

Uniwersytet Bielsko-Bialski

LABORATORIUM

Programowanie dla Internetu w technologii ASP.NET

Sprawozdanie nr 9

Wprowadzenie styli sb-admin-2/Panel administratora

Wprowadzenie

W naszym projekcie postanowiliśmy wzbogacić wygląd i funkcjonalność aplikacji poprzez dodanie pakietu styli SB Admin 2. Ten popularny zestaw stylów oparty na Bootstrapie pozwoli nam stworzyć nowoczesny interfejs użytkownika, a także zaimplementować prosty panel administracyjny dostępny tylko dla użytkowników o roli administratora.

Aby rozpocząć, pobraliśmy pakiet styli SB Admin 2 i umieściliśmy go w naszym projekcie, zazwyczaj w folderze **wwwroot** lub **Areas**. Następnie zaktualizowaliśmy nasz plik layoutu (**_Layout.cshtml**), dodając odnośniki do plików CSS z pakietu SB Admin 2.

Następnie do wspólnej części naszego programu czyli pliku **_Layout.cshtml** w repozytorium **Shared** musimy dodać odnośniki do naszych plików

W moim projekcie wygląda to następująco

```
<link
  href="~/sbadmin/vendor/bootstrap/css/bootstrap.min.css"
  rel="stylesheet"
/>
<link href="~/sbadmin/css/sb-admin-2.min.css" rel="stylesheet" />
<link
  href="~/sbadmin/vendor/fontawesome-free/css/all.min.css"
  rel="stylesheet"
  type="text/css"
/>
```

Po dodaniu tych linijek, wszystkie nasze podstrony będą teraz prioretyzować tagi napisane w podanych plikach css, teraz wystarczy że zaaplikujemy ja na naszej stronie.

Mój projekt przed wykorzystaniem styli sb-admin-2

TripAppHomeManage TripsClientsReservations

Log out

Trip List

Destination	Start Date	End Date	Price	
Australia	03.10.2024	24.10.2024	3 500,05 zł	Create Reservation
Egypt	10.04.2024	17.04.2024	1 000,00 zł	Create Reservation
Japan	03.06.2024	13.06.2024	2 500,00 zł	Create Reservation
Australia	03.10.2024	24.10.2024	3 000,00 zł	Create Reservation
France	03.08.2024	10.08.2024	1 800,01 zł	Create Reservation
Australia	03.10.2024	24.10.2024	3 000,00 zł	Create Reservation
Egypt	10.04.2024	17.04.2024	1 000,00 zł	Create Reservation
Japan	03.06.2024	13.06.2024	2 500,00 zł	Create Reservation
Australia	03.10.2024	24.10.2024	3 000,00 zł	Create Reservation
Japan	20.02.2025	27.02.2025	1 000,00 zł	Create Reservation
Egypt	26.04.2024	07.05.2024	2 500,05 zł	Create Reservation

© 2024 - TripApp

Po zastosowaniu nowych stylów

TripAppHomeManage TripsClientsReservations

Log out

Trip List

Available Trips

Destination	Start Date	End Date	Price	
Australia	03.10.2024	24.10.2024	3 500,05 zł	Create Reservation
Egypt	10.04.2024	17.04.2024	1 000,00 zł	Create Reservation
Japan	03.06.2024	13.06.2024	2 500,00 zł	Create Reservation
Australia	03.10.2024	24.10.2024	3 000,00 zł	Create Reservation
France	03.08.2024	10.08.2024	1 800,01 zł	Create Reservation
Australia	03.10.2024	24.10.2024	3 000,00 zł	Create Reservation
Egypt	10.04.2024	17.04.2024	1 000,00 zł	Create Reservation
Japan	03.06.2024	13.06.2024	2 500,00 zł	Create Reservation
Australia	03.10.2024	24.10.2024	3 000,00 zł	Create Reservation
Japan	20.02.2025	27.02.2025	1 000,00 zł	Create Reservation
Egypt	26.04.2024	07.05.2024	2 500,05 zł	Create Reservation

Dodatkowo wykorzystując tę bibliotekę utworzymy prosty panel administratora dostępny tylko przez użytkownika o takiej właśnie roli

Zacniemy klasycznie od utworzenia kontrolera

```
using Microsoft.AspNetCore.Authorization;
using Microsoft.AspNetCore.Identity;
using Microsoft.AspNetCore.Mvc;
using System.Threading.Tasks;
using TripApp.Services;
```

```

namespace TripApp.Controllers
{
    [Authorize(Roles = "Admin")]
    public class AdminController : Controller
    {
        private readonly IAdminService _adminService;

        public AdminController(IAdminService adminService)
        {
            _adminService = adminService;
        }

        public async Task<IActionResult> Index()
        {
            ViewData["Roles"] = await _adminService.GetAllRolesAsync();
            var users = await _adminService.GetAllUsersAsync();
            return View(users);
        }

        [HttpPost]
        public async Task<IActionResult> UpdateUserRole(string userId, string
newRole)
        {
            if (string.IsNullOrEmpty(userId) || string.IsNullOrEmpty(newRole))
            {
                return RedirectToAction("Index");
            }

            await _adminService.UpdateUserRoleAsync(userId, newRole);
            return RedirectToAction("Index");
        }
    }
}

```

A także serwisu który go obsługuje

```

using Microsoft.AspNetCore.Identity;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using Microsoft.EntityFrameworkCore;

namespace TripApp.Services
{
    public class AdminService : IAdminService
    {
        private readonly UserManager<IdentityUser> _userManager;
        private readonly RoleManager<IdentityRole> _roleManager;

        public AdminService(UserManager<IdentityUser> userManager,
RoleManager<IdentityRole> roleManager)

```

```

    {
        _userManager = userManager;
        _roleManager = roleManager;
    }

    public async Task<List<IdentityUser>> GetAllUsersAsync()
    {
        return await _userManager.Users.ToListAsync();
    }

    public async Task<List<string>> GetAllRolesAsync()
    {
        return await _roleManager.Roles.Select(r => r.Name).ToListAsync();
    }

    public async Task UpdateUserRoleAsync(string userId, string newRole)
    {
        var user = await _userManager.FindByIdAsync(userId);
        if (user != null)
        {
            var currentRoles = await _userManager.GetRolesAsync(user);
            await _userManager.RemoveFromRolesAsync(user, currentRoles);
            await _userManager.AddToRoleAsync(user, newRole);
        }
    }
}

```

Z racji że korzystamy z tabel wygenerowanych przez ASP.NET Core nie mamy struktury tych tabel w naszych modelach więc nie stworzymy tutaj repozytoriów, jednak możemy wykorzystać wbudowane w tę bibliotekę klasy **UserManager** i **RoleManager**, które pozwolą nam na zacytanie oraz modyfikację odpowiednich danych

Ostatnim już krokiem jest utworzenie pliku **index.cshtml** dla naszego panelu

Admin Panel

User Management				
ID	Username	Email	Roles	Actions
9ef8d4e4-14d0-44f0-9e6f-f0404a757c06	admin@admin.com	admin@admin.com	Admin	<div><div>Admin</div><div>Update Role</div></div>
9f027f50-212a-4b77-9ac9-f81d4e12d892	igor.gawlowicz@gmail.com	igor.gawlowicz@gmail.com	Admin	<div><div>Admin</div><div>Admin</div><div>Employee</div><div>Client</div></div>

Widzimy tutaj listę wszystkich użytkowników w bazie a także ich rolę oraz możliwość jej zmiany na jedną z istniejących w bazie.

Podsumowanie

Dodanie pakietu stylów SB Admin 2 do naszej aplikacji było krokiem w kierunku poprawy wyglądu i funkcjonalności. Dzięki nowym stylom nasza aplikacja zyskała nowoczesny interfejs użytkownika, co przyczyniło się do lepszej czytelności i atrakcyjności dla użytkowników.

Dodatkowo, wykorzystując SB Admin 2, udało nam się również stworzyć prosty panel administracyjny, dostępny tylko dla administratorów. Panel ten umożliwia zarządzanie użytkownikami, w tym zmianę ich ról, co może być przydatne w przypadku aplikacji wieloosobowej.

Wprowadzenie tych zmian pozwoliło nam nie tylko poprawić wygląd aplikacji, ale także zwiększyć jej funkcjonalność i użyteczność dla użytkowników.