các phụ thuộc trong Clean Architecture luôn hướng về phía trong. Điều này có nghĩa là các lớp ngoài cùng phụ thuộc vào các lớp bên trong chứ không phải ngược lại. Các giao diện (interfaces) được định nghĩa trong các lớp bên trong và được triển khai trong các lớp bên ngoài.



Hình 2.4. Mô hình chi tiết Clean Architecture

##### Giới thiệu về SignalR

SignalR là thư viện mã nguồn mở của Microsoft giúp phát triển các ứng dụng web thời gian thực. Nó cho phép kết nối hai chiều giữa máy khách và máy chủ, giúp bạn dễ dàng xây dựng các tính năng như trò chuyện trực tuyến, bảng tin cập nhật theo thời gian thực, ứng dụng cộng tác và nhiều hơn nữa.

Ưu điểm của SignalR:

* Kết nối thời gian thực: SignalR sử dụng kết nối WebSocket để cung cấp kết nối thời gian thực giữa máy khách và máy chủ. Điều này cho phép bạn gửi và nhận dữ liệu ngay lập tức, mà không cần phải yêu cầu HTTP liên tục.
* Dễ sử dụng: SignalR cung cấp API đơn giản và dễ sử dụng, giúp bạn dễ dàng xây dựng các ứng dụng web thời gian thực.
* Hỗ trợ đa nền tảng: SignalR hỗ trợ nhiều nền tảng máy khách, bao gồm JavaScript, .NET, Java và Python.
* Có thể mở rộng: SignalR có thể được mở rộng để đáp ứng nhu cầu của các ứng dụng lớn và phức tạp.

Ví dụ về ứng dụng của SignalR:

* Trò chuyện trực tuyến: SignalR có thể được sử dụng để xây dựng các ứng dụng trò chuyện trực tuyến, cho phép người dùng trò chuyện với nhau theo thời gian thực.
* Bảng tin cập nhật theo thời gian thực: SignalR có thể được sử dụng để xây dựng các bảng tin cập nhật theo thời gian thực, hiển thị các thay đổi gần đây cho tất cả người dùng ngay lập tức.
* Ứng dụng cộng tác: SignalR có thể được sử dụng để xây dựng các ứng dụng cộng tác, cho phép nhiều người dùng làm việc trên cùng một tài liệu cùng lúc.
* Trò chơi trực tuyến: SignalR có thể được sử dụng để xây dựng các trò chơi trực tuyến nhiều người chơi, cho phép người chơi tương tác với nhau theo thời gian thực.

### Công cụ

* Visual Studio Code : phục vụ code giao diện người dùng
* Visual Studio: phục vụ code API
* MariaDB : cơ sở dữ liệu dùng để lưu trữ dữ liệu.
* Postman: sử dụng để thử nghiệm API.
* Chrome DevTools: gỡ lỗi khi phát triển web.
* Diagram : thiết kế cơ sở dữ liệu.
* Rational Rose: thiết kế hệ thống.

**Kết luận chương**

Là kết quả của quá trình tìm hiểu, phân tích và tổng hợp các nguồn tài liệu và nghiên cứu liên quan đến đề tài "Xây dựng website mạng xã hội sử dụng React.js và ASP.NET". Trong chương này, chúng tôi đã trình bày các khái niệm và nguyên lý cơ bản về hai công nghệ chính được sử dụng trong dự án của chúng tôi.

Đầu tiên, chúng tôi đã giới thiệu về ReactJS - một framework JavaScript phổ biến được sử dụng cho việc phát triển giao diện người dùng tương tác. Chúng tôi đã trình bày về các đặc điểm và ưu điểm của ReactJS như tính linh hoạt, dễ dàng học và sử dụng, cũng như cách nó hoạt động trong việc tạo ra các component, quản lý trạng thái ứng dụng và xử lý sự kiện.

Tiếp theo, chúng tôi đã tập trung vào ASP.NET - một framework phổ biến cho việc phát triển ứng dụng web và dịch vụ web. Chúng tôi đã giới thiệu về kiến trúc và các thành phần chính của ASP.NET, cũng như cách nó được sử dụng trong việc xây dựng backend cho trang web của chúng tôi. Chúng tôi đã tóm tắt về việc xử lý yêu cầu HTTP, truy cập cơ sở dữ liệu, xác thực và quản lý phiên trong ASP.NET.

Từ việc hiểu biết sâu rộng về ReactJS và ASP.NET qua các nội dung được trình bày trong chương cơ sở lý thuyết, chúng tôi sẽ tiếp tục triển khai và phát triển ứng dụng website mạng xã hội của mình, hy vọng rằng sẽ đem lại trải nghiệm tuyệt vời cho người dùng cuối.

## XÂY DỰNG WEBSITE MẠNG XÃ HỘI

### Nghiên cứu, khảo sát

#### Xác định mục tiêu nghiên cứu:

* Hiểu rõ về hành vi, sở thích của người dùng khi sử dụng mạng xã hội.
* Xác định các yếu tố quyết định khi khách hàng quyết định sử dụng mạng xã hội
* Tìm hiểu về các dịch vụ và tính năng mà khách hàng mong đợi từ một trang web mạng xã hội.

#### Thu thập dữ liệu cơ bản:

* Thu thập thông tin về các trang web mạng xã hội, trang phim đang hoạt động trên thị trường.
* Phân tích xu hướng, hành vi của người dùng trên các trang mạng xã hội
* Nghiên cứu về người tiêu dùng:
* Tiến hành cuộc khảo sát hoặc phỏng vấn với một mẫu ngẫu nhiên các người tiêu dùng để hiểu rõ hơn về hành vi của người dùng khi sử dụng mạng xã hội.
* Phân tích dữ liệu từ các trang diễn đàn để tìm hiểu về những ý kiến và phản hồi của người dùng về các trang web mạng xã hội hiện tại.

#### Đánh giá đối thủ:

* Phân tích các trang web bán mạng xã hội hàng đầu trên thị trường để hiểu về các dịch vụ và tính năng mà họ cung cấp.
* Đánh giá điểm mạnh và điểm yếu của các đối thủ trong lĩnh vực mạng xã hội.

#### Phân tích xu hướng và tiềm năng thị trường:

* Tìm hiểu về các xu hướng mới trong lĩnh vực mạng xã hội, như Sự trỗi dậy của các nền tảng video, hoặc sự lan rộng của tin nhắn.
* Xem xét các chính sách pháp lý và các yếu tố văn hóa xã hội có thể ảnh hưởng đến thị trường mạng xã hội.

#### Tổng hợp và phân tích dữ liệu:

* Tổng hợp và phân tích dữ liệu thu thập được từ các nguồn khác nhau để đưa ra các kết luận và hướng đi cho chiến lược kinh doanh của bản thân.

#### Xây dựng chiến lược:

* Dựa trên những phân tích và kết luận từ nghiên cứu và khảo sát, xây dựng một chiến lược tiếp thị và kinh doanh hiệu quả để thu hút khách hàng và tạo ra lợi ích cho doanh nghiệp của bản thân.

### Phân tích nghiệp vụ

#### Xác định yêu cầu hệ thống

##### Quản lý thông tin cá nhân:

* Hiển thị thông tin chi tiết về người dùng bao gồm các thông tin cơ bản như: họ và tên, giới tính, ngày tháng năm sinh, avatar, email, địa chỉ, …
* Cho phép người dùng cập nhật thông tin cá nhân.

##### Quản lý bài viết:

* Hệ thống lưu trữ các bài viết của người dùng gồm: nội dung, ảnh, video, lượt thích.
* Cho phép người dùng thêm, sửa, xóa bài viết của bản thân.

##### Quản lý bình luận:

* Hệ thống lưu trữ các bài viết của người dùng gồm: nội dung, ảnh, video, lượt thích.
* Cho phép người dùng thêm, sửa, xóa bình luận của bản thân.

##### Quản lý tin nhắn:

* Hệ thống lưu trữ các tin nhắn của người dùng gồm: nội dung, ảnh, video.
* Cho phép người dùng thêm, sửa, xóa tin nhắn của bản thân.

##### Quản lý yêu cầu kết bạn:

* Hệ thống lưu trữ các yêu cầu kết bạn của người dùng gồm: người gửi, người nhận, trạng thái yêu cầu.
* Cho phép người dùng thêm, sửa, xóa yêu cầu kết bạn.

##### Quản lý thông báo:

* Hệ thống lưu trữ các thông báo của người dùng gồm: người tạo thông báo, người nhận thông báo, nội dung, đường dẫn
* Cho phép người dùng thêm, sửa, xóa thông báo.

##### Bảo mật và quản lý dữ liệu:

* Bảo vệ thông tin cá nhân của người dùng và giao dịch thanh toán bằng cách sử dụng mã hóa và các biện pháp bảo mật khác.
* Sao lưu và phục hồi dữ liệu định kỳ để đảm bảo tính toàn vẹn và sẵn sàng của hệ thống.

##### Giao diện thân thiện với người dùng:

* Thiết kế giao diện người dùng đẹp mắt, dễ sử dụng và thân thiện với người dùng.
* Tối ưu hóa trải nghiệm người dùng trên cả máy tính và thiết bị di động.
* Tích hợp dịch vụ bên thứ ba:
* Tích hợp các API cung cấp dữ liệu thời tiết, cung cấp dữ liệu phim ảnh.

### Đặc tả yêu cầu

#### Biểu đồ Use Case tổng quát



Hình 3.1. Biểu đồ use case tổng quát

#### Mô tả chi tiết Use Case

##### Use case: Đăng nhập

* Mô tả use case :

Bảng 3.1. Mô tả use case đăng nhập

|  |  |
| --- | --- |
| **Tên Use Case** | Đăng nhập |
| **Mô tả** | Use case này cho phép người dùng đăng nhập vào trang web |
| **Người thực hiện** | Người dùng |
| **Tiền điều kiện** | Đã đăng ký |
| **Luồng cơ bản** | Use case này bắt đầu khi người dùng kích vào “Login”. Hệ thống sẽ hiển thị lên màn hình giao diện đăng nhập bao gồm: Tên đăng nhập, mật khẩu  Khi người dùng nhập xong thông tin. Rồi nhấn vào nút “Đăng nhập”, hệ thống sẽ kiểm tra thông tin từ bảng USER và điều hương người dùng vào trang chủ. Use case kết thúc. |
| **Luồng rẽ nhánh** | Tại bước 2 trong luồng cơ bản nếu nhập sai tên đăng nhập hoặc mật khẩu, hệ thống sẽ báo “Tài khoản hoặc mật khẩu không đúng” và use case kết thúc.  Tại bất kỳ bước nào trong luồng cơ bản nếu không được kết nối với cơ sở dư liệu thì hệ thống sẽ hiển thị một thông báo lỗi. Use case kết thúc. |
| **Hậu điều kiện** | Không. |

* Biểu đồ trình tự :



Hình 3.2. Biểu đồ tuần tự của use case đăng nhập

##### Use case: Đăng ký

* Mô tả use case :

Bảng 3.2. Mô tả use case đăng ký

|  |  |
| --- | --- |
| **Tên Use Case** | Đăng ký |
| **Mô tả** | Use case này cho phép khách hàng đăng ký tài khoản |
| **Người thực hiện** | Người dùng |
| **Tiền điều kiện** | Không |
| **Luồng cơ bản** | Use case này bắt đầu khi khách hàng kích vào “Đăng ký”". Hệ thống sẽ hiển thị giao diện đăng ký.  Hệ thống sẽ hiển thị giao diện đăng kí bao gồm: họ và tên, giới tính, ngày sinh , email, mật khẩu  iii. Người dùng điền đầy đủ thông tin theo yêu cầu, rồi kích nút “Đăng kí”, hệ thống sẽ tạo một mã tài khoản mới, lưu thông tin vào bảng USER. Use case kết thúc. |
| **Luồng rẽ nhánh** | Tại bước 3 trong luồng cơ bản nếu khách hàng nhập thiếu thông tin, hệ thống sẽ yêu cầu khách hàng nhập đầy đủ thông tin. Nếu khách hàng nhập ko đúng định dạng email hoặc mật khẩu dưới 6 kí tự, hệ thống sẽ hiển thị lên 1 bảng thông báo nhỏ và yêu cầu nhập lại. Use case kết thúc.  Tại bất kỳ bước nào trong luồng cơ bản nếu không được kết nối với cơ sở dư liệu thì hệ thống sẽ hiển thị một thông báo lỗi. Use case kết thúc. |
| **Hậu điều kiện** | Không |

* Biểu đồ trình tự :



Hình 3.3. Biểu đồ trình tự use case đăng ký

##### Use case : Quản lý thông tin cá nhân

##### Mô tả use case :

Bảng 3.3. Mô tả use case quản lý thông tin cá nhân

|  |  |
| --- | --- |
| **Tên Use Case** | Quản lý thông tin cá nhân |
| **Mô tả** | Use case này cho phép người dùng quản lý thông tin cá nhân theo tài khoản |
| **Người thực hiện** | Người dùng |
| **Tiền điều kiện** | Đã đăng nhập |
| **Luồng cơ bản** | Use case bắt đầu khi người dùng kích vào thông tin của bản thân. Hệ thống sẽ lấy thông tin người dùng trong bảng User hiển thị lên màn hình.   * Sửa thông tin cá nhân: * Người dùng click vào nút sửa thông tin cá nhân trên màn hình. Hệ thống sẽ hiển thị ra form sửa thông tin cá nhân. * Người dùng nhập thông tin và click vào nút sửa. Hệ thống sẽ sửa lại thông tin cá nhân của người dùng   . Use case kết thúc |
| **Luồng rẽ nhánh** | Tại bất kỳ bước nào trong luồng cơ bản nếu không được kết nối với cơ sở dư liệu thì hệ thống sẽ hiển thị một thông báo lỗi. Use case kết thúc. |
| **Hậu điều kiện** | Bảng User được cập nhật |

##### Biểu đồ trình tự:



Hình 3.4. Biểu đồ trình tự use case quản lí thông tin cá nhân

##### Use case: Quản lý bài viết

* Mô tả use case

Bảng 3.4. Mô tả use case quản lý bài viết

|  |  |
| --- | --- |
| **Tên Use Case** | Quản lý tài khoản |
| **Mô tả** | Use case cho phép người dùng xem, thêm, sửa và xóa bài viết. |
| **Người thực hiện** | Người dùng |
| **Tiền điều kiện** | Đã đăng nhập |
| **Luồng cơ bản** | Use case bắt đầu khi người dùng đăng nhập vào và được chuyển đến trang chủ. Hệ thống sẽ lấy lên các bài viết lên trang chính. Người quản trị có thể thêm , sửa tài khoản .   * Thêm bài viết: * Người dùng click vào input để tạo bài viết. Hệ thống sẽ hiển thị lên một modal để nhập nội dùng cho bài viết gồm ảnh, video, text. * Người dùng nhập thông tin vào kích vào nút thêm mới . Hệ thống sẽ sinh ra một bài viết mới trong bảng Post và bài viết mới lên màn hình * Sửa tài khoản: * Người dùng nhấn vào nút ba chấm và chọn nút sửa trên một bài viết. Hệ thống sẽ lấy ra thông tin của bài viết người dùng muốn sửa và hiển thị lên màn hình. * Người dùng nhập thông tin mới cho bài viết và kích vào nút cập nhật. Hệ thống sẽ sửa thông tin của bài viết người dùng vừa sửa và hiển thị bài viết vừa sửa lên màn hình * Xóa bài viết: * Người dùng kích vào ba chấm và chọn nút xóa trên một bài viết. Hệ thống sẽ hiển thị lên màn hình yêu cầu xác nhận xóa. * Người quản trị kích vào nút “Đồng ý”. Hệ thống sẽ xóa bài viết được chọn khỏi bảng Post và hiển thị danh sách các bình luận đã cập nhật.   Use case kết thúc. |
| **Luồng rẽ nhánh** | Trong luồng cơ bản, nếu người dùng chưa có bài viết nào, hệ thống sẽ hiển thị "Không có bài viết". Use case kết thúc.  Tại bất kì thời điểm nào trong quá trình thực hiện ca sủ dụng nếu không kết nối được với CSDL thì hệ thống sẽ hiển thị thông báo lỗi và ca sử dụng kết thúc. |
| **Hậu điều kiện** | Bảng Post được cập nhật |

* Biểu đồ trình tự :



Hình 3.5. Biểu đồ trình tự quản lý bài viết

##### Use case: Quản lý bình luận

* Mô tả use case:

Bảng 3.5. Mô tả use case quản lý bình luận

|  |  |
| --- | --- |
| **Tên Use Case** | Quản lý bình luận |
| **Mô tả** | Use case này cho người dùng quản lý bình luận |
| **Người thực hiện** | Người dùng |
| **Tiền điều kiện** | Đã đăng nhập |
| **Luồng cơ bản** | Use case này bắt đầu khi người dùng chọn vào phần bình luận của bài viết để hiển thị phần comment.   * Thêm bình luận: * Người dùng click vào input bình luận. Hệ thống sẽ hiển thị lên một phần nhập nội dụng bình luận gồm ảnh, video, text. * Khi người dùng nhập vào các thông tin cho bình luận như , text, ảnh, video vào input và nhấn enter. Hệ thống sẽ tạo ra 1 bình luận mới trong bảng Comment, hiển thị lên bình luận mới và cập nhật lại số lượng bình luận. * Sửa bình luận: * Người dùng nhấn vào nút ba chấm và chọn nút bình luận trên một bài viết. Hệ thống sẽ lấy ra thông tin bình luận của người dùng muốn sửa và hiển thị lên màn hình. * Người dùng nhập thông tin mới cho bình luận và kích vào nút cập nhật. Hệ thống sẽ sửa thông tin của bài viết người dùng vừa sửa và hiển thị bài viết vừa sửa lên màn hình * Xóa nhân viên : * Người dùng kích vào ba chấm và chọn nút xóa trên một bình luận. Hệ thống sẽ hiển thị lên màn hình yêu cầu xác nhận xóa. * Người dùng kích vào nút “Đồng ý”. Hệ thống sẽ xóa bình luận được chọn khỏi bảng Comment và hiển thị danh sách các bình luận đã cập nhật. |
| **Luồng rẽ nhánh** | Tại bước 3 trong luồng cơ bản nếu người dùng không nhập thông tin, sẽ không gửi được. Use case kết thúc.  Tại bất kì thời điểm nào trong quá trình thực hiện ca sủ dụng nếu không kết nối được với CSDL thì hệ thống sẽ hiển thị thông báo lỗi và ca sử dụng kết thúc. |
| **Hậu điều kiện** | Bảng Comment được cập nhật |

* Biểu đồ trình tự:



Hình 3.6. Biểu đồ trình tự quản lí bình luận

##### Use case : Quản lý tin nhắn:

* Mô tả use case :

Bảng 3.6. Mô tả use case quản lý tin nhắn

|  |  |
| --- | --- |
| **Tên Use Case** | Quản lý tin nhắn |
| **Mô tả** | Use case này cho người dùng quản lý tin nhắn |
| **Người thực hiện** | Người dùng |
| **Tiền điều kiện** | Đã đăng nhập |
| **Luồng cơ bản** | * Use case này bắt đầu khi người dùng chọn người nhắn tin ở trong box chat hoặc ở menu. Box chat sẽ hiện ra vào sẽ có input để cho người dùng nhập để nhắn tin * Gửi tin nhắn: * Người dùng mở box chat. Hệ thống sẽ lấy ra các đoạn tin nhắn. * Người dùng nhập thông tin cho tin nhắn gồm text, video và nhấn enter để nhập. Hệ thống sẽ tạo ra 1 tin nhắn mới trong bảng message, hiển thị lên tin nhắn mới và cập nhật lại số lượng tin nhắn.   Use case kết thúc. |
| **Luồng rẽ nhánh** | 1) Tại bước 2b,3b trong luồng cơ bản nếu người dùng nhập thông tin thương hiệu không hợp lệ thì hệ thống sẽ không cho phép gửi tin nhắn.  2) Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sử dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc. |
| **Hậu điều kiện** | Bảng Conversation, Message được cập nhật |

* Biểu đồ trình tự:



Hình 3.7. Biểu đồ trình tự quản lí tin nhắn

##### Use case: Quản lý yêu cầu kết bạn:

* Mô tả use case:

Bảng 3.7. Mô tả use case quản lý yêu cầu kết bạn

|  |  |
| --- | --- |
| **Tên Use Case** | Quản lý yêu cầu kết bạn |
| **Mô tả** | Use case này cho phép người dùng xem, thêm, xóa được các yêu cầu kết bạn |
| **Người thực hiện** | Người dùng |
| **Tiền điều kiện** | Đã đăng nhập |
| **Luồng cơ bản** | * Use case bắt đầu khi người dùng vào trang kết bạn trên thahnh navbar. Hệ thống sẽ lấy ra các yêu cầu kết bạn đang có hoặc gợi ý kết bạn.Gửi tin nhắn: * Gửi yêu cầu kết bạn: * Người dùng click vào trang cá nhân của một người dùng nào đó rồi nhấn nút kết bạn hoặc nhấn nút gửi yêu cầu ở phần gợi ý kết bạn. * Sau khi người dùng nhấn nút gửi. Hệ thống sẽ sinh ra một yêu cầu mới mới trong bảng FriendRequest và sẽ sinh ra một yêu cầu kết bạn.   Use case kết thúc.   * Hủy yêu cầu kết bạn: * Người dùng nhấn vào nút từ chối trên yêu cầu kết bạn. * Sau khi người dùng nhấn vào nút từ chối. Hệ thống sẽ xóa bản ghi từ bảng FriendRequest và cập nhật lại danh sách các yêu cầu kết bạn.   Use case kết thúc. |
| **Luồng rẽ nhánh** | Tại bất kỳ thời điểm nào nếu use case không kết nối được với cơ sở dữ liệu thì hệ thống sẽ báo lỗi và use case kết thúc. |
| **Hậu điều kiện** | Bảng FriendRequest được cập nhật |

* Biểu đồ trình tự:



Hình 3.8. Biểu đồ trình tự quản lí yêu cầu kết bạn

##### Use case: Quản lý thông báo:

* Mô tả use case:

Bảng 3.8. Mô tả use case quản lý thông báo

|  |  |
| --- | --- |
| **Tên Use Case** | Quản lý thông báo |
| **Mô tả** | Use case này cho phép người dùng quản lý thông báo. |
| **Người thực hiện** | Người dùng |
| **Tiền điều kiện** | Đã đăng nhập |
| **Luồng cơ bản** | Use case bắt đầu khi người dùng kích vào nút thông báo trên thành navbar. Box thông báo sẽ hiện ra. Hệ thống sẽ lấy các bản ghi thông báo từ bảng Notification gồm text, url, thông tin người gửi.   * Đánh dấu là đã đọc: * Người dùng click vào một thông báo bất kỳ. Hệ thống sẽ cập nhật lại thông báo trong bảng Notification đó vào đánh dấu là đã đọc   Use case kết thúc.   * Xóa thông báo: * Người click vào nút 3 chấm trên thông báo và chọn nút xóa. Hệ thống sẽ xóa bản ghi thông báo trong bảng Notification và cập nhật lại danh sách thông báo.   Use case kết thúc. |
| **Luồng rẽ nhánh** | Tại bất kỳ thời điểm nào nếu use case không kết nối được với cơ sở dữ liệu thì hệ thống sẽ báo lỗi và use case kết thúc. |
| **Hậu điều kiện** | Bảng Notification được cập nhật |

* Biểu đồ trình tự:



Hình 3.9. Biểu đồ trình tự quản lí thông báo

#### Xây dựng cơ sở dữ liệu

##### Biểu đồ Entity RelationShip

A screenshot of a computer

Description automatically generated

Hình 3.10. Biểu đồ Entity Relationship

##### Các bảng trong CSDL

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

Hình 3.11. Hình các bảng trong CSDL

### Kết quả đạt được

A screenshot of a computer

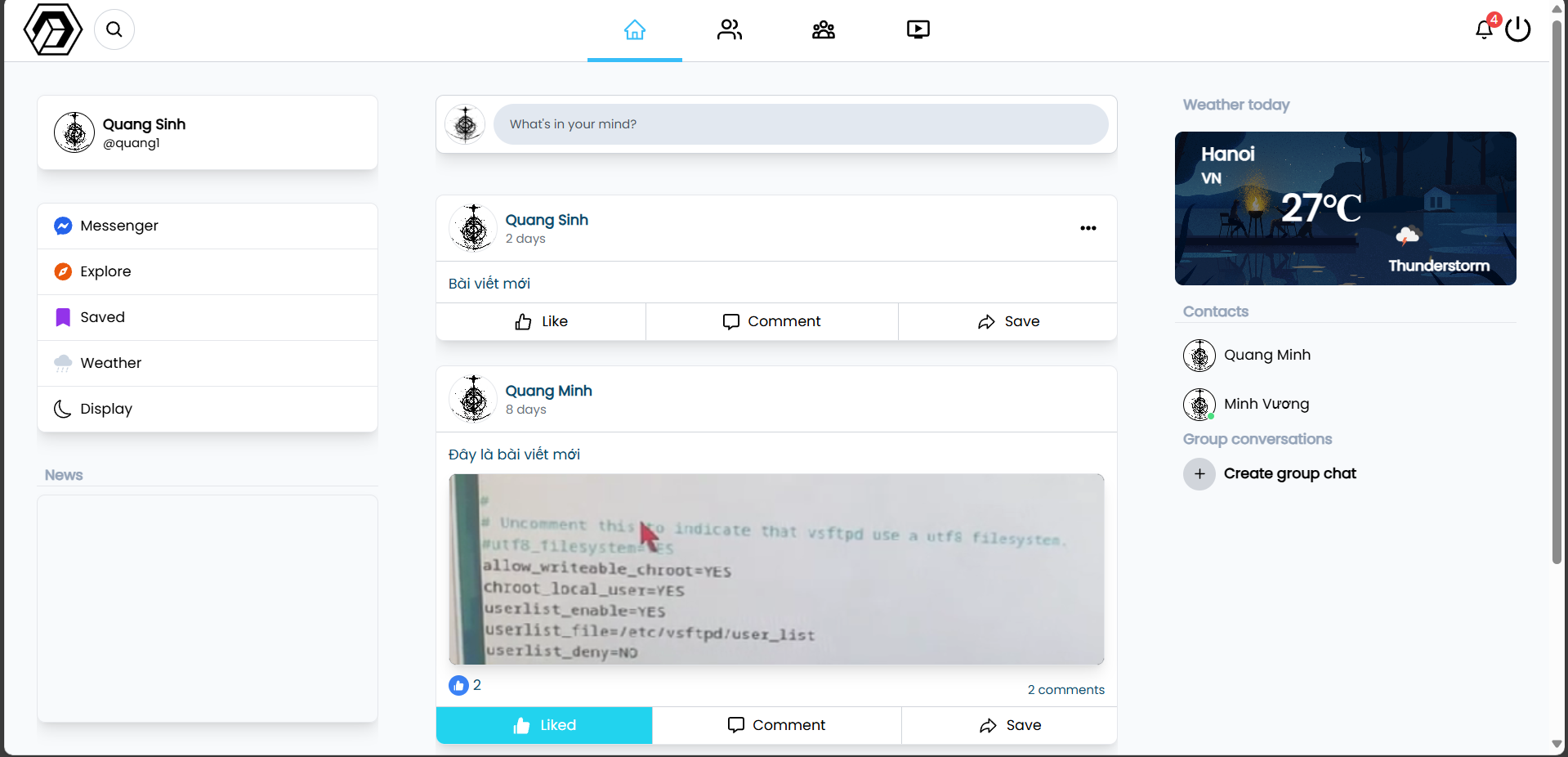
Description automatically generated

Hình 3.12. Màn hình đăng nhập

A screenshot of a computer

Description automatically generated

Hình 3.13. Màn hình đăng ký



Hình 3.14. Màn hình trang chủ

A screenshot of a video game

Description automatically generated

Hình 3.15. Màn hình trang cá nhân

A screenshot of a computer

Description automatically generated

Hình 3.16. Màn hình hội thoại

A screenshot of a computer

Description automatically generated

Hình 3.17. Màn hình quản lý thông báo

A screenshot of a computer

Description automatically generated

Hình 3.18. Màn hình quản lý kết bạn

### Kiểm thử

#### Kế hoạch kiểm thử

Bảng 3.9. Thông tin kiểm thử

|  |  |  |  |
| --- | --- | --- | --- |
| Tên dự án | UniSocial | Người lập | Lê Minh Quang |
| Mã dự án | UNSC | Người kiểm tra/ xét duyệt | Lê Minh Quang |
| Mã tài liệu | UNSC \_TL\_01 | Ngày lập | 01/05/2024 |
| Phiên bản | 1 |

Bảng 3.10. Nội dung kiểm thử

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên chức năng** | **Mô tả** |
| 1 | Kiểm thử chức năng | Kiểm thử chức năng:   * Đăng nhập |
| 2 | Kiểm thử giao diện | Kiểm thử chức năng:   * Đăng nhập |

#### Thực thi kiểm thử

##### Kiểm thử chức năng :

Bảng 3.11. Kiểm thử chức năng hệ thống

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Tên test case** | **Kịch bản kiểm thử** | **Các bước thực hiện** | **Kết quả mong đợi** | **Pass(X)**  **/Fail** |
| **Đăng nhập** | | | | | |
| 1 | Đăng nhập thành công | Người dùng nhập đúng tên đăng nhập và mật khẩu | 1. Người dùng chọn nút đăng nhập trên menu.  2. Người dùng nhập tên đăng nhập, mật khẩu và kích vào nút đăng nhập. | 1. Hệ thống hiển thị màn hình đăng nhập.  2. Hệ thống hiển thị màn hình chính của người dùng sau khi đã đăng nhập thành công. | X |
| 2 | Đăng nhập không thành công 1 | Người dùng nhập sai tên hoặc bỏ trống tên đăng nhập | 1. Người dùng chọn nút đăng nhập trên menu.  2. Người dùng nhập sai hoặc bỏ trống tên đăng nhập, nhập mật khẩu và kích vào nút đăng nhập. | 1. Hệ thống hiển thị màn hình đăng nhập.  2. Hệ thống hiển thị màn hình thông báo lỗi do sai tên đăng nhập và quay lại màn hình đăng nhập. | X |
| 3 | Đăng nhập không thành công 2 | Người dùng nhập sai hoặc bỏ trống mật khẩu | 1. Người dùng chọn nút đăng nhập trên menu.  2. Người dùng nhập tên đăng nhập, nhập sai hoặc bỏ trống mật khẩu và kích vào nút đăng nhập. | 1. Hệ thống hiển thị màn hình đăng nhập.  2. Hệ thống hiển thị màn hình thông báo lỗi do sai mật khẩu và quay lại màn hình đăng nhập. | X |
| 4 | Đăng nhập không thành công 3 | Người dùng nhập sai tên hoặc mật khẩu x lần | 1. Người dùng chọn nút đăng nhập trên menu.  2. Người dùng nhập sai tên đăng nhập và mật khẩu và kích vào nút đăng nhập x lần. | 1. Hệ thống hiển thị màn hình đăng nhập.  2. Hệ thống hiển thị mà hình thông báo lỗi và khóa màn hình, người dùng không thể đăng nhập lại. | X |
| 5 | Đăng nhập không thành công 4 | Không kết nối được với cơ sở dữ liệu | 1. Người dùng chọn nút đăng nhập trên menu.  2. Người dùng nhập tên đăng nhập, mật khẩu và kích vào nút đăng nhập. | 1. Hệ thống hiển thị màn hình đăng nhập.  2. Hệ thống hiển thị thông báo Không kết nối được với cơ sở dữ liệu. | X |

##### Kiểm thử giao diện

Bảng 3.12. Bảng mô tả kiểm thử giao diện

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Tên test case** | **Kịch bản kiểm thử** | **Các bước thực hiện** | **Kết quả mong đợi** | **Pass(X)**  **/Fail** |
| 1 | Màn hình chung. | Đảm bảo có thể dùng chuột, các phím tab, lên, xuống, trái, phải...để điều hướng các từ màn hình này sang màn hình khác, từ trường này qua trường khác. | 1. Người dùng bật màn hình giao diện người dùng.  2. Chọn di chuyển giữa các trường bằng các phím tab, phím điều hướng (lên xuống trái phải)" | 1. Hệ thống hiển thị giao diện người dùng đang bật  2. Trỏ chuột di chuyển đúng yêu cầu nghiệp vụ. | X |
| 2 | Màn hình đăng nhập | Đảm bảo trường tên đăng nhập và trường mật khẩu bắt buộc nhập phải được đánh dấu. | Nhìn vào các giao diện đang mở và đối chiếu với Figma | 1. Hệ thống hiển thị những trường cần nhập dữ liệu có dấu \*. | X |

##### Kết quả kiểm thử :

* Tỉ lệ test case đạt (passed): 100%
* Tỉ lệ test case không đạt (failed): 0%

KẾT LUẬN

1. Những kết quả sau quá trình nghiên cứu

Đề tài “*Xây dựng website mạng xã hội ReactJS và ASP.NET Core*” xuất phát từ những nhu cầu thực tế trong lĩnh vực hiện nay. Mạng xã hội đang ngày càng cần thiết, giúp người dùng dễ dàng liên lạc, trao đổi thông tin với nhau một cách nhanh chóng và thuận tiện.

Qua quá trình thực hiện đề tài, em đã tìm hiểu, tích lũy và học hỏi thêm được nhiều kinh nghiệm cũng như kiến thức công nghệ mới.

**Những kết quả đạt được:**

* **Về công nghệ:**
* Hiểu được quá trình thiết kế một website thực tế đi từ bước cơ bản: khảo sát dự án, phân tích hệ thống, thiết kế, thực hiện, kiểm thử, triển khai, bảo trì.
* Xây dựng thành công website đáp ứng nhu cầu sử dụng mạng xã hội.
* Nắm được các kiến thức xây dựng giao diện website: HTML, CSS, JavaScript.
* Nắm vững về cách xây dựng website thông qua thư viện, framework: ReactJS trong lập trình.
* Sử dụng thành thạo các công cụ hỗ trợ lập trình: Visual Studio Code, Visual Studio .
* Sử dụng công cụ thiết kế figma.
* **Về cài đặt chương trình:**
* Lấy code từ github
* Sử dụng VSCode chạy lệnh npm install để tải các package rồi chạy lệnh npm run dev để chạy server localhost cho fron-tend
* Sử dụng Visual Studio để chạy code back-end.
* **Tính năng khác:**
* Giao diện đẹp gây ấn tượng thân thiện với khách hàng.
* Tính bảo mật.

1. Kiến nghị về những nghiên cứu tiếp theo.

* *Tích hợp Chatbot:*

Sử dụng chatbot để đưa ra các thông tin về hoạt động của bạn bè, những bộ phim mà họ xem gần đây hay gợi ý những bộ phim phù hợp với bạn.

* *Tính Năng Kho Game:*

Cung cấp các tựa game để người dùng thỏa thích giải trí sau những ngày làm việc mệt mỏi.

* *Tính Năng Hỗ Trợ Khách Hàng:*

Tạo tính năng trò chuyện trực tuyến hoặc hỗ trợ qua điện thoại để giúp người dùng giải đáp thắc mắc và giải quyết vấn đề.

# TÀI LIỆU THAM KHẢO

|  |  |
| --- | --- |
| [1] | Phùng Đức Hòa (Chủ biên); Hoàng Quang Huy; Trịnh Bá Quý, Giáo trình Nhập môn công nghệ phần mềm, Hà Nội: NXB Thống Kê, 2019. |
| [2] | Vũ Thị Dương; Phùng Đức Hòa; Nguyễn Thị Hương Lan, Giáo trình phân tích thiết kế hướng đối tượng, Hà Nội: Nhà Xuất Bản Khoa Học Và Kĩ Thuật, 2015. |
| [3] | Tham khảo từ Microsoft : “ https://docs.asp.net/en/latest/tutorials/first-mvc-app/index.html” |
| [4] | Tham khảo từ :” https://react.dev/learn” |
| [5] | Tham khảo từ :”Fullstack.edu.vn” |