

Міністерство освіти і науки України  
Національний університет “Львівська політехніка”  
Інститут прикладної математики та фундаментальних наук  
Кафедра прикладної математики

ЗВІТ  
про виконання лабораторної роботи №3  
з курсу «Серверне Web-програмування»  
Варіант – 3

**Виконав:**  
Студент групи КНІК-11  
Гевич Олексій  
**Прийняв:**  
Ментинський С. М.

**Тема:**

Авторизація та автентифікація користувачів на основі ASP.Net Core Identity

**Завдання:**

До проекту, створеного у ході виконання лабораторних робіт №№ 1-2

додати наступний функціонал:

- 1) Реалізувати можливість входу в систему та самостійної реєстрації користувачів на основі індивідуальних аккаунтів, що зберігають дані в БД проєкту;
- 2) Розмежувати доступ до ресурсів проєкту так, щоб вносити чи редагувати дані могли лише авторизовані користувачі;
- 3) Додати підтримку ролей користувачів та створити функціонал адмін панелі для керування списком зареєстрованих користувачів, доступний лише для користувачів з роллю адміністратора.

**Результат виконання роботи:**

- 1) Фрагменти коду, який відповідає за налаштування автентифікації користувачів:

```
using Microsoft.AspNetCore.Identity;
using Microsoft.EntityFrameworkCore;
using Lab1.Data;

var builder = WebApplication.CreateBuilder(args);

builder.Services.AddDbContext<ApplicationContext>(options =>
    options.UseSqlServer(builder.Configuration.GetConnectionString("DefaultConnection")));

builder.Services.AddIdentity<IdentityUser, IdentityRole>(options =>
{
    options.Password.RequireDigit = true;
    options.Password.RequireUppercase = false;
    options.Password.RequireNonAlphanumeric = false;
    options.Password.RequiredLength = 6;
})
    .AddEntityFrameworkStores<ApplicationContext>()
    .AddDefaultTokenProviders();

builder.Services.ConfigureApplicationCookie(options =>
{
    options.LoginPath = "/Account/Login";
    options.AccessDeniedPath = "/Account/AccessDenied";
});

builder.Services.AddControllersWithViews();

var app = builder.Build();

app.UseAuthentication();
app.UseAuthorization();
```

```

app.MapControllerRoute(
    name: "default",
    pattern: "{controller=Home}/{action=Index}/{id?}");

app.Run();

```

2) Код контролера та представлень (або код Razor-сторінок), які реалізують функціонал адмін-панелі:

### **AdminController.cs**

```

using Microsoft.AspNetCore.Authorization;
using Microsoft.AspNetCore.Mvc;
using Microsoft.AspNetCore.Identity;

namespace Lab1.Controllers
{
    [Authorize(Roles = "Admin")]
    public class AdminController : Controller
    {
        private readonly UserManager<IdentityUser> _userManager;
        private readonly RoleManager<IdentityRole> _roleManager;

        public AdminController(UserManager<IdentityUser> userManager,
RoleManager<IdentityRole> roleManager)
        {
            _userManager = userManager;
            _roleManager = roleManager;
        }

        public IActionResult Index()
        {
            return View();
        }

        public IActionResult Users()
        {
            var users = _userManager.Users.ToList();
            return View(users);
        }

        public async Task<IActionResult> MakeAdmin(string id)
        {
            var user = await _userManager.FindByIdAsync(id);
            if (user != null)
            {
                await _userManager.AddToRoleAsync(user, "Admin");
            }
            return RedirectToAction("Users");
        }
    }
}

```

### **View: Admin/Index.cshtml**

```

@{
    ViewData["Title"] = "Admin Panel";
}

<h1>❖ Admin Panel</h1>

<ul>

```

```
<li><a asp-action="Users">Manage Users</a></li>
</ul>
```

## View: Admin/Users.cshtml

```
@model IEnumerable<Microsoft.AspNetCore.Identity.IdentityUser>

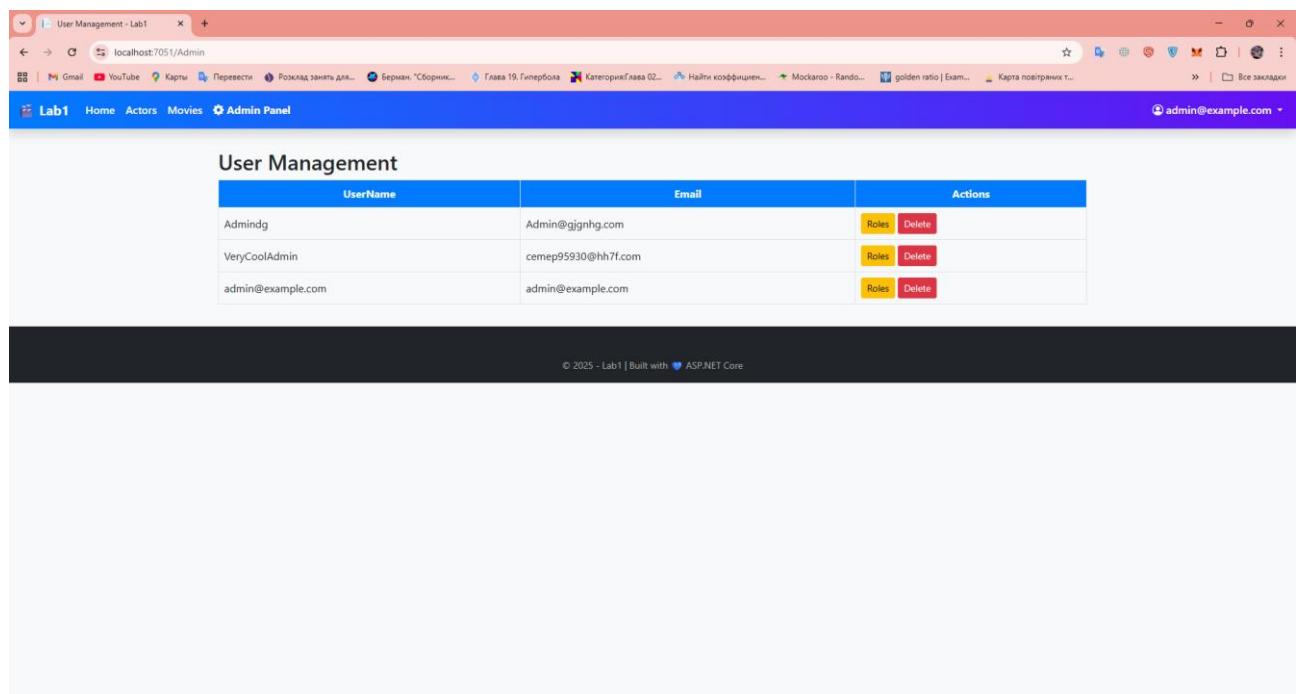
<h2>👤 Users Management</h2>



| Email | Actions |
|-------|---------|
|-------|---------|


```

3) Знімок екрану (screenshot) з виглядом адмін-панелі у браузері:



Репозиторій з проектом:

<https://github.com/OleksiyHevych/Server-web-programing/tree/main/Lab3>