Лабораторна робота №5

Студент: Антропов В. О.
Група: 301-ТК

Мета роботи

Ознайомитися з системою аутентифікації та авторизації в ASP.NET Core. Навчитися створювати користувачів, призначати ролі, видавати JWT-токени, а також обмежувати доступ до API через атрибути [Authorize].

Хід роботи

1. Створено ASP.NET Core Web API-проєкт Lab5IdentityAPI на .NET 6.

2. Додано підтримку ASP.NET Core Identity з класом ApplicationUser.

3. Налаштовано JWT-аутентифікацію з використанням SymmetricSecurityKey.

4. Реалізовано контролер AuthController з методами POST /auth/register та POST /auth/login.

5. Підключено Swagger UI та додано схему авторизації Bearer.

6. Приклад запиту на реєстрацію з неправильним паролем:

Висновки

У ході лабораторної роботи було реалізовано систему реєстрації та логіну користувачів з використанням ASP.NET Core Identity та JWT. Було налаштовано Swagger для зручного тестування та вивчено принципи захисту REST API за допомогою авторизації.

Контрольні запитання

1. 1. Що таке ASP.NET Core Identity?

Це система управління користувачами в ASP.NET, яка дозволяє реєструвати, аутентифікувати та авторизовувати користувачів.

2. 2. Що таке JWT?

JSON Web Token — це компактний безпечний формат для передачі даних між сторонами як об'єкта JSON.

3. 3. Які атрибути використовуються для захисту методів API?

[Authorize], [AllowAnonymous], [Authorize(Roles="Admin")]

4. 4. Для чого використовується SignInManager?

Для аутентифікації користувачів (логіну, перевірки пароля тощо).

5. 5. Що таке Claims?

Claims — це пари ключ-значення, які зберігають інформацію про користувача в токені.

6. 6. Як працює видача токена при логіні?

Після успішного входу генерується JWT-токен, який підписується та повертається користувачу.

7. 7. Чим корисна рольова авторизація?

Дозволяє надавати доступ до частин API лише користувачам з певними ролями.

8. 8. Як додати роль користувачу?

Через UserManager.AddToRoleAsync(user, "RoleName") після створення користувача.