

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ОДЕСЬКА ПОЛІТЕХНІКА»**

**НАВЧАЛЬНО-НАУКОВИЙ
ІНСТИТУТ КОМП'ЮТЕРНИХ СИСТЕМ**

**КАФЕДРА ПРИКЛАДНОЇ МАТЕМАТИКИ ТА ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ**

**МЕТОДИЧНІ ВКАЗІВКИ
до виконання лабораторних робіт
з дисципліни «Сучасні мови програмування»
для здобувачів першого (бакалаврського) рівня вищої освіти
за спеціальністю
124 Системний аналіз**

Одеса – 2025

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ОДЕСЬКА ПОЛІТЕХНІКА»**

**НАВЧАЛЬНО-НАУКОВИЙ
ІНСТИТУТ КОМП'ЮТЕРНИХ СИСТЕМ**

**КАФЕДРА ПРИКЛАДНОЇ МАТЕМАТИКИ ТА ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ**

**МЕТОДИЧНІ ВКАЗІВКИ
до виконання лабораторних робіт
з дисципліни «Сучасні мови програмування»
для здобувачів першого (бакалаврського) рівня вищої освіти
за спеціальністю
124 Системний аналіз**

**Затверджено
на засіданні кафедри прикладної
математики та інформаційних
технологій
Протокол № _ від _____**

Одеса – 2025

Методичні вказівки до виконання лабораторних робіт з дисципліни «Сучасні мови програмування» для здобувачів першого (бакалаврського) рівня вищої освіти за спеціальністю 124 – Системний аналіз (освітньо-професійна програма: «Системний аналіз та наука про дані»)/ Укл.: К. О. Трифонова. – Одеса, 2025. – 16 с.

Укладач: Трифонова К. О., ст. викл.

ЗМІСТ

Вступ	4
Лабораторна робота 1. Веб-застосування	5
Лабораторна робота 2. Контролери та методи дій	7
Лабораторна робота 3. Веб-застосування з доступом до баз даних	8
Лабораторна робота 4. Робота з відношеннями	9
Лабораторна робота 5. Формування шаблонів для існуючих баз даних	11
Лабораторна робота 6. Введення до системи ідентифікації	12
Лабораторна робота 7. Розширені можливості системи ідентифікації	13
Література	15

Вступ

Навчальна дисципліна «Сучасні мови програмування» є необхідною дисципліною для підвищення рівня теоретичних і прикладних знань, що формують фахівця в галузі інформаційних технологій, які сприяють утворенню у здобувачів поглиблених вмінь та навичок проектування та розробки системи управління доступом для веб-застосувань із застосуванням сучасних парадигм програмування, тобто підготовці спеціалістів для створення, впровадження та підтримки професійно-орієнтованих комп'ютерних технологій у професійній діяльності.

Дисципліна «Сучасні мови програмування» відповідає освітньо-професійній програмі, навчальному плану підготовки фахівців першого (бакалаврського) рівня вищої освіти спеціальності 124 – Системний аналіз і є складовою циклу дисциплін професійної підготовки обов'язкової частини навчального плану.

Дисципліну викладають впродовж сьомого семестру першого (бакалаврського) рівня. У процесі навчання передбачено лекції, лабораторні заняття.

Згідно навчального плану передбачено підсумковий контроль у вигляді заліку.

Робочу програму навчальної дисципліни укладено згідно з вимогами кредитно-модульної системи організації навчального процесу. Програма визначає обсяги компетентностей, які повинен опанувати здобувач відповідно до освітньо-професійної програми, алгоритму вивчення навчального матеріалу дисципліни «Сучасні мови програмування», необхідне методичне забезпечення, складові та технологію оцінювання навчальних досягнень.

Предмет навчальної дисципліни «Сучасні мови програмування» – сучасні технології проектування та розробки системи управління доступом для веб-застосувань.

Мета вивчення навчальної дисципліни «Сучасні мови програмування» – забезпечити формування поглибленої системи теоретичних і практичних знань у галузі проектування та розробки системи управління доступом для веб-застосувань, пов'язаною з наданням можливості санкціонованого та попередженням несанкціонованого доступу до даних, для реалізації здатності використання інформаційно-комунікаційних технологій, сучасних методів і моделей інформаційної безпеки та/або кібербезпеки.

Завдання вивчення дисципліни:

- формування теоретичних основ та практичних навичок для проектування та розробки системи автентифікації, перевірки справжності користувача шляхом порівняння за допомогою певної унікальної інформації для веб-застосувань;
- формування теоретичних основ та практичних навичок для проектування та розробки системи авторизації, перевірки та визначення повноважень на виконання певних дій для веб-застосувань;
- формування теоретичних основ та практичних навичок для проектування та розробки системи конфіденційності, гарантування захисту важливих даних, що передаються між клієнтом та сервером або зберігаються для веб-застосувань;
- формування теоретичних основ та практичних навичок для проектування та розробки системи цілісності, гарантування внаслідок неавторизованого втручання не зміненості важливих даних, що передаються між клієнтом та сервером для веб-застосувань.

Стратегічні цілі дисципліни – націлити майбутніх фахівців на творче застосування отриманих знань у подальшій професійній підготовці та їх наступній практичній діяльності.

ЛАБОРАТОРНА РОБОТА 1. ВЕБ-ЗАСТОСУВАННЯ

Мета роботи

1. Закріплення і поглиблення теоретичних знань, отриманих при вивченні розділу «Веб-застосування».
2. Набуття практичних навичок розробки та реалізації веб-застосування з використанням шаблону MVC.
3. Набуття практичних навичок налагодження веб-застосування, в якому реалізовано шаблон MVC.

Теоретичні відомості

Для виконання лабораторної роботи необхідні теоретичні відомості наводяться в конспекті лекцій у розділі «Веб-застосування».

Хід роботи

1. Наведіть послідовність дій для створення застосування ASP.NET Core Empty.
2. Наведіть послідовність дій для додавання контролера для застосування ASP.NET Core Empty.
3. Наведіть приклад коду на C# для файлу Program.cs для налаштування маршрутизації застосування ASP.NET Core MVC.
4. Наведіть послідовність дій для додавання представлення для застосування ASP.NET Core Empty.
5. Наведіть приклад коду на C# методу дії контролера, в якому виконується передача даних з контролера у представлення для застосування ASP.NET Core MVC.
6. Наведіть послідовність дій для додавання моделі для застосування ASP.NET Core Empty.
7. Наведіть приклад коду на C# кількох методів дії в одному контролері для застосування ASP.NET Core MVC.
8. Наведіть приклад коду на C# підключення моделі представленню для застосування ASP.NET Core MVC.
9. Наведіть приклад коду деякого дескриптору з використанням атрибуту asp-action у представленні для застосування ASP.NET Core MVC.
10. Перерахуйте кілька можливих складових файлу імпортування для застосування ASP.NET Core MVC.
11. Перерахуйте необхідні дескриптори для побудови форми для застосування ASP.NET Core MVC.
12. Наведіть приклад коду деякого дескриптору з використанням атрибуту asp-for у представленні для застосування ASP.NET Core MVC.
13. Наведіть приклад коду на C# двох перевантажених методів дії контролера, які обробляють GET та POST запити для застосування ASP.NET Core MVC.
14. Наведіть приклад коду на C# методу дії контролера, для якого функціонує прив'язка моделі для застосування ASP.NET Core MVC.
15. Наведіть приклад порушення перевірки достовірності для дескриптору input представлення для застосування ASP.NET Core MVC.
16. Перерахуйте кілька атрибутів перевірки достовірності для моделі для застосування ASP.NET Core MVC.
17. Наведіть приклад коду на C# методу дії контролера, в якому виконується перевірка наявності проблем з достовірністю даних для застосування ASP.NET Core MVC.

18. Наведіть приклад коду на С# для файлу Program.cs для налаштування дозволу на обслуговування статичних файлів для застосування ASP.NET Core MVC.

19. Наведіть послідовність дій для підключення пакету Bootstrap для застосування ASP.NET Core Empty.

20. Перерахуйте кілька стилів з пакету Bootstrap для дескриптору button представлення для застосування ASP.NET Core MVC.

ЛАБОРАТОРНА РОБОТА 2. КОНТРОЛЕРИ ТА МЕТОДИ ДІЙ

Мета роботи

1. Закріплення і поглиблення теоретичних знань, отриманих при вивченні розділу «Контролери та методи дій».

2. Набуття практичних навичок розробки та реалізації веб-застосування з використанням контролерів та методів дій.

3. Набуття практичних навичок налагодження веб-застосування, в якому реалізовані контролери та методи дій.

Теоретичні відомості

Для виконання лабораторної роботи необхідні теоретичні відомості наводяться в конспекті лекцій у розділі «Контролери та методи дій».

Хід роботи

1. Поясніть механізм реалізації роботи контролера для застосування ASP.NET Core MVC.

2. Перерахуйте кроки, які потрібно виконати для додавання контролера для застосування ASP.NET Core MVC.

3. Наведіть приклад коду на С# контролера, який містить метод дії РОСО для застосування ASP.NET Core MVC.

4. Наведіть приклад коду на С# контролера, який містить метод дії, що використовує деякий метод базового класу Controller для застосування ASP.NET Core MVC.

5. Перерахуйте способи отримання даних контексту для методу дії контролера для застосування ASP.NET Core MVC.

6. Наведіть приклад коду на С# контролера, який містить метод дії, що отримує дані контексту за допомогою властивості класу Controller для застосування ASP.NET Core MVC.

7. Наведіть приклад коду на С# контролера, який містить метод дії, що отримує дані контексту за допомогою параметрів для застосування ASP.NET Core MVC.

8. Наведіть приклад коду на С# контролера, який містить метод дії, що генерує відповідь з використанням об'єкта HttpResponseMessage для застосування ASP.NET Core MVC.

9. Назвіть інтерфейс, який реалізують усі класи результатів методу дії контролера для застосування ASP.NET Core MVC.

10. Наведіть приклад коду на С# контролера, який містить метод дії, що виконує виклик представлення, ім'я якого збігається з ім'ям методу дії для застосування ASP.NET Core MVC.

11. Поясніть правила організації зберігання представлень для застосування ASP.NET Core MVC.

12. Наведіть приклад коду на С# контролера, який містить метод дії, що виконує передачу об'єкта до представлення за допомогою методу View() для застосування ASP.NET Core MVC.

13. Поясніть можливі помилки під час використання методу View() з одним вхідним параметром, який виконує передачу змінної строкового типу даних з методу дії контролера до представлення для застосування ASP.NET Core MVC.

14. Наведіть приклад коду на С# контролера, який містить метод дії, що виконує передачу об'єкта до представлення за допомогою ViewBag для застосування ASP.NET Core MVC.

15. Наведіть приклад коду на С# контролера, який містить метод дії, що виконує тимчасове перенаправлення браузера на певний URL для застосування ASP.NET Core MVC.

16. Наведіть приклад коду на С# контролера, який містить метод дії, що виконує передачу об'єкта в інший метод дії за допомогою TempData для застосування ASP.NET Core MVC.

17. Наведіть приклад коду на С# контролера, який містить метод дії, що повертає об'єкт у форматі JSON для застосування ASP.NET Core MVC.

18. Наведіть приклад коду на С# контролера, який містить метод дії, що повертає файл CSS для застосування ASP.NET Core MVC.

19. Наведіть приклад коду на С# контролера, який містить метод дії, що повертає код HTTP 404, який означає, що ресурс не існує для застосування ASP.NET Core MVC.

20. Наведіть приклад коду на С# контролера, який містить метод дії, що повертає часткове представлення для застосування ASP.NET Core MVC.

ЛАБОРАТОРНА РОБОТА 3. ВЕБ-ЗАСТОСУВАННЯ З ДОСТУПОМ ДО БАЗ ДАНИХ

Мета роботи

1. Закріплення і поглиблення теоретичних знань, отриманих при вивченні розділу «Веб-застосування з доступом до баз даних».

2. Набуття практичних навичок розробки та реалізації веб-застосування з використанням інфраструктури для доступу до баз даних.

3. Набуття практичних навичок налагодження веб-застосування, в якому використана інфраструктура для доступу до баз даних.

Теоретичні відомості

Для виконання лабораторної роботи необхідні теоретичні відомості наводяться в конспекті лекцій у розділі «Веб-застосування з доступом до баз даних».

Хід роботи

1. Наведіть практичний приклад необхідності використання інфраструктури Entity Framework Core для застосування ASP.NET Core MVC.

2. Наведіть кілька способів, які використовуються для встановлення пакетів необхідних для функціонування інфраструктури Entity Framework Core для застосування ASP.NET Core MVC.

3. Наведіть приклад коду на С# класу контексту БД для застосування ASP.NET Core MVC, яке використовує Entity Framework Core.

4. Наведіть приклад коду на С# сутнісного класу для застосування ASP.NET Core MVC, яке використовує Entity Framework Core.

5. Наведіть приклад коду на С# контролера для застосування ASP.NET Core MVC, яке використовує Entity Framework Core.

6. Наведіть приклад постачальника БД для застосування ASP.NET Core MVC, яке використовує Entity Framework Core.

7. Наведіть приклад строчки підключення для конфігурування постачальника БД для застосування ASP.NET Core MVC, яке використовує Entity Framework Core.

8. Наведіть приклад коду на С# для файлу Program.cs конфігурування постачальника БД для застосування ASP.NET Core MVC, яке використовує Entity Framework Core.

9. Поясніть спосіб конфігурування ведення журналу Entity Framework Core для застосування ASP.NET Core MVC.

10. Наведіть команди PowerShell для створення нової міграції та застосування міграції до БД для застосування ASP.NET Core MVC, яке використовує Entity Framework Core.
11. Наведіть команду SQL для заповнення таблиці БД значеннями для застосування ASP.NET Core MVC, яке використовує Entity Framework Core.
12. Наведіть команду PowerShell для запуску застосування ASP.NET Core MVC, яке використовує Entity Framework Core.
13. Перерахуйте складові, необхідні для організації функціонування патерну «Сховище» для застосування ASP.NET Core MVC, яке використовує Entity Framework Core.
14. Наведіть приклад коду на C# реалізації патерну «Сховище» для застосування ASP.NET Core MVC, яке використовує Entity Framework Core.
15. Наведіть практичний приклад необхідності організації вирішення задачі, для якої дані вилучаються з таблиці БД, фільтрація даних та відображення користувачеві виконується в застосуванні ASP.NET Core MVC, яке використовує Entity Framework Core.
16. Наведіть практичний приклад необхідності організації вирішення задачі, для якої відфільтровані дані вилучаються з таблиці БД, відображення користувачеві виконується в застосуванні ASP.NET Core MVC, яке використовує Entity Framework Core.
17. Наведіть інструмент інфраструктури, який допомагає проаналізувати запити, що виконуються, для застосування ASP.NET Core MVC, яке використовує Entity Framework Core.
18. Наведіть механізм організації приховування операцій над даними, за допомогою реалізації патерну «Сховище», для застосування ASP.NET Core MVC, яке використовує Entity Framework Core.
19. Наведіть приклад коду на C# інтерфейсу патерна «Сховище», який містить основні операції над даними для застосування ASP.NET Core MVC, яке використовує Entity Framework Core.
20. Наведіть приклад коду на C# методу дії контролера, який використовує патерн «Сховище» для застосування ASP.NET Core MVC, яке використовує Entity Framework Core.

ЛАБОРАТОРНА РОБОТА 4. РОБОТА З ВІДНОШЕННЯМИ

Мета роботи

1. Закріплення і поглиблення теоретичних знань, отриманих при вивченні розділу «Робота з відношеннями».
2. Набуття практичних навичок розробки та реалізації веб-застосування з використанням роботи з відношеннями.
3. Набуття практичних навичок налагодження веб-застосування, в якому реалізована робота з відношеннями.

Теоретичні відомості

Для виконання лабораторної роботи необхідні теоретичні відомості наводяться в конспекті лекцій у розділі «Робота з відношеннями».

Хід роботи

1. Наведіть приклад коду на C# класу контексту БД, організованого таким чином, що дозволяє отримати доступ за допомогою навігаційної властивості до даних двох зв'язаних таблиць для застосування ASP.NET Core MVC, яке використовує Entity Framework Core.
2. Наведіть приклад коду на C# класу контексту БД, організованого таким чином, що дозволяє отримати доступ безпосередньо до даних двох зв'язаних таблиць для застосування ASP.NET Core MVC, яке використовує Entity Framework Core.

3. Поясніть причину необхідності створення міграції при реалізації механізму підвищення зв'язаних даних у класі контексту БД для застосування ASP.NET Core MVC, яке використовує Entity Framework Core.

4. Поясніть причину необхідності реалізації патерну «Сховище» після виконання механізму підвищення зв'язаних даних у класі контексту БД для застосування ASP.NET Core MVC, яке використовує Entity Framework Core.

5. Наведіть приклад коду на C# інтерфейсу, для реалізації узагальненого патерну «Сховище» для застосування ASP.NET Core MVC, яке використовує Entity Framework Core.

6. Наведіть приклад коду на C# класу, для реалізації узагальненого патерну «Сховище» для застосування ASP.NET Core MVC, яке використовує Entity Framework Core.

7. Наведіть приклад коду на C# класу, для реалізації узагальненого патерну «Сховище», який містить віртуальні методи, для застосування ASP.NET Core MVC, яке використовує Entity Framework Core.

8. Наведіть приклад коду на C# класу, для реалізації узагальненого патерну «Сховище», на який накладається обмеження на базовий клас, для застосування ASP.NET Core MVC, яке використовує Entity Framework Core.

9. Наведіть приклад коду на C# для файлу Program.cs конфігурування конкретного типу, при реалізації узагальненого патерну «Сховище», для створення служби впровадження залежностей для застосування ASP.NET Core MVC, яке використовує Entity Framework Core.

10. Наведіть приклад коду на C# двох класів, для яких організовано обов'язкове укомплектування відношення для застосування ASP.NET Core MVC, яке використовує Entity Framework Core.

11. Поясніть причину відсутності необхідності створення міграції для організації укомплектування відношення між даними для застосування ASP.NET Core MVC, яке використовує Entity Framework Core.

12. Наведіть приклад коду на C# методу класу патерну «Сховище», який виконує повернення всіх строк таблиці БД з відповідними зв'язаними даними, для яких організовано укомплектування відношення, для застосування ASP.NET Core MVC, яке використовує Entity Framework Core.

13. Наведіть приклад коду на C# методу класу патерну «Сховище», з використанням явного завантаження, який виконує повернення всіх строк таблиці БД з відповідними зв'язаними даними, для яких організовано укомплектування відношення, для застосування ASP.NET Core MVC, яке використовує Entity Framework Core.

14. Наведіть приклад коду на C# методу класу патерну «Сховище», з використанням явного завантаження, який виконує повернення всіх строк таблиці БД з відповідними обмеженими зв'язаними даними, для яких організовано укомплектування відношення, для застосування ASP.NET Core MVC, яке використовує Entity Framework Core.

15. Наведіть приклад коду на C# методу класу патерну «Сховище», з використанням засобу виправлення, який виконує повернення всіх строк таблиці БД з відповідними обмеженими зв'язаними даними, для яких організовано укомплектування відношення, для застосування ASP.NET Core MVC, яке використовує Entity Framework Core.

16. Наведіть рекомендації використання засобу виправлення, який виконує повернення всіх строк таблиці БД з відповідними зв'язаними даними, для яких організовано укомплектування відношення, для застосування ASP.NET Core MVC, яке використовує Entity Framework Core.

17. Наведіть приклад коду представлення, що виконує відображення даних, для яких організовано укомплектування відношення для застосування ASP.NET Core MVC, яке використовує Entity Framework Core.

18. Наведіть приклад коду представлення, що виконує відображення даних, для яких організовано укомплектування відношення з використанням часткового представлення для застосування ASP.NET Core MVC, яке використовує Entity Framework Core.

19. Поясніть послідовність дій необхідних для реалізації створення даних для моделей даних, між якими організовано укомплектування відношення для застосування ASP.NET Core MVC, яке використовує Entity Framework Core.

20. Поясніть послідовність дій необхідних для реалізації зміни відношень між даними для моделей даних, між якими організовано укомплектування відношення для застосування ASP.NET Core MVC, яке використовує Entity Framework Core.

ЛАБОРАТОРНА РОБОТА 5. ФОРМУВАННЯ ШАБЛОНІВ ДЛЯ ІСНУЮЧИХ БАЗ ДАНИХ

Мета роботи

1. Закріплення і поглиблення теоретичних знань, отриманих при вивченні розділу «Формування шаблонів для існуючих баз даних».

2. Набуття практичних навичок розробки та реалізації веб-застосування з використанням формування шаблонів для існуючих баз даних.

3. Набуття практичних навичок налагодження веб-застосування, в якому реалізовано формування шаблонів для існуючих баз даних.

Теоретичні відомості

Для виконання лабораторної роботи необхідні теоретичні відомості наводяться в конспекті лекцій у розділі «Формування шаблонів для існуючих баз даних».

Хід роботи

1. Поясніть умови, за яких необхідно використовувати підхід «спочатку код» для застосування ASP.NET Core MVC, яке використовує Entity Framework Core.

2. Поясніть умови, за яких необхідно використовувати підхід «спочатку БД» для застосування ASP.NET Core MVC, яке використовує Entity Framework Core.

3. Наведіть команду PowerShell для виконання засобу формування шаблонів для наявної БД для застосування ASP.NET Core MVC, яке використовує Entity Framework Core.

4. Поясніть результат роботи засобу формування шаблонів для існуючої БД для застосування ASP.NET Core MVC, яке використовує Entity Framework Core.

5. Назвіть метод, у якому засіб формування шаблонів для існуючої БД визначає строчку підключення для конфігурування постачальника БД для застосування ASP.NET Core MVC, яке використовує Entity Framework Core.

6. Наведіть приклад строчки підключення для конфігурування постачальника БД, згенерованої засобом формування шаблонів для існуючої БД, для застосування ASP.NET Core MVC, яке використовує Entity Framework Core.

7. Наведіть приклад коду на C# для файлу Program.cs конфігурування постачальника БД, після застосування засобу формування шаблонів для існуючої БД, для застосування ASP.NET Core MVC, яке використовує Entity Framework Core.

8. Наведіть практичний приклад необхідності реагування на зміни у БД після застосування засобу формування шаблонів для існуючої БД для застосування ASP.NET Core MVC, яке використовує Entity Framework Core.

9. Наведіть практичний приклад можливих модифікацій БД, які коректно застосовуються засобом формування шаблонів для існуючої БД для застосування ASP.NET Core MVC, яке використовує Entity Framework Core.

10. Наведіть практичний приклад можливих модифікацій БД, які некоректно застосовуються засобом формування шаблонів для існуючої БД для застосування ASP.NET Core MVC, яке використовує Entity Framework Core.

11. Наведіть команду PowerShell для виконання засобу формування шаблонів для існуючої БД після оновлення БД для застосування ASP.NET Core MVC, яке використовує Entity Framework Core.

12. Поясніть зміни класу контексту в результаті застосування засобу формування шаблонів для існуючої БД після додавання даних таблиці в БД для застосування ASP.NET Core MVC, яке використовує Entity Framework Core.

13. Поясніть зміни класу контексту в результаті застосування засобу формування шаблонів для існуючої БД після додавання нової таблиці в БД для застосування ASP.NET Core MVC, яке використовує Entity Framework Core.

14. Поясніть зміни класу контексту в результаті застосування засобу формування шаблонів для існуючої БД після додавання необов'язкового відношення «один до багатьох» між двома таблицями в БД для застосування ASP.NET Core MVC, яке використовує Entity Framework Core.

15. Поясніть зміни класу контексту в результаті застосування засобу формування шаблонів для існуючої БД після додавання обов'язкового відношення «один до багатьох» між двома таблицями в БД для застосування ASP.NET Core MVC, яке використовує Entity Framework Core.

16. Поясніть зміни класу контексту в результаті застосування засобу формування шаблонів для існуючої БД після додавання укомплектованого відношення «один до багатьох» між двома таблицями в БД для застосування ASP.NET Core MVC, яке використовує Entity Framework Core.

17. Поясніть зміни класу контексту в результаті застосування засобу формування шаблонів для існуючої БД після додавання обов'язкового відношення «один до одного» між двома таблицями в БД для застосування ASP.NET Core MVC, яке використовує Entity Framework Core.

18. Поясніть зміни класу контексту в результаті застосування засобу формування шаблонів для існуючої БД після додавання обов'язкового відношення «багато до багатьох» між двома таблицями в БД для застосування ASP.NET Core MVC, яке використовує Entity Framework Core.

19. Наведіть складові проекту, які необхідно модифікувати в результаті застосування засобу формування шаблонів для існуючої БД після оновлення БД для застосування ASP.NET Core MVC, яке використовує Entity Framework Core.

20. Наведіть спосіб додавання можливості сталості моделі даних в результаті застосування засобу формування шаблонів для існуючої БД після оновлення БД для застосування ASP.NET Core MVC, яке використовує Entity Framework Core.

ЛАБОРАТОРНА РОБОТА 6. ВВЕДЕННЯ ДО СИСТЕМИ ІДЕНТИФІКАЦІЇ

Мета роботи

1. Закріплення і поглиблення теоретичних знань, отриманих при вивченні розділу «Введення до системи ідентифікації».

2. Набуття практичних навичок розробки та реалізації веб-застосування з використанням системи ідентифікації.

3. Набуття практичних навичок налагодження веб-застосування, в якому реалізовано систему ідентифікації.

Теоретичні відомості

Для виконання лабораторної роботи необхідні теоретичні відомості наводяться в конспекті лекцій у розділі «Введення до системи ідентифікації».

Хід роботи

1. Поясніть порядок підключення ASP.NET Core Identity до нового проєкту ASP.NET Core MVC.

2. Опишіть послідовність створення власного класу користувача на основі IdentityUser.

3. Поясніть, як реалізується клас контексту бази даних для роботи з ASP.NET Core Identity.
4. Опишіть налаштування підключення до SQL Server LocalDB для збереження даних користувачів.
5. Поясніть процес реєстрації контексту бази даних у контейнері служб застосунку.
6. Охарактеризуйте порядок додавання та конфігурації служб Identity у методі ConfigureServices.
7. Поясніть призначення виклику методу UseAuthentication у конвеєрі обробки запитів.
8. Опишіть кроки створення та застосування міграцій Entity Framework Core для бази даних Identity.
9. Поясніть, як перевірити коректність створення таблиць Identity у базі даних.
10. Охарактеризуйте практичне використання служби UserManager для отримання списку користувачів.
11. Поясніть порядок створення нового користувача за допомогою API ASP.NET Core Identity.
12. Опишіть обробку результатів виконання операцій створення користувача через IdentityResult.
13. Поясніть, як реалізується валідація введених даних під час створення облікового запису.
14. Охарактеризуйте налаштування політики перевірки паролів у конфігурації застосунку.
15. Поясніть порядок реалізації власного валідатора паролів у системі Identity.
16. Опишіть реєстрацію спеціального валідатора паролів у контейнері служб.
17. Поясніть, як налаштовується перевірка унікальності адрес електронної пошти користувачів.
18. Охарактеризуйте реалізацію власного валідатора користувачів у ASP.NET Core Identity.
19. Поясніть порядок видалення облікового запису користувача за допомогою UserManager.
20. Опишіть практичні кроки редагування даних користувача з використанням засобів ASP.NET Core Identity.

ЛАБОРАТОРНА РОБОТА 7. Розширені можливості системи ідентифікації

Мета роботи

1. Закріплення і поглиблення теоретичних знань, отриманих при вивченні розділу «Розширені можливості системи ідентифікації».
2. Набуття практичних навичок розробки та реалізації веб-застосунку з використанням розширених можливостей системи ідентифікації.
3. Набуття практичних навичок налагодження веб-застосунку, в якому реалізовано розширені можливості системи ідентифікації.

Теоретичні відомості

Для виконання лабораторної роботи необхідні теоретичні відомості наводяться в конспекті лекцій у розділі «Розширені можливості системи ідентифікації».

Хід роботи

1. Поясніть порядок додавання нових властивостей до моделі користувача в застосунку ASP.NET Core Identity.
2. Опишіть послідовність дій для оновлення схеми бази даних після розширення класу користувача.
3. Поясніть, як забезпечується збереження наявних даних користувачів під час виконання міграцій.

4. Опишіть практичне використання механізму керування користувачами для оновлення їхніх додаткових властивостей.
5. Поясніть, яким чином отримуються дані поточного автентифікованого користувача в контролері.
6. Охарактеризуйте дії, необхідні для відображення спеціальних властивостей користувача у представленні.
7. Поясніть процес перевірки коректності введених користувачем даних перед їх збереженням.
8. Опишіть порядок збереження змінених властивостей користувача в базі даних.
9. Поясніть, чому при прямій зміні властивостей користувача необхідно явно ініціювати оновлення даних.
10. Охарактеризуйте типові помилки, що можуть виникати під час розширення моделі користувача.
11. Поясніть практичне використання заявок для контролю доступу до методів дій контролера.
12. Опишіть порядок створення та реєстрації власного джерела заявок у застосунку.
13. Поясніть, як перевірити наявність певної заявки у поточного користувача.
14. Охарактеризуйте послідовність налаштування політики авторизації на основі заявок.
15. Поясніть застосування політики авторизації до контролера або окремого методу дії.
16. Опишіть практичний механізм створення власної вимоги авторизації.
17. Поясніть, як реалізується обробник вимоги авторизації та його реєстрація в контейнері служб.
18. Охарактеризуйте порядок авторизації доступу користувача до окремого ресурсу застосунку.
19. Поясніть дії, які виконуються у разі відмови в авторизації доступу до ресурсу.
20. Опишіть практичні кроки інтеграції сторонньої аутентифікації в застосунок ASP.NET Core Identity.

ЛІТЕРАТУРА

Базова

1. Freeman A. Pro Entity Framework Core 2 for ASP.NET Core MVC. Apress, 2018. 652 p.

2. Naylor L. ASP.NET MVC with Entity Framework and CSS. Apress, 2016. 628 p.
3. Gorman B. Practical Entity Framework Core 6: Database Access for Enterprise Applications. Apress, 2022. 812 p.
4. Smith J. Entity Framework Core in Action. Manning, 2021. 626 p.
5. Freeman A. Pro ASP.NET Core Identity: Under the Hood with Authentication and Authorization in ASP.NET Core 5 and 6 Applications. Apress, 2021. 746 p.

Допоміжна

6. Jürgen W. HTML. The Comprehensive Guide. Rheinwerk Computing, 2023. 817 p.
7. Meyer E. CSS. The Definitive Guide. O'Reilly, 2023. 1129 p.
8. Flanagan D. JavaScript. The Definitive Guide. O'Reilly, 2020. 707 p.
9. Schildt H. C# 4.0 The Complete Reference. McGraw-Hill, 2010. 976 p.
10. Freeman A. Pro ASP.NET Core 6 Develop Cloud-Ready Web Applications Using MVC, Blazor, and Razor Pages. Apress, 2022. 1267 p.

Інтернет ресурси

11. freeCodeCamp.org. ASP.NET Core MVC Course for Beginners (.NET 9). URL: <https://www.youtube.com/watch?v=RWXKysImabs>
12. Frank Liu. Full Course – Learn ASP.NET Core MVC in .NET 8 | CRUD Operations | EntityFramework | MVC Tutorial. URL: https://www.youtube.com/watch?v=BzlPrVB_DwA
13. Frank Liu. ASP.NET CORE Authentication & Authorization Overview | Security & Identity Series. URL: https://www.youtube.com/watch?v=p-00eBUpGa4&list=PLgRlicSxjeMOxypAEL2XqIc2m_gPmoVN
14. Teddy Smith. ASP.NET Core MVC 2022. 13. Identity Framework. URL: <https://www.youtube.com/watch?v=THKTQKrg4a4>
15. Milan Jovanović. Authentication made easy with ASP.NET Core Identity in .NET 8. URL: <https://www.youtube.com/watch?v=S0RSsHKiD6Y>

Методичні вказівки до виконання лабораторних робіт з дисципліни «Сучасні мови програмування» для здобувачів першого (бакалаврського) рівня вищої освіти за спеціальністю 124 Системний аналіз / Укл.: К. О. Трифонова. – Одеса, 2025. – 16 с.

Укладач: Трифонова К. О., ст. викл.

Підписано до друку _____. Формат 60х84/16. Папір газетний. Друк офсетний. 0,87 ум. друк. арк. 0,94 обл. - вид. арк.
Тираж 100 пр. Зам. №

Національний університет «Одеська політехніка»
65044, Одеса, пр. Шевченка, 1