SAU KHI HOÀN TẤT BACKEND TA SẼ CHUYỂN TỚI FE

Phải tách 2 solution ra tuy nhiên lúc code thì chưa cần tách

Step 1: CẤU HÌNH SET UP

B1: Click vào solution add thêm project mới chọn razor page đặt tên theo đề

B2 :Tạo folder mới trong trang Pages có sẵn tên là FootballPlayerPages

B3 : Click vào project add preference Bos

B4: add razor page trong folder tạo ở trên

A screenshot of a computer

Description automatically generated

Sau đó nó sẽ tự generate ra UI

CONFIG LẠI STARTUP PROJECT ĐỂ NÓ RUN BE VÀ FE CÙNG LÚC

A screenshot of a project

Description automatically generated

Step 2: code login

Tạo 1 folderDTO

Tạo 2 class Copy LoginRequestDTO và LoginResponseDTO từ BE qua

@page

@model IndexModel

@{

ViewData["Title"] = "Login Page";

}

<h1 class="display-4">Login Page</h1>

<div>

<**form** method="post">

<div class="form-group">

<label for="email">Email</label>

<**input** **type**="email" **asp-for**="LoginRequestDTO.Email" class="form-control" id="email" placeholder="Enter email" />

<**span** **asp-validation-for**="LoginRequestDTO.Email" class="text-danger"></**span**>

</div>

<div class="form-group">

<label for="password">Password</label>

<**input** **type**="password" **asp-for**="LoginRequestDTO.Password" class="form-control" id="password" placeholder="Password">

<**span** **asp-validation-for**="LoginRequestDTO.Password" class="text-danger"></**span**>

</div>

<button type="submit" class="btn btn-primary">Login</button>

</**form**>

@if (Model.Message != null)

{

<p class="text-danger">@Model.Message</p>

}

</div>

Bên file .cs

Copy đường dẫn trong swagger Port ngẫu nhiên nhớ qua swagger copy

public class IndexModel : PageModel

{

// Đảm bảo LoginRequestDTO được khai báo đúng

[BindProperty]

public LoginRequestDTO LoginRequestDTO { get; set; }

public string Message { get; set; } // Để hiển thị thông báo lỗi nếu có

// Hàm OnPostAsync để xử lý đăng nhập

public async Task<IActionResult> OnPostAsync()

{

try

{

// Tạo HttpClient để gửi yêu cầu POST

using (var httpClient = new HttpClient())

{

var response = await httpClient.PostAsJsonAsync("http://localhost:5098/api/PremierLeagueAccounts/Login", LoginRequestDTO);

if (response.IsSuccessStatusCode)

{

// Đọc phản hồi từ API

var result = await response.Content.ReadFromJsonAsync<LoginResponseDTO>();

// Lưu Token, Role và AccountId vào Session

HttpContext.Session.SetString("Token", result.Token);

HttpContext.Session.SetString("Role", result.Role);

HttpContext.Session.SetString("AccountId", result.AccountId);

// Chuyển hướng đến trang Index của FootballPlayerPages sau khi đăng nhập thành công

return RedirectToPage("/FootballPlayerPages/Index");

}

else

{

var errorContent = await response.Content.ReadAsStringAsync();

if (errorContent.Length != 0)

{

throw new Exception(errorContent);

}

else

{

throw new Exception("An error occurred during login.");

}

}

}

}

catch (Exception e)

{

// Hiển thị thông báo lỗi nếu có

Message = e.Message;

return Page();

}

}

}

Config thêm trong program.cs của frontEnd

builder.Services.AddSession();

Sau đó sử dụng

app.UseSession();

A computer screen shot of a program

Description automatically generated