

## **Завдання для лабораторних робіт з курсу «Серверне Web-програмування»**

Загальні положення

ЗАВДАННЯ ДЛЯ ЛАБОРАТОРНИХ РОБІТ З КУРСУ «СЕРВЕРНЕ WEB-ПРОГРАМУВАННЯ» .....	1
ЗАГАЛЬНІ ПОЛОЖЕННЯ .....	2
ЛАБОРАТОРНА РОБОТА № 1. “РОЗРОБКА WEB-ДОДАТКУ ASP.NET CORE MVC” .....	3
ЛАБОРАТОРНА РОБОТА № 2. “ВИКОРИСТАННЯ СЕСІЙ ДЛЯ ТИМЧАСОВОГО ЗБЕРІГАННЯ ДАНИХ В ASP.NET CORE ДОДАТКАХ” .....	4
ЛАБОРАТОРНА РОБОТА № 3. “АВТОРИЗАЦІЯ ТА АВТЕНТИФІКАЦІЯ КОРИСТУВАЧІВ НА ОСНОВІ ASP.NET CORE IDENTITY” .....	5
ЛАБОРАТОРНА РОБОТА № 4. “ВИКОРИСТАННЯ БІБЛІОТЕКИ SIGNALR CORE ДЛЯ СТВОРЕННЯ ЧАТІВ З МИТТЄВИМ ОБМІНОМ ПОВІДОМЛЕННЯМИ.” .....	6
ЛАБОРАТОРНА РОБОТА № 5. “ПОБУДОВА REST-СЕРВІСУ З ВИКОРИСТАННЯМ WEB API” .....	7
ЛАБОРАТОРНА РОБОТА № 6. “РОЗРОБКА МІКРОСЕРВІСІВ” .....	8
ВАРІАНТИ ЗАВДАНЬ ДО ЛАБОРАТОРНИХ РОБІТ 1-3,5,6 .....	9
ВАРІАНТИ ЗАВДАНЬ ДЛЯ ЛАБОРАТОРНОЇ РОБОТИ № 4.....	11

## Загальні положення

- лабораторні роботи виконуються та оцінюються у формі проєктів MS Visual Studio 2022 (2019 або інших версій, що дозволяють створювати проєкти ASP.Net Core чи ASP.Net MVC);
- проєкти мають відповідати предметній області, вказаній у варіантах завдань нижче. Варіант обирається відповідно до номера студента у списку **підгрупи**:

Підгрупа 1.		Підгрупа 2.	
1	Вівчар Богдан	1	Добровольський Володимир
2	Голод Михайло	2	Король Станіслав
3	Голубнича Маргарита	3	Нагірна Андріана
4	Данильченко Катерина	4	Пушкар Максим
5	Козак Богдан	5	Романів Юліана
6	Костирка Анастасія	6	Савчак Юлія
7	Кравчишин Сергій	7	Торська Марія
	Ярослав-	8	Фесенко Мартін
8	Луцик Денис		
9	Младьонова Уляна		

- проєкти зберігаються у власному репозиторії на github, адреса якого вказується у звіті до лабораторної роботи;
- звіт до кожної лабораторної роботи повинен містити титульний аркуш, постановку задачі, вказані у лабораторній роботі елементи та посилання на репозиторій з працездатним проєктом;
- звіти до лабораторних робіт завантажуються у відповідне завдання курсу у ВНС у форматах текстових документів pdf, docx, odt, тощо.
- оцінка за лабораторну роботу виставляється на основі зданого до ВНС звіту після захисту роботи студентом на лабораторному занятті. Захист лабораторної роботи передбачає демонстрацію студентом основних елементів проєкту та їх коду, демонстрацію працездатності проєкту та основних елементів його функціоналу в дії, усні відповіді на запитання та зауваження викладача;
- максимальна кількість балів п.у. до кожної роботи вказана у відповідному завданні у ВНС;
- лабораторні роботи мають граничні терміни здачі (завантаження звіту) у ВНС. Для робіт зданих після вказаного терміну максимальна оцінка знижується на один бал за кожен тиждень протермінування.

## Лабораторна робота № 1. “Розробка web-додатку ASP.Net Core MVC”

Для заданої у варіанті завдання предметної області реалізувати інформаційну систему на базі додатку ASP.NET Core Web MVC.

- 1) Описати моделі предметної області та контекст БД з урахуванням реляційних зв'язків, властивості моделей задати самостійно (не менше 5 полів на кожную модель, але з врахуванням усіх особливостей предметної області);
- 2) Додаток повинен забезпечувати можливість виконання набору CRUD-операцій для кожної із сутностей заданої предметної області;
- 3) Реалізувати потрібні контролери та представлення (views) для роботи з даними ІС та розробити зручний інтерфейс для роботи з об'єктами заданої предметної області.

**У звіті має бути наведено:**

- 1) схема даних (малюнок з вказанням типів зв'язків між таблицями та відповідними ключовими полями);
- 2) 3-5 знімків різних сторінок інтерфейсу додатку у робочому стані (screenshot).
- 3) фрагменти коду одного з контролерів, та одного-двох представлень.

## **Лабораторна робота № 2. “Використання сесій для тимчасового зберігання даних в ASP.NET Core додатках”**

**До проєкту, створеного у ході виконання лабораторної роботи № 1 додати наступний функціонал:**

- 1) Забезпечити зберігання даних отриманих з бази для редагування в сесії та запис оновлених даних в базу лише після натискання відповідної кнопки;
- 2) Реалізувати валідацію даних на рівні моделі та її підтримку у представленнях;
- 3) Додати збереження у сесії даних що вводяться для запису в систему на час їх редагування та очистку сесії після запису.

**У звіті має бути наведено:**

- 1) Код моделей з атрибутами валідації;
- 2) Код компонентів, що відповідають за зберігання даних сесії.
- 3) Знімки екрану (screenshot) з прикладами вводу невалідних даних форми.

## **Лабораторна робота № 3. “Авторизація та автентифікація користувачів на основі ASP.Net Core Identity”**

**До проєкту, 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.

**У звіті має бути наведено:**

- 1) Фрагменти коду, який відповідає за налаштування автентифікації користувачів;
- 2) Код контролера та представлень (або код Razor-сторінок), які реалізують функціонал адмін-панелі;
- 3) Знімок екрану (screenshot) з виглядом адмін-панелі у браузері.

## **Лабораторна робота № 4. “Використання бібліотеки SignalR Core для створення чатів з миттєвим обміном повідомленнями.”**

**У проєкті, створеному у ході виконання лабораторних робіт №№ 1-3, відповідно до варіанту завдання:**

- 1) Створити Web-Чат з використанням технології SignalR для обміну повідомленнями між зареєстрованими користувачами;
- 2) Реалізувати можливості для надсилання приватних та загальних повідомлень;
- 3) Додати можливість передачі зображень чи інших файлів.

**У звіті має бути наведено:**

- 1) Фрагменти коду, що відповідає за логіку роботи чату;
- 2) Знімки екрану із списками повідомлень, форм для їх надсилання, тощо.

## Лабораторна робота № 5. “Побудова REST-сервісу з використанням Web API”

Для заданої у варіанті завдання предметної області реалізувати серверну частину REST-API з використанням ASP.NET Core Web API.

- 1) схему предметної області реалізувати аналогічно до попередніх лабораторних робіт;
- 2) для роботи з БД реалізувати патерн Repository;
- 3) додаток повинен забезпечувати можливість виконання набору CRUD-операцій для кожної із сутностей заданої предметної області;
- 4) Реалізувати потрібні WebAPI-контролери для кожної моделі, для тестування роутів можна скористатися вбудованим Swagger UI.

**У звіті має бути наведено:**

- 1) код одного з контролерів;
- 2) знімки екрану з прикладами виконання REST-запитів за допомогою Swagger UI.

## **Лабораторна робота № 6. “Розробка мікросервісів”**

**Для заданої у варіанті завдання предметної області реалізувати серверну частину REST-API з аналогічним до лабораторної роботи 5 функціоналом, перепроєктувавши схему даних під мікросервісну архітектуру окремими сервісами для кожної з моделей.**

**У звіті має бути наведено:**

- 1) Мікро сервісну архітектуру проекту.
- 2) Фрагменти коду основного функціоналу додатку.



## Варіанти завдань до лабораторних робіт 1-3,5,6

### 1. Соціальна мережа

- **Сутності:** Користувач і Група.
- **Опис:** Кожен користувач може бути учасником багатьох груп, і кожна група може мати багато користувачів. Завдання — розробити функціонал для управління користувачами та групами, включаючи додавання/видалення користувачів з груп.

### 2. Бібліотека

- **Сутності:** Читач і Книга.
- **Опис:** Кожен читач може взяти багато книг, і кожна книга може бути взята багатьма читачами (у різний час). Проєкт дозволяє управляти читачами та книгами, а також веде облік видачі та повернення книг.

### 3. Кінотеатр

- **Сутності:** Актор і Фільм.
- **Опис:** Кожен актор може зніматися в багатьох фільмах, і кожен фільм може мати багато акторів. Завдання — створити застосунок для управління акторами та фільмами, включаючи можливість додавати акторів до фільмів та навпаки.

### 4. Менеджер проєктів

- **Сутності:** Співробітник і Проєкт.
- **Опис:** Один співробітник може працювати над багатьма проєктами, і кожен проєкт може мати багато співробітників. Проєкт передбачає систему для управління співробітниками та проєктами, а також призначення співробітників до проєктів.

### 5. Інтернет-магазин

- **Сутності:** Товар і Замовлення.
- **Опис:** Кожен товар може бути в багатьох замовленнях, і кожне замовлення може містити багато товарів. Завдання — розробити систему для управління товарами та замовленнями, включаючи деталі замовлення (кількість кожного товару).

### 6. Спортивна ліга

- **Сутності:** Команда і Гравець.
- **Опис:** Гравець може грати за кілька команд (у різні періоди), і кожна команда має багато гравців. Завдання — створити систему для управління командами та гравцями, а також їхнім складом.

### 7. Туристична агенція

- **Сутності:** Клієнт і Тур.
- **Опис:** Клієнт може придбати багато турів, і кожен тур може бути придбаний багатьма клієнтами. Проєкт дозволяє вести облік клієнтів, турів і їхніх бронювань.

### 8. Онлайн-журнал

- **Сутності:** Автор і Стаття.

- **Опис:** Один автор може писати багато статей, і над однією статтею може працювати кілька авторів (співавторство). Завдання — створити систему управління авторами та статтями.

## 9. Кухня ресторану

- **Сутності:** Страва і Інгредієнт.
- **Опис:** Кожна страва складається з багатьох інгредієнтів, і один інгредієнт може використовуватися в багатьох стравах. Проєкт передбачає систему для управління меню та інгредієнтами, а також їхнім складом.

## 10. Музичний фестиваль

- **Сутності:** Виконавець і Сцена.
- **Опис:** Один виконавець може виступати на кількох сценах, і на кожній сцені можуть виступати багато виконавців. Проєкт дозволяє керувати інформацією про виконавців та сцени, а також формувати розклад виступів.

## 11. Форум

- **Сутності:** Користувач і Тег.
- **Опис:** Користувач може створювати багато тем з різними тегами, і кожен тег може бути використаний у багатьох темах. Завдання — створити систему для управління користувачами та тегами, а також ведення статистики використання тегів.

## 12. Електронний курс

- **Сутності:** Студент і Курс.
- **Опис:** Кожен студент може бути записаний на багато курсів, і кожен курс може мати багато студентів. Завдання — створити систему, що дозволяє управляти студентами, курсами та їхнім записом. Потрібно реалізувати CRUD-операції для обох сутностей, а також механізм додавання/видалення студентів з курсів.

## Варіанти завдань для лабораторної роботи № 4

### 1. Соціальна мережа

- **Завдання:** Інтегруйте чат для груп, щоб учасники могли спілкуватися.
  - **Функціонал:**
    - Створіть **хаб SignalR**, який дозволить спілкуватися в межах кожної групи.
    - Доступ до чату мають лише учасники групи.
- 

### 2. Бібліотека

- **Завдання:** Додайте чат для запитань читачів до бібліотекаря.
  - **Функціонал:**
    - Реалізуйте **хаб SignalR** для обміну повідомленнями між читачами та адміністратором (бібліотекарем).
    - Чат може бути загальним для всіх користувачів, які бажають задати запитання.
    - Повідомлення мають містити ім'я читача та текст запитання.
- 

### 3. Кінотеатр

- **Завдання:** Створіть чат для обговорення конкретного фільму на його сторінці.
  - **Функціонал:**
    - Розробіть **хаб SignalR**, який дозволить обговорювати фільми в реальному часі.
    - Чат прив'язаний до ідентифікатора фільму, що дозволяє користувачам обговорювати лише той фільм, на сторінці якого вони знаходяться.
- 

### 4. Менеджер проєктів

- **Завдання:** Інтегруйте чат для обговорення завдань усередині кожного проєкту.
  - **Функціонал:**
    - Створити **хаб SignalR** для кожного проєкту, щоб члени команди могли спілкуватися в реальному часі.
    - Чат повинен бути доступним тільки для учасників певного проєкту.
- 

### 5. Інтернет-магазин

- **Завдання:** Додайте чат для консультації клієнтів з менеджером магазину.
- **Функціонал:**
  - Реалізуйте **хаб SignalR** для підтримки в реальному часі.

- Клієнти можуть звертатися до менеджера з запитаннями щодо замовлень або товарів.

---

## 6. Спортивна ліга

- **Завдання:** Створіть чат для обговорення майбутніх матчів усередині кожної команди.
- **Функціонал:**
  - Розробіть **хаб SignalR** для спілкування між гравцями однієї команди.
  - Чат має бути закритим для інших команд.

---

## 7. Туристична агенція

- **Завдання:** Інтегруйте чат для клієнтів з менеджером для обговорення турів.
- **Функціонал:**
  - Створіть **хаб SignalR** для підтримки клієнтів.
  - Клієнти можуть ставити запитання щодо турів, готелів або авіаквитків.

---

## 8. Онлайн-журнал

- **Завдання:** Додайте чат для обговорення статті на її сторінці.
- **Функціонал:**
  - Реалізуйте **хаб SignalR** для кожної статті.
  - Користувачі можуть коментувати статтю в реальному часі, обмінюючись думками.

---

## 9. Кухня ресторану

- **Завдання:** Створіть чат для клієнтів, щоб вони могли спілкуватися з адміністратором щодо замовлень.
- **Функціонал:**
  - Розробіть **хаб SignalR** для запитань і уточнень щодо замовлень.
  - Клієнти можуть запитувати про статус замовлення або вносити корективи.

---

## 10. Музичний фестиваль

- **Завдання:** Додайте чат для обговорення кожного виконавця на його сторінці.
  - **Функціонал:**
    - Реалізуйте **хаб SignalR** для обговорення виступів та виконавців.
    - Користувачі можуть залишати свої враження про виступи в реальному часі.
-

## 11. Форум

- **Завдання:** Створіть чат для обговорення тем форуму.
  - **Функціонал:**
    - Розробіть **хаб SignalR**, який дозволить спілкуватися в межах кожної теми.
    - Користувачі можуть задавати запитання або відповідати на них, не перезавантажуючи сторінку.
- 

## 12. Електронний курс

- **Завдання:** Інтегруйте чат для кожного курсу, де студенти та викладачі можуть обмінюватися повідомленнями.
- **Функціонал:**
  - Створити **хаб SignalR**, що приймає повідомлення від користувачів.
  - Повідомлення повинні містити ім'я користувача, текст та ідентифікатор курсу.
  - Чат має бути ізольованим для кожного курсу.
  - Повідомлення відображаються миттєво для всіх учасників чату, що знаходяться на сторінці курсу.