TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**LƯU MINH NHIỀU - 52100456**

**BLOGPOTS**

**SỬ DỤNG ASP.NET CORE**

**BÁO CÁO CUỐI KỲ**

**CÔNG NGHỆ .NET**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2024**

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**LƯU MINH NHIỀU - 52100456**

**BLOGPOTS**

**SỬ DỤNG ASP.NET CORE**

**BÁO CÁO CUỐI KỲ**

**CÔNG NGHỆ .NET**

Người hướng dẫn

**Dzoãn Xuân Thanh**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2024**

**LỜI CẢM ƠN**

Trước tiên, em xin chân thành cảm ơn Thầy Dzoãn Xuân Thanh đã hướng dẫn và hỗ trợ em trong quá trình thực hiện báo cáo này. Những kiến thức và kinh nghiệm quý báu từ Thầy đã giúp em có cái nhìn sâu sắc hơn về công nghệ ASP.NET. Em cũng xin cảm ơn các bạn đã theo dõi và góp ý cho bài báo cáo của em. Hy vọng rằng những gì em trình bày hôm nay sẽ mang lại giá trị và góp phần vào sự phát triển chung của lớp. Một lần nữa, em xin chân thành cảm ơn!

*TP. Hồ Chí Minh, ngày 30 tháng 11 năm 2024*

*Tác giả*

*(Ký tên và ghi rõ họ tên)*

*Nhiều*

*Lưu Minh Nhiều*

**CÔNG TRÌNH ĐƯỢC HOÀN THÀNH**

**TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

Tôi xin cam đoan đây là công trình nghiên cứu của riêng tôi và được sự hướng dẫn khoa học của Dzoãn Xuân Thanh. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong Dự án còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

**Nếu phát hiện có bất kỳ sự gian lận nào tôi xin hoàn toàn chịu trách nhiệm về nội dung Dự án của mình**. Trường Đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

*TP. Hồ Chí Minh, ngày 30 tháng 11 năm 2024*

*Tác giả*

*(Ký tên và ghi rõ họ tên)*

*Nhiều*

*Lưu Minh Nhiều*

**BLOGPOTS SỬ DỤNG ASP.NET CORE**

**TÓM TẮT**

(Dự án BlogPots được phát triển với công nghệ ASP.NET Core kết hợp với Entity Framework (EF) Core để tạo ra một nền tảng quản lý blog mạnh mẽ và linh hoạt. Hệ thống sử dụng PostgreSQL làm cơ sở dữ liệu, đảm bảo hiệu suất và khả năng mở rộng.

Phần giao diện người dùng được xây dựng bằng Vue.js, cho phép tạo ra trải nghiệm người dùng mượt mà và tương tác cao.

**BLOGPOTS USING ASP.NET CORE**

**ABSTRACT**

The BlogPots project is developed using ASP.NET Core combined with Entity Framework (EF) Core to create a powerful and flexible blog management platform. PostgreSQL is utilized as the database, ensuring performance and scalability.

The user interface is built with Vue.js, allowing for a smooth and highly interactive user experience.

**MỤC LỤC**

[DANH MỤC HÌNH VẼ vii](#_Toc7149)

[DANH MỤC BẢNG BIỂU viii](#_Toc2371)

[DANH MỤC CÁC CHỮ VIẾT TẮT ix](#_Toc11059)

[CHƯƠNG 1. MỞ ĐẦU VÀ TỔNG QUAN ĐỀ TÀI 1](#_Toc13689)

[1.1 Lý do chọn đề tài 1](#_Toc31889)

[1.2 Kiến trúc tổng thể 1](#_Toc4865)

[1.3 Sơ đồ ERD (Tạm Thời) 2](#_Toc29757)

[1.4 Khác biệt giữa Client Side Rendering và Server Side Rendering 2](#_Toc27600)

[CHƯƠNG 2. CHI TIẾT DỰ ÁN 3](#_Toc22641)

[2.1 Auth: Authentication and Authorization 3](#_Toc26867)

[2.1.1 Authentication 3](#_Toc752)

[2.1.2 Authorization 3](#_Toc9819)

[2.2 Cấu trúc dự án 13](#_Toc11297)

[2.3 Giao diện Client Side 15](#_Toc15917)

[2.3.1 Home (Route: /) 16](#_Toc929)

[2.3.2 Blogs List (Route: /blogs) 17](#_Toc20336)

[2.3.3 Blog Detail (Tạm Thời) (Route: /blogs/[blogId, blogname, blogtitle…]) 18](#_Toc10930)

[2.3.4 User Profile (Tạm Thời) (Route: /user/[userId]/[Route~]) 21](#_Toc25422)

[2.3.5 Post Create (Tạm Thời) (Route: blog/[…]/post/create) 21](#_Toc1792)

[CHƯƠNG 3. TỔNG KẾT 23](#_Toc27976)

[CHƯƠNG 4. KẾT LUẬN 24](#_Toc22660)

[4.1 Kết luận 24](#_Toc17925)

[4.2 Hướng phát triển 24](#_Toc5011)

[TÀI LIỆU THAM KHẢO 25](#_Toc15784)

# DANH MỤC HÌNH VẼ

[Hình 1 : Sơ đồ ERD 2](#_Toc26073)

[Hình 2 : Register UI 4](#_Toc15202)

[Hình 3 : RegisterPage.vue 5](#_Toc11869)

[Hình 4 : RegisterForm.vue 6](#_Toc23105)

[Hình 5 : Register.cs 7](#_Toc26930)

[Hình 6 : Login UI 8](#_Toc16807)

[Hình 7 : LoginPage.vue 9](#_Toc25101)

[Hình 8 : LoginForm.vue 10](#_Toc25241)

[Hình 9 : Login.cs 11](#_Toc5810)

[Hình 10 : JWTServices.cs 11](#_Toc21138)

[Hình 11 : BcryptService.cs 12](#_Toc16148)

[Hình 12 : Project Structure (1) 14](#_Toc13512)

[Hình 13 : Project Structure (2) 14](#_Toc8382)

[Hình 14 : Project Structure (3) 15](#_Toc18608)

[Hình 15 : Home UI 17](#_Toc12657)

[Hình 16 : Blogs List UI 18](#_Toc16836)

[Hình 17 : Blog Detail UI 21](#_Toc19727)

[Hình 18 : User Profile UI 21](#_Toc19588)

[Hình 19 : Post Create UI 22](#_Toc695)

# DANH MỤC BẢNG BIỂU

[Bảng 4.1: Thống kê kiểu thực thể trong tập VLSP 2016 4](#_Toc142677565)

# DANH MỤC CÁC CHỮ VIẾT TẮT

|  |  |
| --- | --- |
| API | Application Programming Interface |
| SSR | Server Side Rendering |
| CSR | Client Side Rendering |
| SQL | Structure Query Language |
| EF | Entity Framework |

# MỞ ĐẦU VÀ TỔNG QUAN ĐỀ TÀI

## Lý do chọn đề tài

Xây dựng và phát triển 1 dự án riêng cho bản thân với đề tài khác trong khoảng kinh nghiệm về 1 Software Development. Cùng với đó là được tiếp xúc với công nghệ mới như .Net hay các loại cơ sở dữ liệu, framework mới đối với bản thân như: PostgreSQL, VueJS…

Trong báo cáo này, tôi sẽ trình bày về dự án blog được phát triển bằng Client Side Rendering (CRS) ASP.NET Core API, sử dụng EF Core cho truy cập dữ liệu, PostgreSQL làm cơ sở dữ liệu và VueJS với Composition API cho giao diện người dùng.

## Kiến trúc tổng thể

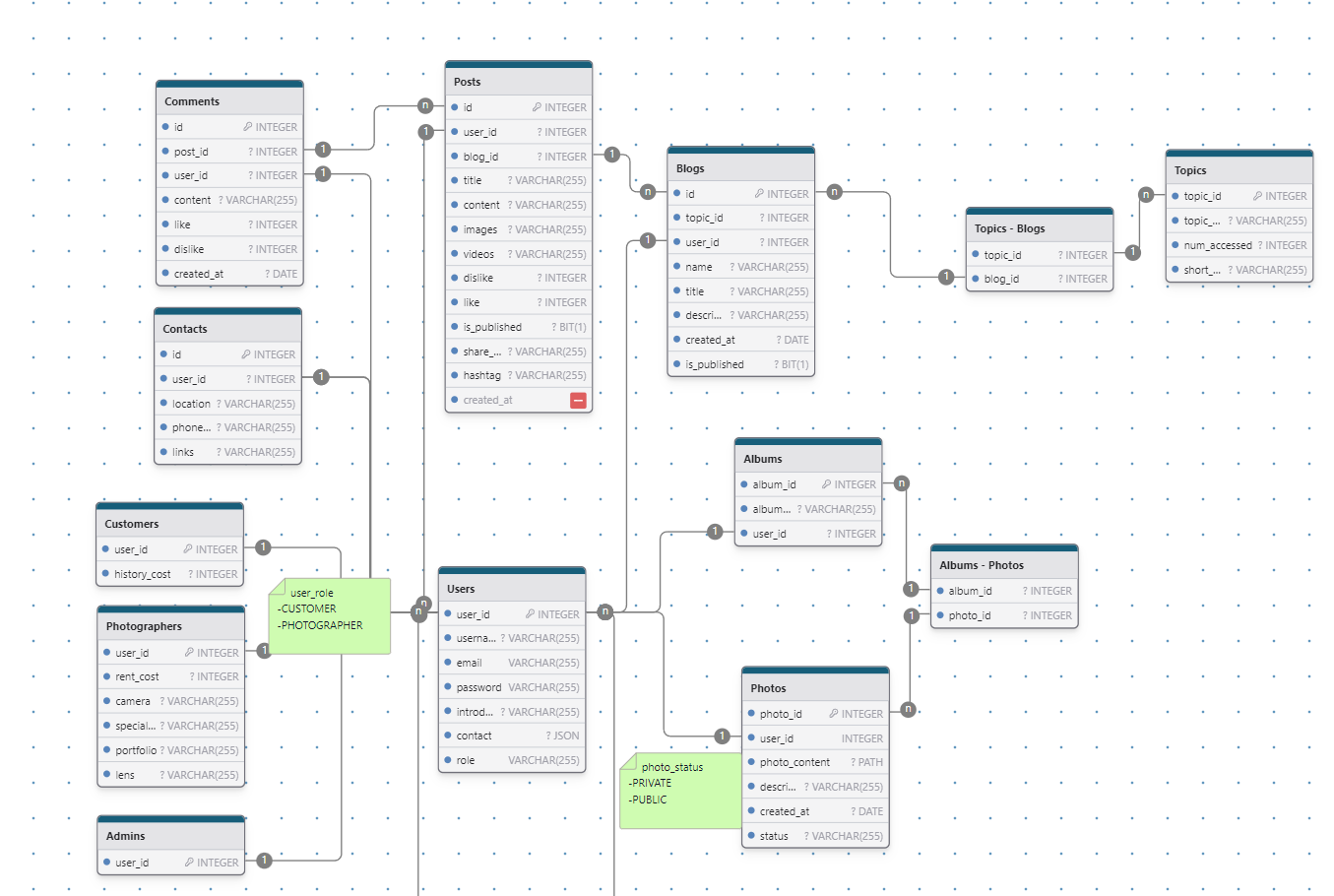
Back-end:

ASP.NET Core API.

Sử dụng EF Core để tương tác với PostgreSQL.

Front-end: VueJS (Composition API).

## Sơ đồ ERD



Hình 1: Sơ đồ ERD

## Khác biệt giữa Client Side Rendering và Server Side Rendering

Server Side Rendering (SSR):

Các trang web được tạo ra trên server và gửi đến trình duyệt.

Tối ưu SEO và tốc độ tải trang đầu tiên.

Client Side Rendering (CSR):

Nội dung được xử lý trên trình duyệt.

Gọi API để lấy dữ liệu, giúp trải nghiệm người dùng mượt mà hơn mà không cần tải lại trang.

# CHI TIẾT DỰ ÁN

## Auth: Authentication and Authorization

### Authentication

Authentication (xác thực) là quá trình xác định danh tính của người dùng khi họ đăng nhập vào ứng dụng. Trong dự án này, tôi đã áp dụng các biện pháp sau để đảm bảo tính an toàn và bảo mật:

- Mã hóa mật khẩu:

Khi người dùng đăng ký tài khoản, mật khẩu của họ được mã hóa bằng phương pháp bcrypt trước khi lưu trữ vào cơ sở dữ liệu.

Bằng cách này, ngay cả khi cơ sở dữ liệu bị xâm nhập, thông tin mật khẩu vẫn được bảo vệ, giảm thiểu rủi ro cho người dùng.

- Tạo JSON Web Token (JWT):

Khi người dùng đăng nhập thành công, hệ thống sẽ tạo một JWT.

Token này chứa thông tin người dùng và được mã hóa để đảm bảo tính toàn vẹn.

Token sẽ được gửi lại cho người dùng và được sử dụng trong các yêu cầu tiếp theo để xác thực danh tính.

### Authorization

Phân quyền là quá trình xác định quyền truy cập của người dùng vào các tài nguyên trong ứng dụng. Trong dự án này, tôi đã thực hiện phân quyền như sau:

Kiểm tra quyền truy cập:

- Mỗi khi người dùng gửi yêu cầu đến server, JWT sẽ được gửi kèm theo.

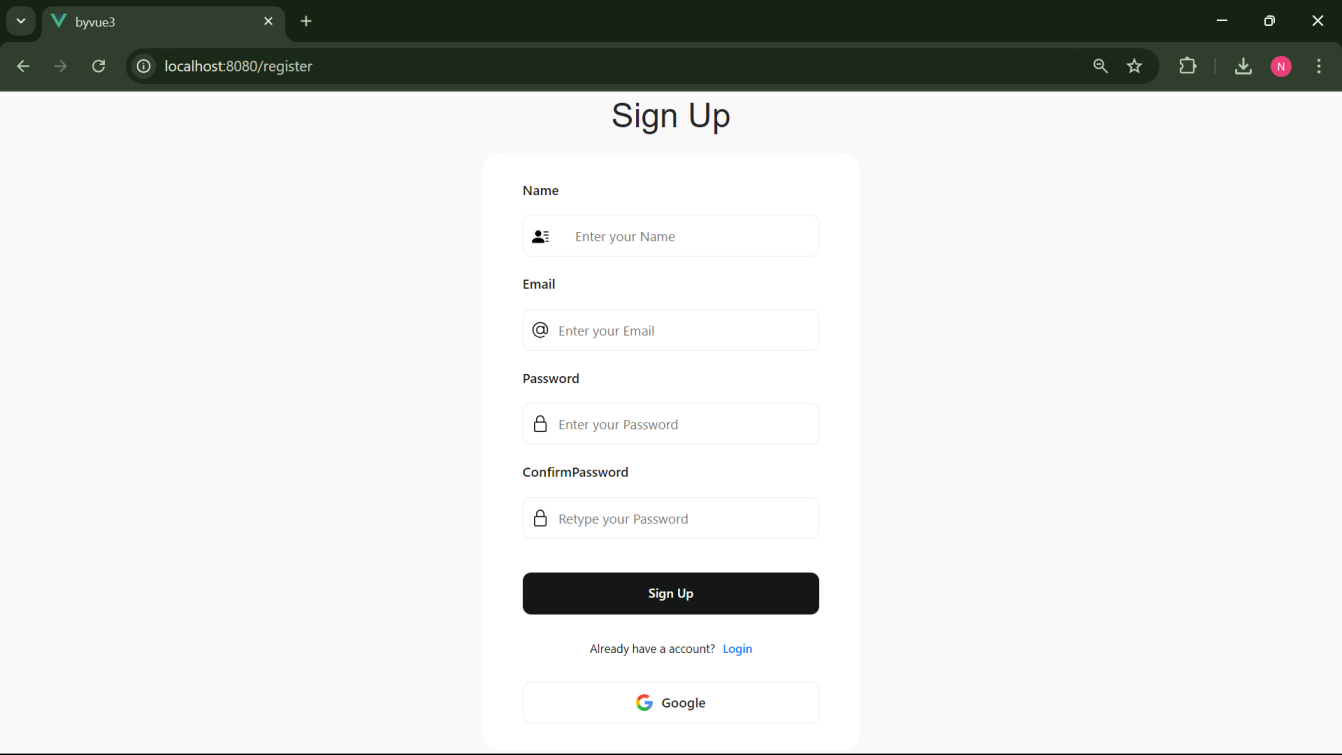
- Server sẽ giải mã token và kiểm tra thông tin người dùng, bao gồm vai trò và quyền hạn của họ.

Quy định quyền truy cập:

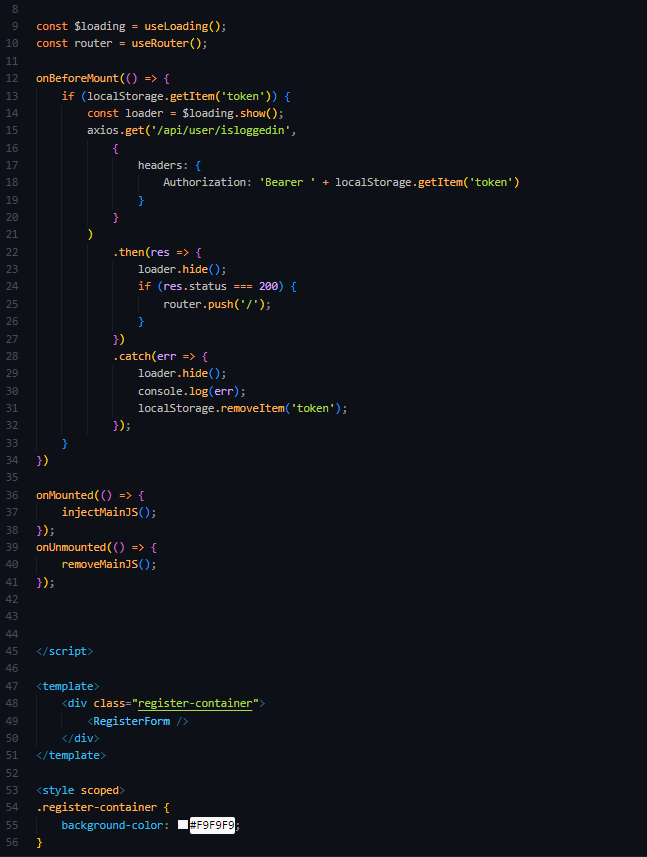
- Dựa trên vai trò của người dùng (ví dụ: quản trị viên, người dùng thường), hệ thống sẽ quyết định quyền truy cập vào các tài nguyên khác nhau.

- Chỉ những người dùng có quyền hợp lệ mới có thể thực hiện các thao tác nhất định như tạo, chỉnh sửa hoặc xóa bài viết.

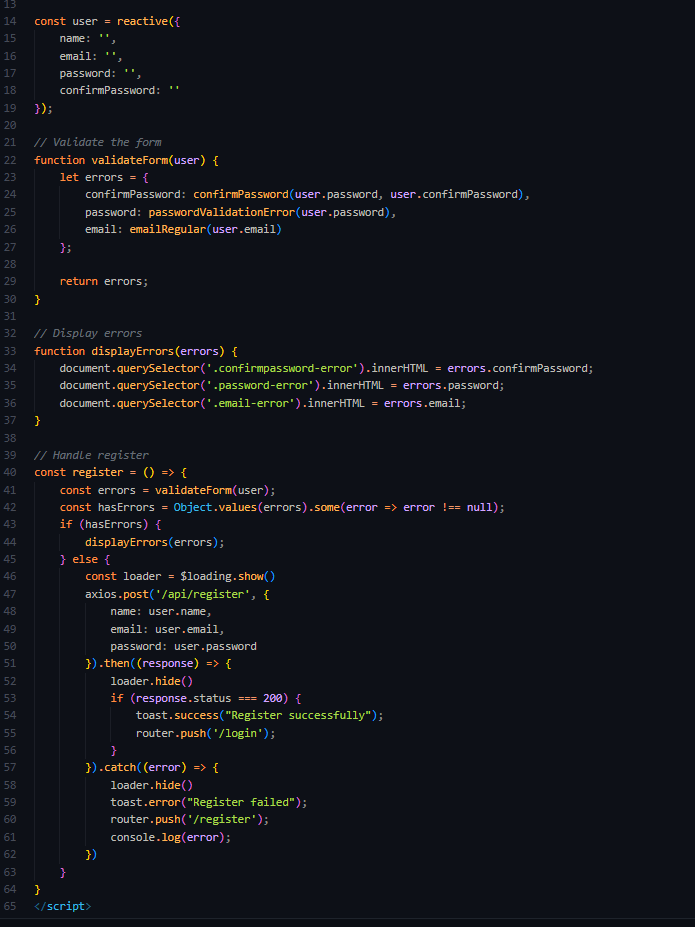
Register - FE:



Hình 2: Register UI

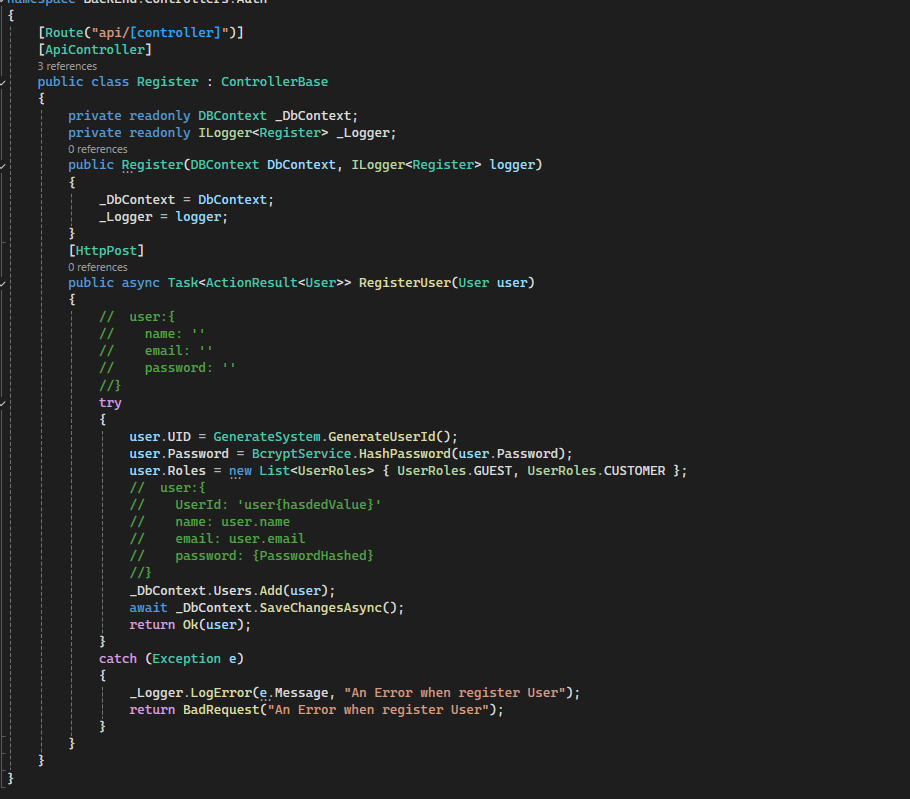


Hình 3: RegisterPage.vue



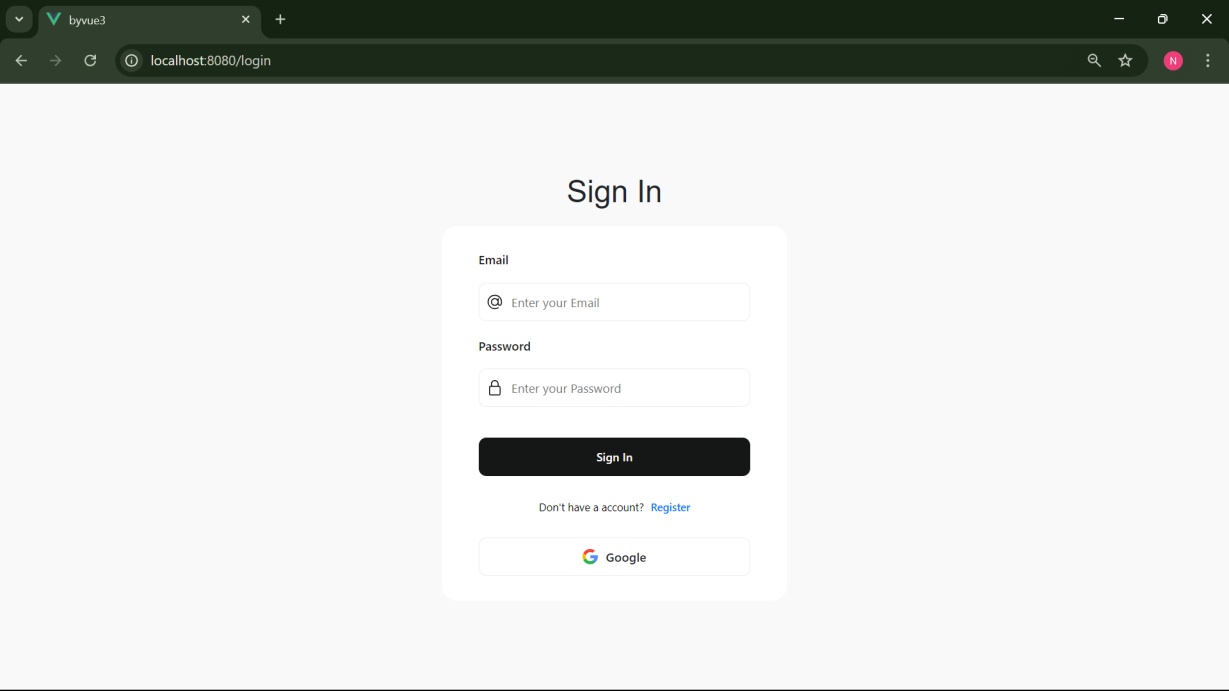
Hình 4: RegisterForm.vue

Register - BE:

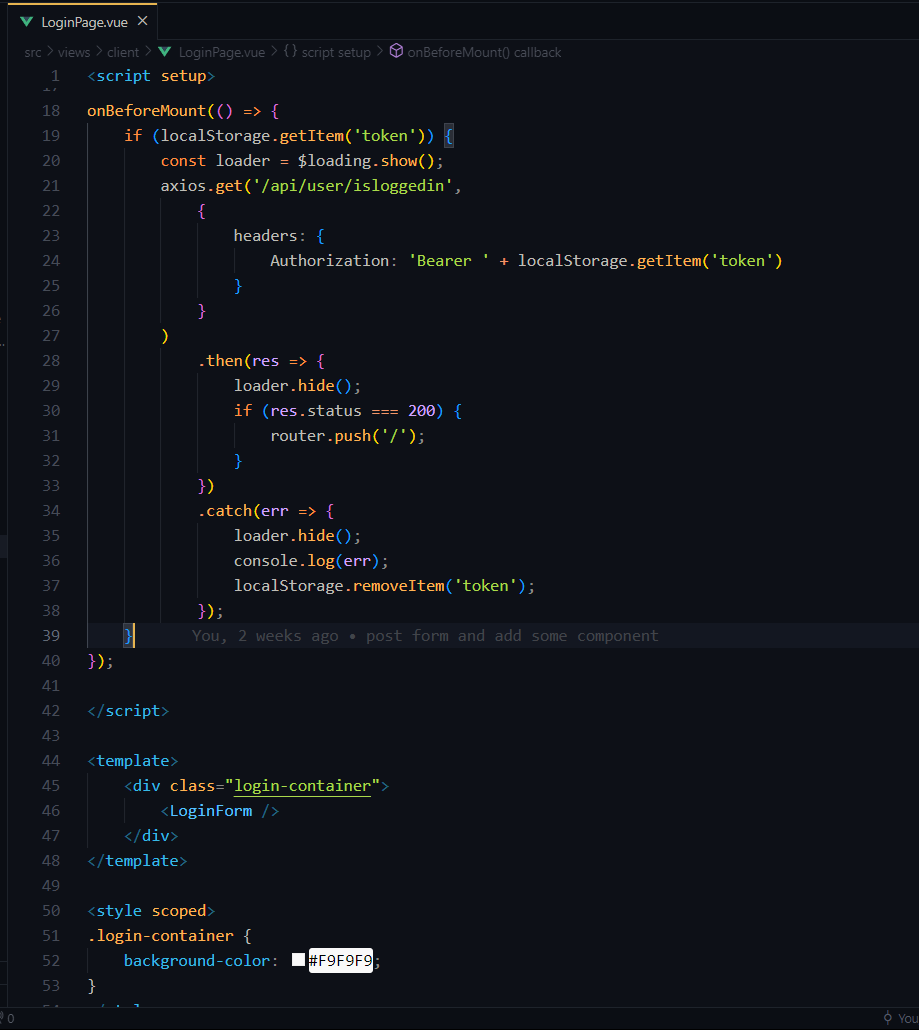


Hình 5: Register.cs

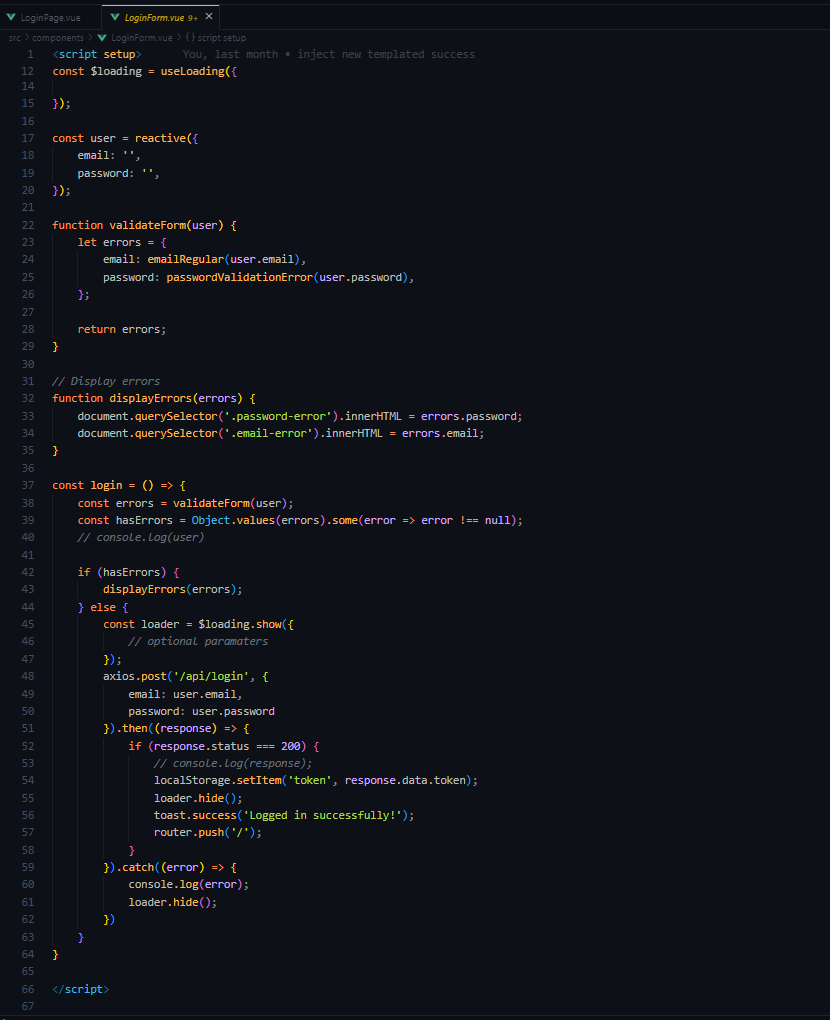
Login - FE:



Hình 6: Login UI

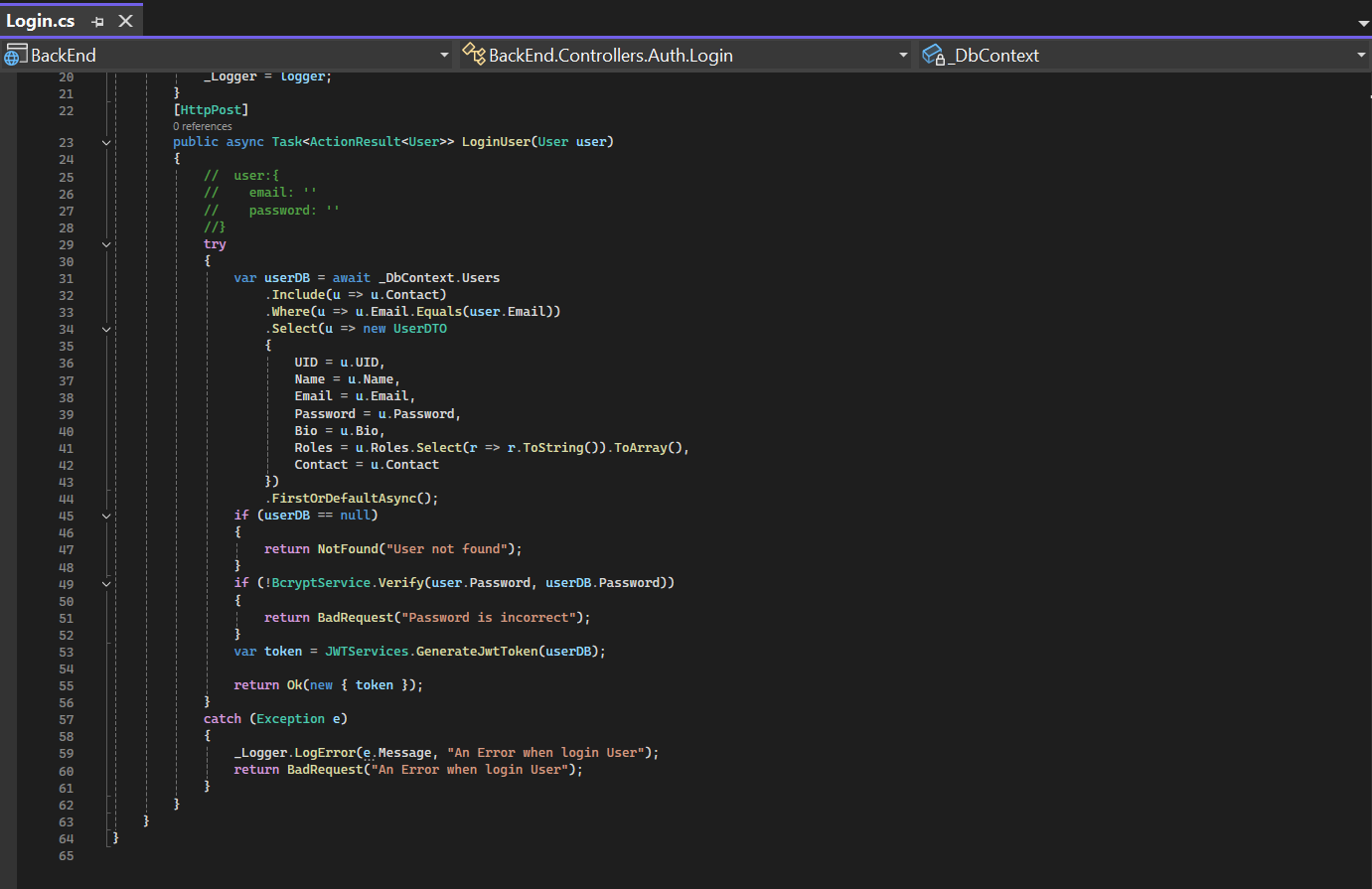


Hình 7: LoginPage.vue



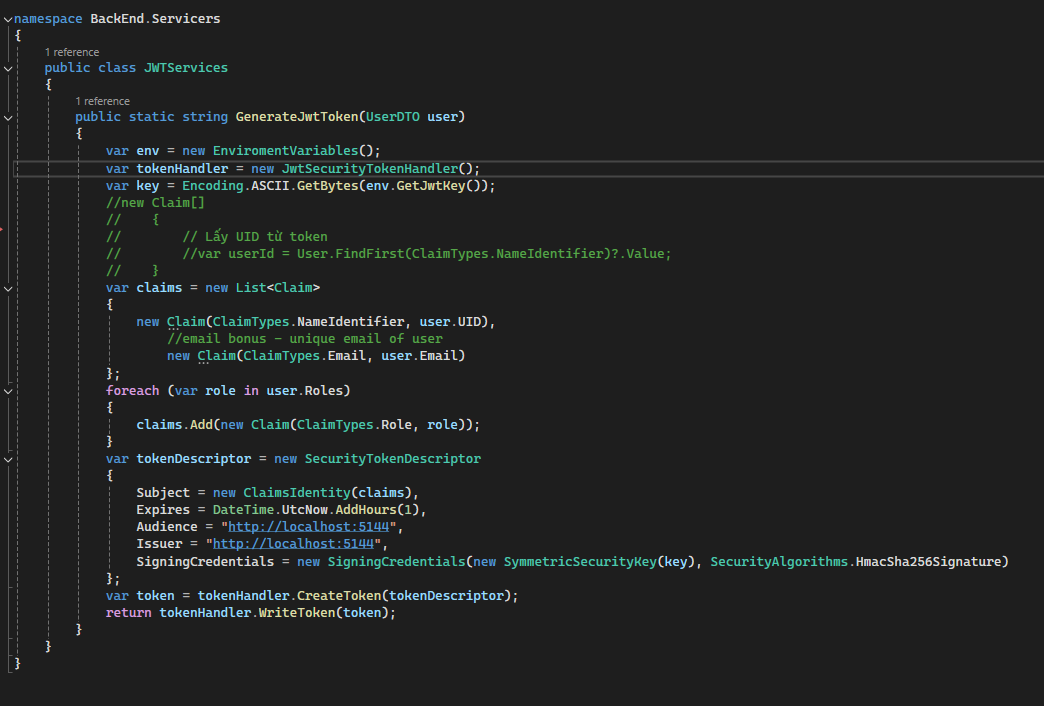
Hình 8: LoginForm.vue

Login - BE:

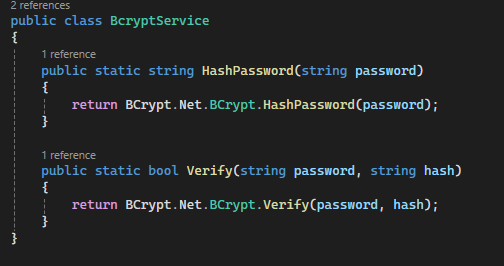


Hình 9: Login.cs

Services Supported:



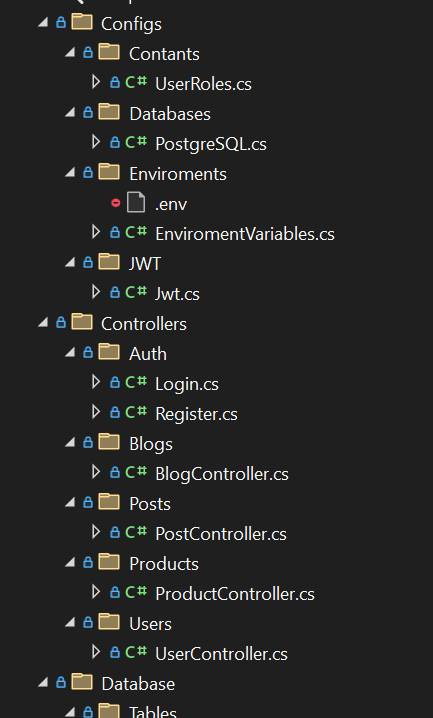
Hình 10: JWTServices.cs



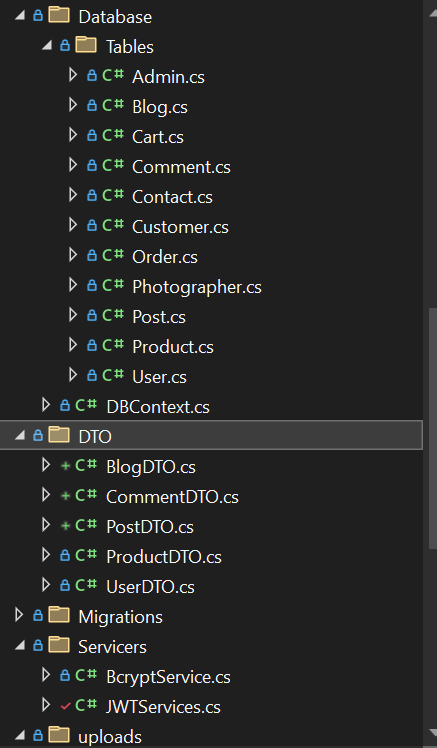
Hình 11: BcryptService.cs

## Cấu trúc dự án

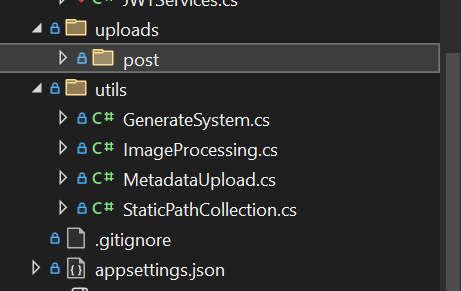
Dự án blog của tôi được phát triển theo kiến trúc microservices, cho phép ứng dụng trở nên linh hoạt, dễ mở rộng và bảo trì hơn. Dưới đây là các thành phần chính trong cấu trúc microservices của dự án:



Hình 12: Project Structure (1)



Hình 13: Project Structure (2)

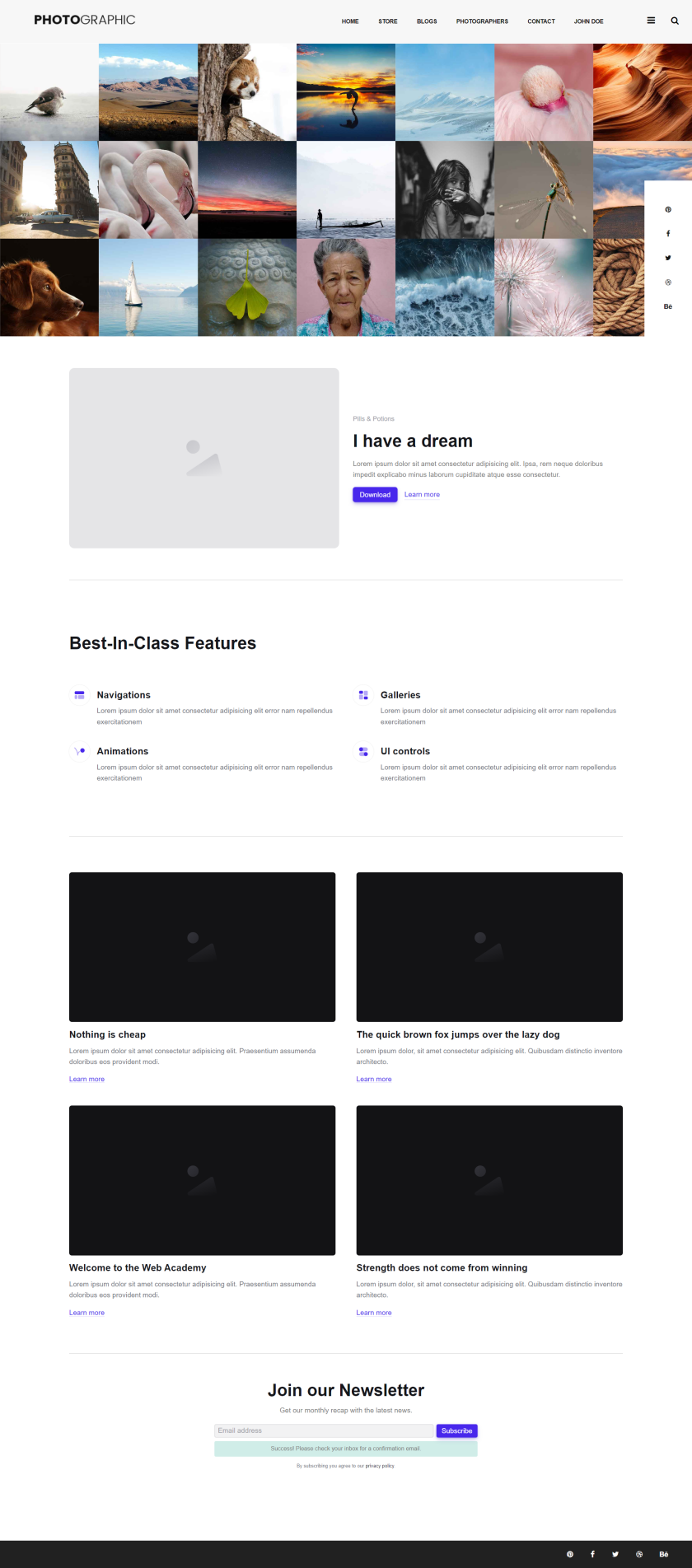


Hình 14: Project Structure (3)

## Giao diện Client Side

Vì đang trong giai đoạn phát triển nên tôi sẽ chỉ demo về UI của VueJS.

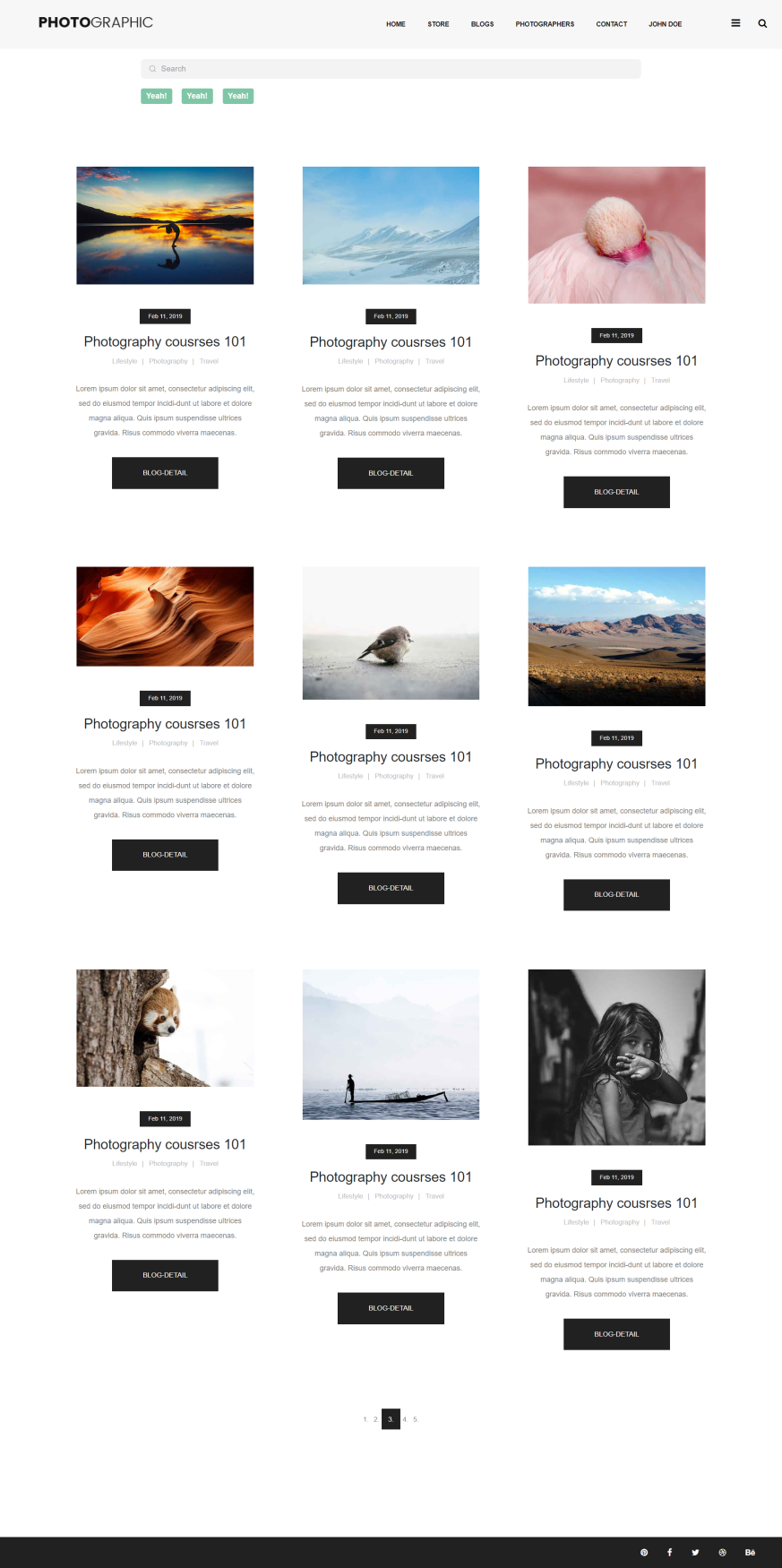
### Home (Route: /)



Hình 15: Home UI

### Blogs List (Route: /blogs)

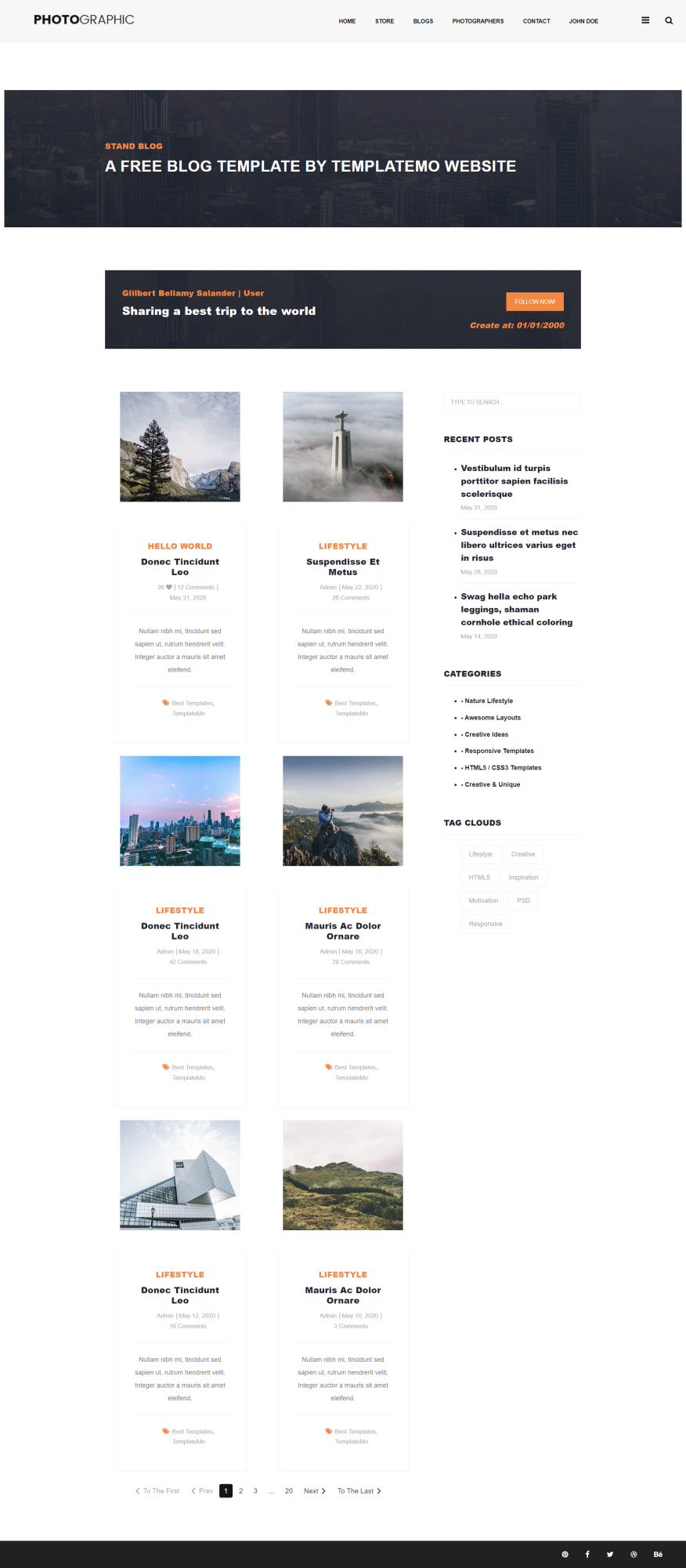
Route truy cập List Blog chung được public của các user có trong hệ thống. Tương tự như Route Discovery (Khám phá) trong những hệ sinh thái tương tự.



Hình 16: Blogs List UI

### Blog Detail (Tạm Thời) (Route: /blogs/[blogId, blogname, blogtitle…])

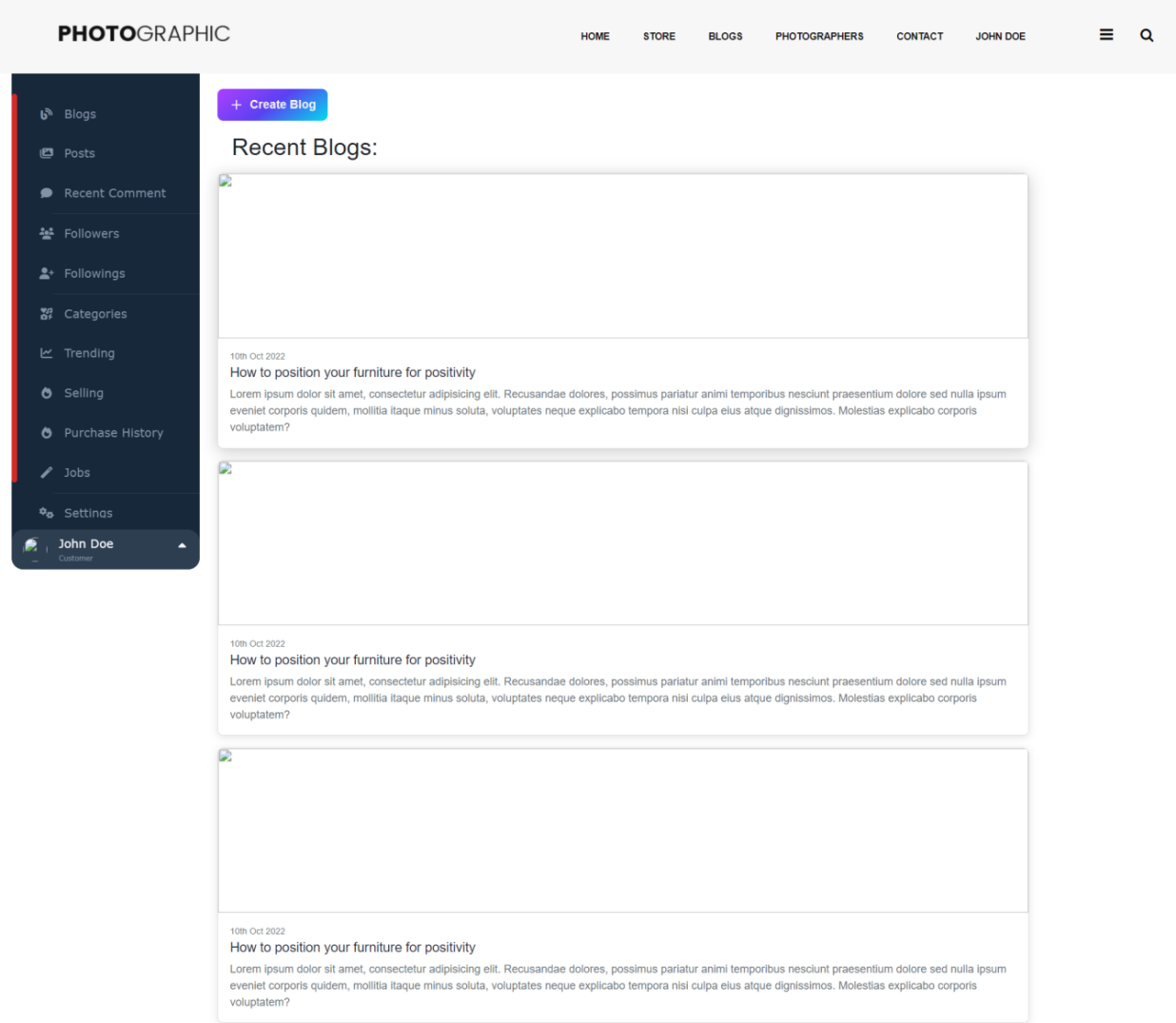
Route về chi tiết của 1 bài blog bao gồm các bài post được public thuộc bài blog đó. Bên trong Route […] là những slug sẽ được custom riêng tùy vào tình huống để đảm bảo việc SEO là hợp lý trong quá trình phát triển.



Hình 17: Blog Detail UI

### User Profile (Tạm Thời) (Route: /user/[userId]/[Route~])

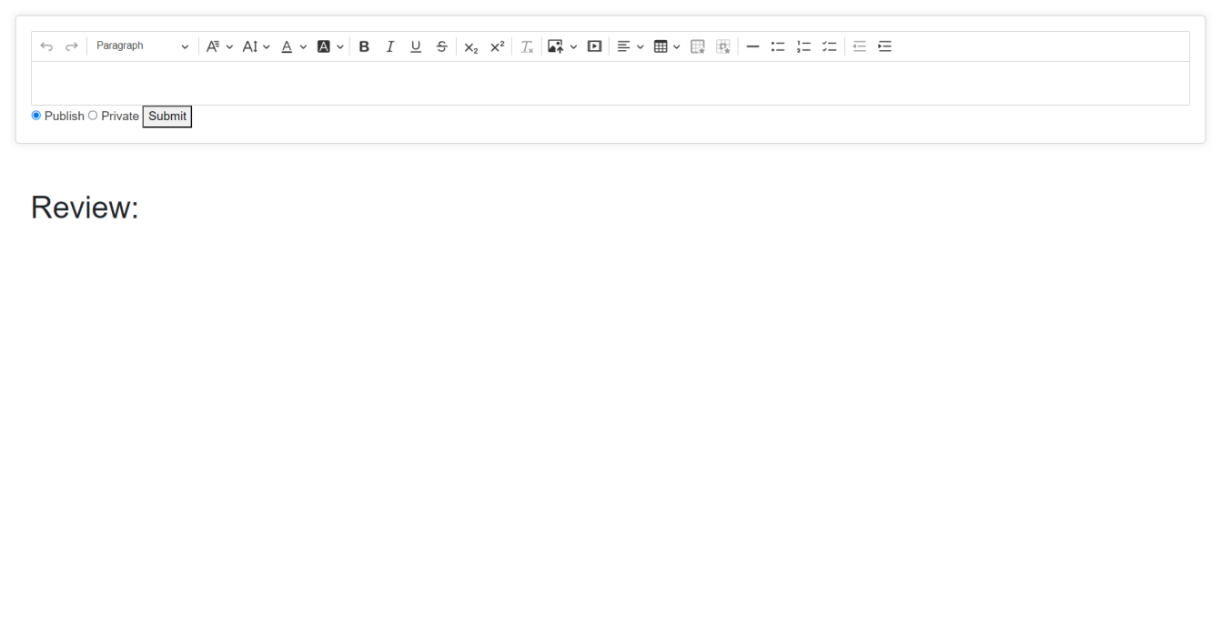
Route này dùng để người dùng quản lý trang cá nhân của bản thân bao gồm: các blogs, các posts, comment,… cùng với đó là các chức năng khác của một user có trong hệ thống được tương tác.



Hình 18: User Profile UI

### Post Create (Tạm Thời) (Route: blog/[…]/post/create)

Route dùng để tạo Post trong 1 blog cụ thể, những gì người dùng tạo ra trong frame sẽ được render ra đúng như những gì người dùng thấy (What you see is what you get).



Hình 19: Post Create UI

# TỔNG KẾT

Dự án Blogpots Web sử dụng ASP.NET Core API cho phép xây dựng một hệ thống quản lý bài viết blog với các chức năng CRUD (Create, Read, Update, Delete) thông qua API. Phần backend được triển khai bằng ASP.NET Core API để quản lý dữ liệu từ cơ sở dữ liệu (PostgreSQL) thông qua EFCore, trong khi phần frontend sử dụng các công nghệ HTML, CSS, SASS và Javascript hoặc các framework hiện đại (Vue JS) để tạo ra giao diện người dùng thân thiện.

Sự tích hợp giữa API và giao diện người dùng đảm bảo tính tương tác cao, mang lại trải nghiệm mượt mà cho người dùng khi truy cập và quản lý nội dung trên web.

Link github: https://github.com/So-Much/ASPdotNet

# TÀI LIỆU THAM KHẢO

Tiếng Việt

…

Tiếng Anh

https://learn.microsoft.com/fr-fr/aspnet/core/?view=aspnetcore-8.0&WT.mc\_id=dotnet-35129-website.