

ВИПУСКНИЙ КВАЛІФІКАЦІЙНИЙ ПРОЄКТ

на тему:

«Клієнт-серверний додаток з реалізації наплічників»

Студентки 4 курсу, 7 групи,
спеціальності 121 «Інженерія
програмного забезпечення»
освітньої програми «Інженерія
програмного забезпечення»

Пересадько Ольги
Олегівни

підпис студента

Науковий керівник
кандидат технічних наук,
доцент кафедри інженерії
програмного забезпечення та
кібербезпеки

Власенко Лідія
Олександрівна

підпис керівника

Гарант освітньої програми
кандидат технічних наук,
доцент кафедри інженерії
програмного забезпечення та
кібербезпеки

Рзаєва Світлана
Леонідівна

підпис гаранта

Державний торговельно-економічний університет

Факультет інформаційних технологій

Кафедра інженерії програмного забезпечення та кібербезпеки

Освітній ступінь бакалавр

Спеціальність 121 «Інженерія програмного забезпечення»

Затверджую

Зав. кафедри інженерії програмного
забезпечення та кібербезпеки

Криворучко О. В.

«14» листопада 2023 р.

Завдання

на випускний кваліфікаційний проєкт студентіві

Пересадько Ользі Олегівні

(прізвище, ім'я, по батькові)

1. Тема випускного кваліфікаційного проєкту «Клієнт-серверний додаток
реалізації наплічників»

Затверджена наказом ректора від «6» грудня 2023 р. № 3288

2. Строк здачі студентом закінченого проєкту 5 червня 2023

3. Цільова установка та вихідні дані до проєкту

Мета проєкту створення клієнт-серверного додатку з реалізації наплічників

Об'єкт дослідження клієнт-серверний додаток реалізації наплічників

Предмет дослідження розробка клієнт-серверного додатку з реалізації наплічників

4. Консультанти проекту із зазначенням розділів, які консультують:

| Розділ | Консультант (прізвище, ініціали) | Підпис, дата | |
|--------|-------------------------------------|----------------|------------------|
| | | Завдання видав | Завдання прийняв |
| | | | |
| | | | |

5. Зміст випускного кваліфікаційного проекту (перелік питань за кожним розділом)

Вступ

РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ЗАСТОСУВАННЯ КЛІЄНТ-СЕРВЕРНОГО ДОДАТКУ

1. Загальні відомості

1.1. Найменування системи

1.2. Планові терміни початку та закінчення робіт

2. Мета та призначення створення системи

2.1. Призначення системи

2.2. Мета створення системи

3. Вимоги до системи

3.1. Загальні вимоги до системи

3.2. Вимоги до функціональності системи

3.3. Вимоги до надійності та продуктивності системи

3.4. Вимоги до безпеки системи

3.5. Вимоги до інтерфейсу користувача

4. Вимоги до програмного забезпечення

5. Обґрунтування вибору мови програмування

6. Вибір середовища розробки

7. Інструментарій розробки клієнт-серверного додатку для реалізації наплічників

8. Предметна область

9. Висновки до розділу 1

Розділ 2 Вибір програмних засобів та проектування веб-сайту

1. Архітектура додатку

1.1. Use-case Діаграма

1.2. Еталонна модель

1.3. Блок схема

1.4. Діаграма послідовності

1.5. Діаграма стану

1.6. Діаграма класів

1.7. Діаграма розгортання

2. Бази даних

2.1. Логічна модель бази даних

2.2. Фізична модель бази даних

3. Висновки до розділу 2

| | | | | | | | |
|-----|-------|---------|--------|------|--|-------------------|-------|
| | | | | | | ДТЕУ 121 07-19.БР | Аркуш |
| Зм. | Аркуш | № докум | Підпис | Дата | | | 6 |

РОЗДІЛ 3 РЕАЛІЗАЦІЯ КЛІЄНТ-СЕРВЕРНОГО ДОДАТКУ

- 1. Встановлення Visual Studio**
- 2. Встановлення SQL Server Express**
- 3. Створення проєкту SwitchBook**
- 4. Підключення бази даних до проєкту та налаштування Identity Framework**
- 5. Створення моделей даних, бази даних і таблицок.**
- 6. Створення контролерів.**
- 7. Розробка логіки та додавання контексту даних до контролерів.**
- 8. Результат роботи.**

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ



Державний торговельно-економічний університет

6. Календарний план виконання проекту

| № пор. | Назва етапів випускного кваліфікаційного проекту | Строк виконання етапів проекту | |
|--------|---|--------------------------------|------------|
| | | за планом | Фактично |
| 1 | 2 | 3 | 4 |
| 1. | <i>Вибір теми випускного кваліфікаційного проекту</i> | 21.09.2022 | 21.09.2022 |
| 2. | <i>Розробка та затвердження завдання на проект</i> | 14.11.2022 | 14.11.2022 |
| 3. | <i>Вступ та перелік літературних джерел</i> | 23.12.2022 | 23.12.2022 |
| 4. | <i>Розділ 1. Аналіз предметної області продажу вантажних шин</i> | 27.01.2023 | 27.01.2023 |
| 5. | <i>Розділ 2. Вибір програмних засобів та проектування веб-додатку</i> | 03.03.2023 | 03.03.2023 |
| 6. | <i>Розділ 3. Реалізація веб-додатку</i> | 14.04.2023 | 14.04.2023 |
| 7. | <i>Висновки</i> | 28.04.2023 | 28.04.2023 |
| 8. | <i>Здача випускного кваліфікаційного проекту на кафедрі (перша перевірка)</i> | 17.05.2023 | 17.05.2023 |
| 9. | <i>Підготовка автореферату та презентації доповіді</i> | 26.05.2023 | 26.05.2023 |
| 10. | <i>Попередній захист випускного кваліфікаційного проекту</i> | 29.05.2023 – 02.06.2023 | |
| 11. | <i>Зовнішнє рецензування випускного кваліфікаційного проекту</i> | 05.06.2023 | 05.06.2023 |
| 12. | <i>Здача прошого випускного кваліфікаційного проекту на кафедрі</i> | 05.06.2023 | 05.06.2023 |
| 13. | <i>Публічний захист випускного кваліфікаційного проекту</i> | | |

7. Дата видачі завдання «14» листопада 2022 р.

7. Науковий керівник випускного кваліфікаційного проекту

Власенко ЛідіяОлександрівна

(прізвище, ініціали, підпис)

8. Гарант освітньої програми

Рзаєва С.Л.

(прізвище, ініціали, підпис)

9. Завдання прийняв до виконання студент

Пересадько О. О.

(прізвище, ініціали, підпис)

11. Відгук керівника випускного кваліфікаційного проєкту

Науковий керівник випускного кваліфікаційного проєкту

(підпис, дата)

Відмітка про попередній захист

(ПІБ, підпис, дата)

12. Висновок про випускний кваліфікаційний проєкт

Випускний кваліфікаційний проєкт студента Пересадько О. О.

(прізвище, ініціали)

може бути допущений до захисту екзаменаційній комісії.

Гарант освітньої програми

Рзаєва С.Л.

(прізвище, ініціали, підпис)

Завідувач кафедри

Криворучко О. В.

(підпис, прізвище, ініціали)

« » 20 р.

АНОТАЦІЯ

Відповідно до мети дослідження робота присвячена реалізації клієнт-серверного додатку для наплічників.

В результаті порівняльного аналізу аналогічних рішень визначено, що на ринку України недостатньо подібних продуктів.

Розробка серверної частини виконана за допомогою .NET Core 9.0.

Готовий Клієнт-серверний додаток реалізації наплічників було успішно розроблено і протестовано.

Ключові слова: Клієнт-серверна архітектура, Наплічники, Рюкзаки, База даних, Діаграма, Логіка програмних продуктів, Проєкт SwitchBook, Ukrainian backpacks.

ABSTRACT

According to the purpose of the study, the work is devoted to the implementation of a client-server application for shoulder pads.

As a result of the comparative analysis of similar solutions, it was determined that there are not enough similar products on the Ukrainian market.

The development of the server part was carried out using .NET Core 9.0.

The finished Client-server application for the implementation of shoulder pads has been successfully developed and tested.

As a result of a comparative analysis of similar solutions, it was determined that there are few similar software applications on the Ukrainian market

The development of the server side was performed using .NET Core 9.0 The finished DMS software package was successfully developed and tested.

Keywords: Client-server architecture, Backpacks, Backpacks, Database, Diagram, Logic of software products, SwitchBook project, Ukrainian backpacks.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

C# - мова програмування C-Sharp

XSS - Cross-Site Scripting

PHP - мова програмування PHP

Asp.net - фреймворк Asp.net (Active Server Pages .NET)

SQL - мова структурованих запитів SQL (Structured Query Language)

.NET - фреймворк .NET (Microsoft .NET)

OS - операційна система

ПЗ - програмне забезпечення

БД – база даних

ПК - персональний комп'ютер

IDE - інтегроване середовище розробки

Предметна область - область дослідження/застосування

TCP (протокол TCP), SSL-сертифікат (SSL-сертифікат)

UML (Unified Modeling Language) - уніфікована мова моделювання

Ст. (Стан) - позначення для стану

ООП - Об'єктно-орієнтоване програмування

СУБД - система управління базами даних

URL - веб-адреса

EXE - виконуваний файл

СМС - SQL Server Management Studio

| | | | | | | | |
|---|--------------|------------------|---------------|-------------|---------------|--------------|----------------|
| <i>ДТЕУ 121 07-19.БР</i> | | | | | | | |
| <i>Зм.</i> | <i>Аркуш</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> | | | |
| Зав. каф. | | Криворучко О.В. | | 23.12.22 | | | |
| Керівник | | Власенко Л. О. | | 23.12.22 | | | |
| Гарант | | Рзаева С.Л. | | 23.12.22 | | | |
| Розробив | | Пересацько О. О. | | 23.12.22 | | | |
| <i>Зміст</i> | | | | | | | |
| Клієнт-серверний додаток реалізації наплічників | | | | | <i>Стадія</i> | <i>Аркуш</i> | <i>Аркушів</i> |
| | | | | | <i>ПС</i> | <i>1</i> | <i>51</i> |
| <i>Факультет інформаційних технологій 4 курс, 7 група</i> | | | | | | | |

ЗМІСТ

| | |
|---|-----------|
| ВСТУП..... | 11 |
| РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТА ДОСЛІДЖЕННЯ І ОПИС ЙОГО ОСНОВНИХ ПАРАМЕТРІВ ТА ХАРАКТЕРИСТИК..... | 14 |
| 1. Загальні відомості | 14 |
| 1.1. Найменування системи..... | 14 |
| 1.2. Планові терміни початку та закінчення робіт | 14 |
| 2. Мета та призначення створення системи | 14 |
| 2.1. Призначення системи..... | 14 |
| 2.2. Мета створення системи | 14 |
| 3. Вимоги до системи | 15 |
| 3.1. Загальні вимоги до системи..... | 15 |
| 3.2. Вимоги до функціональності системи..... | 16 |
| 3.3. Вимоги до надійності та продуктивності системи..... | 16 |
| 3.4. Вимоги до безпеки системи | 17 |
| 3.5. Вимоги до інтерфейсу користувача..... | 17 |
| 4. Вимоги до програмного забезпечення | 18 |
| 5. Обґрунтування вибору мови програмування | 18 |
| 6. Вибір середовища розробки | 20 |
| 7. Інструментарій розробки клієнт-серверного додатку для реалізації наплічників | 21 |
| 8. Предметна область..... | 22 |
| Висновки до розділу 1..... | 23 |
| РОЗДІЛ 2 ВИБІР ПРОГРАМНИХ ЗАСОБІВ ТА ПРОЄКТУВАННЯ КЛІЄНТ-СЕРВЕРНОГО ДОДАТКУ..... | 24 |
| 1. Архітектура додатку | 24 |
| 1.1. Use-case Діаграма..... | 24 |
| 1.2. Еталонна модель..... | 25 |
| 1.3. Блок схема | 27 |
| 1.4. Діаграма послідовності..... | 28 |
| 1.5. Діаграма стану..... | 29 |
| 1.6. Діаграма класів..... | 30 |
| 1.7. Діаграма розгортання..... | 31 |

| | | | | | |
|--------------------------|-------|------------------|---|----------|---------|
| <i>ДТЕУ 121 07-19.БР</i> | | | | | |
| Зм. | Аркуш | № докум. | Підпис | Дата | |
| Зав. каф. | | Криворучко О.В. | | 23.12.22 | |
| Керівник | | Власенко Л. О. | | 23.12.22 | |
| Гарант | | Рзаєва С.Л. | | 23.12.22 | |
| Розробив | | Пересацько О. О. | | 23.12.22 | |
| Зміст | | | | | |
| | | | Стадія | Аркуш | Аркушів |
| | | | ПС | 2 | 51 |
| | | | Факультет інформаційних технологій 4 курс, 7 група | | |

| | |
|--|-----------|
| 2. Бази даних | 32 |
| 2.1.Логічна модель бази даних | 32 |
| 2.2Фізична модель бази даних | 33 |
| Висновки до розділу 2 | 40 |
| РОЗДІЛ 3 РЕАЛІЗАЦІЯ КЛІЄНТ-СЕРВІСНОГО ДОДАТКУ | 42 |
| 1. Встановлення Visual Studio | 42 |
| 2. Встановлення SQL Server Express | 43 |
| 3. Створення проєкту Switch Book | 46 |
| 4. Підключення бази даних до проєкту та налаштування Identity Framework | 46 |
| 5. Створення моделей даних, бази даних і таблиць | 47 |
| 6. Створення контролерів | 50 |
| 7 Розробка логіки та додавання контексту даних до контролерів | 51 |
| Висновки до розділу 3 | 53 |
| ВИСНОВКИ ТА ПРОПОЗИЦІЇ | 57 |
| СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ | 58 |
| Додатки | 32 |

| | | | | | | | | |
|-----------|-------|------------------|--------|----------|--|--|-------|---------|
| | | | | | <i>ДТЕУ 121 07-19.БР</i> | | | |
| Зм. | Аркуш | № докум. | Підпис | Дата | Клієнт-серверний додаток реалізації наплічників | Стадія | Аркуш | Аркушів |
| Зав. каф. | | Криворучко О.В. | | 23.12.22 | | ПС | 2 | 51 |
| Керівник | | Власенко Л. О. | | 23.12.22 | | Факультет інформаційних технологій 4 курс, 7 група | | |
| Гарант | | Рзаєва С.Л. | | 23.12.22 | | | | |
| Розробив | | Пересацько О. О. | | 23.12.22 | | | | |
| | | | | | Зміст | | | |

ВСТУП

У сучасному світі електронна комерція стала невід'ємною частиною життя. Із зростанням популярності онлайн-покупок, стає все важливішим розвиток технологій для електронної комерції. Один з таких технологій - клієнт-серверні додатки, які дозволяють забезпечувати обробку замовлень онлайн та здійснювати платіжні операції безпечно та зручно.

В рамках даної випускної роботи було створено клієнт-серверний додаток для реалізації рюкзаків. Даний додаток дозволяє користувачам швидко та зручно здійснювати покупки рюкзаків онлайн. Клієнт-серверний додаток дозволяє користувачам шукати рюкзаки за різними критеріями, відбирати за вартістю та функціональністю. Крім того, користувачі можуть забезпечувати оплату за покупки онлайн, зручно та безпечно.

Клієнтський додаток містить детальну інформацію про різні види рюкзаків, яку можна відсортувати за різними критеріями. Крім того, додаток містить можливість зберігання інформації про користувачів та їх замовленнях. Він також має можливість зв'язування з різними платіжними системами, що забезпечує безпечну оплату.

Серверна частина додатку забезпечує зберігання всієї інформації про рюкзаки та користувачів, яка може бути запитана з клієнтської частини. Також серверна частина забезпечує забезпечення безпеки при здійсненні платіжних операцій.

Є багато причин, чому люди надають перевагу інтернет магазинам. Перш за все, перед купівлею покупці можуть переглянути відгуки та оцінки

| | | | | | | | | |
|-----------|-------|------------------|--------|----------|--|--|-------|---------|
| | | | | | <i>ДТЕУ 121 07-19.БР</i> | | | |
| Зм. | Аркуш | № докум. | Підпис | Дата | Клієнт-серверний додаток реалізації наплічників | Стадія | Аркуш | Аркушів |
| Зав. каф. | | Криворучко О.В. | | 23.12.22 | | ПС | 4 | 51 |
| Керівник | | Власенко Л. О. | | 23.12.22 | | Факультет інформаційних технологій 4 курс, 7 група | | |
| Гарант | | Рзаєва С.Л. | | 23.12.22 | | | | |
| Розробив | | Пересацько О. О. | | 23.12.22 | | | | |
| | | | | | <i>Вступ</i> | | | |

інших покупців про конкретний рюкзак, що допомагає їм зробити більш обізнаний вибір.

Другим аспектом є зручність покупок. Покупець може вибрати рюкзак на сайті, ознайомитися з усіма характеристиками, відгуками покупців та зробити замовлення в зручний для нього час. Також, Інтернет-магазини рюкзаків часто пропонують різні знижки та промокоди, що робить покупку ще вигіднішою.

По-третє, в Інтернеті легше зіставляти ціни на різні моделі рюкзаків та знайти найкращу пропозицію. Також, можливість швидко та зручно зв'язатися з продавцем за допомогою онлайн-чату або електронної пошти робить покупку ще комфортнішою.

Окрім того, продаж рюкзаків в Інтернеті дозволяє продавцям ефективно рекламувати свої товари, використовуючи цільове таргетування та інші інструменти маркетингу. Це дозволяє залучати нових клієнтів та збільшувати продажі. Також, продаж рюкзаків в Інтернеті дає змогу залучити клієнтів з усього світу.

Результатом випускної кваліфікаційної роботи є розроблений та протестований клієнт-серверний додаток, який забезпечує зручність та простоту у керуванні наплічниками.

Отже, розробка клієнт-серверного додатка є важливим етапом в розробці програмного забезпечення для керування наплічниками. Таке програмне забезпечення забезпечує користувачам зручність та швидкість у виконанні різних завдань.

Надалі, можливості такого додатка можуть бути розширені та вдосконалені для підтримки більш широкого кола функцій та розвитку сучасних технологій.

Метою даної дипломної роботи є створення простого та зручного клієнт-серверного додатка для інтернет-магазину наплічників. Ця база даних

| | | | | | | | |
|-----|-------|---------|--------|------|--|-------------------|-------|
| | | | | | | ДТЕУ 121 07-19.БР | Аркуш |
| Зм. | Аркуш | № докум | Підпис | Дата | | | 5 |

дозволить власникам бізнесу зберігати інформацію про товари, замовлення та клієнтів, і легко керувати цими даними

Об'єктом дослідження є клієнт-серверний додаток реалізації наплічників

Предметом дослідження є розробка дизайну, вибір технологій для створення веб сайту для реалізації наплічників.



| | | | | | | |
|-----|-------|---------|--------|------|-------------------|-------|
| | | | | | ДТЕУ 121 07-19.БР | Аркуш |
| Зм. | Аркуш | № докум | Підпис | Дата | | 6 |

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТА ДОСЛІДЖЕННЯ І ОПИС ЙОГО ОСНОВНИХ ПАРАМЕТРІВ ТА ХАРАКТЕРИСТИК

1. Загальні відомості

1.1. Найменування системи

Повним найменуванням системи є клієнт-серверний додаток реалізації наплічників « Ukrainian backpacks ».

Скорочене найменування системи: веб сайт «Ukrainian backpacks».

1.2. Планові терміни початку та закінчення робіт

Заплановано почати розробку 30.01.2023р. Закінчити роботу заплановано до 31.04.2023р.

2. Мета та призначення створення системи

2.1. Призначення системи

Створення онлайн-сервісу з реалізацією наплічників.

2.2. Мета створення системи

1. Забезпечення зручної та простої платформи для покупців, де вони зможуть переглядати широкий вибір рюкзаків, порівнювати їх характеристики та зручно оформляти замовлення;

2. Надання інформації про різноманітність брендів, моделей, розмірів, кольорів та стилів рюкзаків, щоб покупці могли знайти саме той, що підходить найкраще.

| | | | | | ДТЕУ 121 07-19.БР | | | |
|-----------|-------|-----------------|--------|----------|---|------------------------------------|-------|---------|
| Зм. | Аркуш | № докум. | Підпис | Дата | | | | |
| Зав. каф. | | Криворучко О.В. | | 27.12.22 | Клієнт-серверний додаток реалізації наплічників | Стадія | Аркуш | Аркушів |
| Керівник | | Власенко О.О. | | 27.12.22 | | РІ | 7 | 51 |
| Гарант | | Рзаєва С.Л. | | 27.12.22 | | Факультет інформаційних технологій | | |
| Розробив | | Пересадько О.О. | | 27.12.22 | | 4 курс, 7 група | | |
| | | | | | Розділ 1 | | | |

3. Вимоги до системи

3.1. Загальні вимоги до системи

3.1.1. Вимоги до структури та функціонування системи включають::

- Забезпечення логічної та зрозумілої структури для користувача;
- Наявність підсистем, таких як авторизація користувачів, каталог наплічників та форма для залишення заявки;
- Необхідно, щоб користувач зрозуміло розумів відображення каталогу товарів та форми оформлення замовлення.

3.1.2 Вимоги до взаємодії з користувачем;

- Користувач має бути в змозі зареєструватися та авторизуватися в системі;
- Користувач повинен мати можливість здійснювати пошук товарів на веб-сайті;
- Користувач повинен мати можливість зробити замовлення товару.

3.1.2. Вимоги до безпеки та захисту даних

- Користувачу треба надати можливість реєструватись в системі та змінювати свої персональні данні і пароль;
- Застосування шифрування SSL/TLS для захисту передачі конфіденційних даних між користувачем і сервером, зокрема при введенні особистої інформації та паролю.

3.1.3. Вимоги до мов та технологій

- Необхідно створити систему з використанням мови програмування C# на платформі Asp.Net, яка використовуватиме базу даних SQL та Entity Framework;

| | | | | | |
|-----|-------|---------|--------|------|-------|
| | | | | | Аркуш |
| | | | | | 8 |
| Зм. | Аркуш | № докум | Підпис | Дата | |

ДТЕУ 121 07-19.БР

- Весь інтерфейс має бути українською мовою.

3.2. Вимоги до функціональності системи

3.2.1. Реєстрація та авторизація користувачів

- Реєстрація користувача можлива, при введенні персональної інформації, такої як ім'я, прізвище, номер мобільного телефону, адреса, поштовий код;

- Авторизація здійснюється при введенні електронної пошти та паролю.

3.2.2. Каталог наплічників

- Система повинна містити каталог товарів, що складається з наступних елементів: назва товару, ціна, склад, виробник, колір;

- Користувач повинен мати можливість переглядати каталог товарів та шукати їх за ключовими словами.

3.2.3. Купівля наплічників

- Користувач повинен мати можливість створити замовлення, вказавши свій номер телефону, адресу та фамілію, ім'я;

- Після підтвердження повинна з'явитися контактна інформація про користувачів.

3.3. Вимоги до продуктивності та надійності системи

3.3.1. Надійність

- Клієнт-серверний додаток повинен працювати без збоїв та помилок.

3.3.2. Продуктивність

- Система повинна забезпечувати швидкий доступ до даних; Система повинна підтримувати багатокористувацьку роботу та обробку багатьох операцій з базою даних одночасно.

| | | | | | |
|-----|-------|---------|--------|------|-------|
| | | | | | Аркуш |
| | | | | | 9 |
| Зм. | Аркуш | № докум | Підпис | Дата | |

ДТЕУ 121 07-19.БР

3.4. Вимоги до безпеки системи

3.4.1. Аутентифікація та авторизація

- Для доступу до приватної інформації користувача, обов'язково користувач має бути авторизований.

3.4.2. Захист даних

- Система має гарантувати, що конфіденційна і особиста інформація користувачів буде захищена від несанкціонованого доступу і втрати даних.;
- Система повинна забезпечувати захист від шкідливого програмного забезпечення та атак на безпеку даних.

3.5. Вимоги до інтерфейсу користувача

3.5.1. Зручність використання

- Інтерфейс повинен бути зрозумілим для користувачів будь-якого рівня технічної грамотності;
- Інтерфейс повинен мати зручну взаємодію на різних пристроях, включаючи і мобільні пристрої, планшети і комп'ютери.

3.5.2. Надійність інтерфейсу

- Інтерфейс повинен бути сумісним з різними веб-браузерами (наприклад, Chrome, Firefox, Safari, Internet Explorer) та пристроями (настільні комп'ютери, ноутбуки, планшети, мобільні телефони). Він повинен належним чином відображатися та працювати на всіх цих платформах;
- Важливо, щоб інтерфейс сайту працював беззбоєю, без збоїв та перебоїв. Це означає, що сайт повинен бути стійким до високого навантаження та забезпечувати швидкий відгук на запити користувачів;
- Інтерфейс повинен бути простим у використанні та інтуїтивно зрозумілим для користувачів. Меню, кнопки та інші елементи керування

| | | | | | | |
|-----|-------|---------|--------|------|-------------------|-------|
| | | | | | ДТЕУ 121 07-19.БР | Аркуш |
| Зм. | Аркуш | № докум | Підпис | Дата | | 10 |

повинні бути розміщені логічно та зрозуміло. Навігація має бути зручною та легко доступною, дозволяючи користувачам швидко знаходити потрібну інформацію та продукти;

4. Вимоги до програмного забезпечення

Система повинна бути розроблена з використанням сучасних технологій та мов програмування.

- Система повинна бути масштабованою та легко розширюватись та зменшуватись;
- Система повинна бути сумісною з різними операційними системами та браузерами;
- Система повинна бути безпечною та захищеною від вторгнень та зловживань;
- Система має бути легкою та зрозумілою для людей з різним рівнем технічної освітченості.

5. Обґрунтування вибору мови програмування

Написання сайту для продажу мовою C# має декілька переваг.

1. Безпека: C# має вбудовану систему безпеки, яка дозволяє писати код з меншою кількістю помилок безпеки, таких як SQL-ін'єкції або XSS-атаки;
2. Розширюваність: C# має потужну систему об'єктно-орієнтованого програмування, що дозволяє легко розширювати функціонал сайту та додавати нові функції без необхідності переписування коду з нуля;
3. Швидкість: C# є компільованою мовою програмування, що дозволяє отримати значно кращу продуктивність та швидкість роботи сайту, порівняно з інтерпретованими мовами, такими як PHP;

| | | | | | |
|-----|-------|---------|--------|------|-------|
| | | | | | Аркуш |
| | | | | | 11 |
| Зм. | Аркуш | № докум | Підпис | Дата | |

ДТЕУ 121 07-19.БР

4. Велика спільнота: C# є досить популярною мовою програмування, що має велику спільноту розробників та підтримувальників, що забезпечує високу якість коду та швидку відповідь на будь-які запитання або проблеми, що виникають під час розробки;

5. Використання фреймворків: C# підтримує різноманітні фреймворки, такі як ASP.NET, які дозволяють розробникам швидко та ефективно створювати високоякісні сайти для продажу, що мають великий функціонал та високу продуктивність.

Отже, писання сайту для продажу мовою C# може допомогти створити безпечний, швидкий та ефективний сайт з великим функціоналом та можливістю легко розширювати його в майбутньому.

Asp.net є однією з найпопулярніших платформ для розробки веб-додатків, особливо для підприємственого застосування. Ось декілька причин, чому Asp.net є відмінним вибором для розробки веб-сайтів:

1. Широкий вибір інструментів: Asp.net має широкий вибір інструментів, які дозволяють розробникам швидко та ефективно створювати веб-додатки. Такі інструменти, як Visual Studio та .NET Framework, надають розробникам потужний набір інструментів для створення веб-сайтів.

2. Висока продуктивність: Asp.net дозволяє розробникам створювати веб-додатки з високою продуктивністю та ефективністю. Він забезпечує швидкий час відгуку та обробки запитів, що дозволяє забезпечити користувачів швидкою та якісною роботою сайту.

3. Масштабованість: Asp.net дозволяє розробникам легко масштабувати веб-сайти від невеликих розмірів до великих підприємствених веб-додатків. Це дозволяє компаніям збільшувати ресурси свого веб-сайту по мірі зростання бізнесу.

| | | | | | | |
|-----|-------|---------|--------|------|-------------------|-------|
| | | | | | | Аркуш |
| | | | | | ДТЕУ 121 07-19.БР | 12 |
| Зм. | Аркуш | № докум | Підпис | Дата | | |

Створення веб-сайту дає можливість використовувати програмний продукт на будь-якій операційній системі, яка дозволяє встановлювати веб-браузер. Це означає, що будь-який користувач, який має доступ до веб-браузера, може використовувати сайт, який буде адаптований до пристроїв з різною шириною екрану, незалежно від того, чи він використовує мобільний пристрій чи ПК.

6. Вибір середовища розробки

Для створення клієнт-серверного додатку обрано середовище Visual Studio. Visual Studio - це інтегроване середовище розробки (IDE), яке підтримує мову C# та фреймворк Asp.Net. Це середовище має багато переваг, ось декілька з них:

1. Інтегроване середовище розробки: Visual Studio надає повний спектр інструментів для розробки веб-сайтів, включаючи редактор коду, візуальний дизайнер, налагоджувач, пакет менеджер і інші інструменти, що зменшують час розробки та збільшують продуктивність.

2. Розширюваність: Visual Studio має велику кількість плагінів та розширень, що дозволяють збільшити функціональність та розширити можливості розробки веб-сайтів.

3. Налагоджування: Visual Studio надає налагоджувач, який дозволяє відстежувати та виправляти помилки в програмному коді в реальному часі, що збільшує ефективність розробки та зменшує час на відлагодження.

4. Вбудована система контролю версій: Visual Studio має вбудовану систему контролю версій, яка дозволяє відстежувати зміни в програмному коді, створювати гілки та об'єднувати їх, що зменшує ризик втрати даних та спрощує роботу.

5. Підтримка платформи Azure: Visual Studio має інтегровану підтримку платформи Azure, що дозволяє розгортати веб-сайти безпосередньо

| | | | | | | |
|-----|-------|---------|--------|------|-------------------|-------|
| | | | | | ДТЕУ 121 07-19.БР | Аркуш |
| Зм. | Аркуш | № докум | Підпис | Дата | | 13 |

з Visual Studio та забезпечує швидке та просте розгортання веб-сайту на хмарній платформі.

7. Інструментарій розробки клієнт-серверного додатку для реалізації наплічників

Для розробки веб-сайту для реалізації наплічників використовуватиметься наступний інструментарій:

1. Visual Studio - це інтегроване середовище розробки (IDE), яке підтримує мову C# та фреймворк Asp.Net. Visual Studio має інтегровану систему контролю версій, що дозволяє розробникам спільно працювати над проектами, а також підтримує різні сервіси хмарної розробки.

2. C# - це об'єктно-орієнтована мова програмування. C# має просту модель пам'яті і підтримує автоматичне управління пам'яттю, що робить її безпечною і зручною у використанні.

3. ASP.NET є фреймворком для веб-розробки. Він дозволяє розробникам створювати динамічні веб-сайти, веб-додатки та веб-сервіси з використанням мов програмування .NET, таких як C#. ASP.NET забезпечує середовище виконання для веб-додатків, що дозволяє розробникам зосередитися на написанні бізнес-логіки, замість того, щоб турбуватися про безпеку, взаємодію з базами даних та інші складнощі. Він також має багатий набір інструментів та бібліотек для швидкої розробки веб-додатків.

4. SQL Server - це програмне забезпечення для управління базами даних. Він дозволяє зберігати, організувати та керувати реляційними базами даних, що містять структуровані дані. SQL Server надає можливість зберігати, оновлювати, видаляти та отримувати дані, а також виконувати складні операції обробки даних та створювати звіти на основі даних.

| | | | | | | |
|-----|-------|---------|--------|------|-------------------|-------|
| 5 | | | | | | Аркуш |
| | | | | | ДТЕУ 121 07-19.БР | 14 |
| Зм. | Аркуш | № докум | Підпис | Дата | | |

5. Entity Framework (EF) - це технологія об'єктно-орієнтованого картографування даних, яка дозволяє розробникам .NET створювати програми, які працюють з базами даних, використовуючи високорівневі об'єкти, які представляють таблиці бази даних.

6. Identity framework - це набір засобів та протоколів, призначених для забезпечення безпеки та ідентифікації користувачів в різних інтернет-системах. Цей фреймворк зазвичай включає стандарти та протоколи, які дозволяють системам автоматично перевіряти та перевіряти ідентифікаційні дані користувачів з довіреними сторонами, такими як соціальні мережі, інші сервіси та технології. Identity framework є важливою складовою сучасної безпеки даних та дозволяє користувачам зручно та безпечно взаємодіяти з багатьма різними сервісами та системами в Інтернеті.

8. Предметна область

Веб-сайт для торгівлі наплічниками - це електронний магазин, який спеціалізується на продажі рюкзаків різних видів та розмірів. На сайті можна знайти широкий вибір рюкзаків для різних потреб - для подорожей, спорту, навчання, роботи тощо. Клієнти можуть шукати продукти за різними критеріями, такими як розмір, матеріал, колір, виробник тощо. Веб-сайт також надає детальну інформацію про кожний продукт, включаючи опис, фотографії, відгуки покупців та ціну. Крім того, клієнти можуть створювати облікові записи на сайті, додавати продукти в кошик, здійснювати онлайн-оплату та відстежувати статус свого замовлення. На сайті також можуть бути різні акції та знижки на певні продукти або для нових клієнтів. Метою веб-сайту є забезпечення зручної та ефективної платформи для покупки рюкзаків, яка задовольняє потреби різних клієнтів.

| | | | | | | |
|-----|-------|---------|--------|------|-------------------|-------|
| | | | | | ДТЕУ 121 07-19.БР | Аркуш |
| | | | | | | 15 |
| Зм. | Аркуш | № докум | Підпис | Дата | | |

Висновки до розділу 1

Отже, вибір мови програмування C# для написання веб-сайту для продажу рюкзаків має багато переваг, таких як швидкість, велика спільнота розробників та підтримувальників, а також підтримка різноманітних фреймворків, таких як ASP.NET. Це дозволяє створити безпечний, швидкий та ефективний веб-сайт з великим функціоналом і можливістю легко розширювати його в майбутньому.

Крім того, вибір середовища розробки Visual Studio є відмінним рішенням, оскільки воно надає повний спектр інструментів для розробки веб-сайтів, забезпечує розширюваність та має вбудовану систему контролю версій. Використання мови програмування C# разом з фреймворком ASP.NET і середовищем Visual Studio дозволить зручно і ефективно розробляти клієнт-серверний додаток для реалізації наплічників.

Також варто зазначити, що для розробки веб-сайту будуть використовуватись інші інструменти, такі як SQL Server для управління базами даних. В цілому, створення веб-сайту для торгівлі наплічками потребує використання різних технологій та інструментів, що дозволяють забезпечити зручну навігацію, безпеку, швидкість та ефективність роботи сайту, а також забезпечити зручність використання для клієнтів.

| | | | | | | |
|-----|-------|---------|--------|------|-------------------|-------|
| | | | | | ДТЕУ 121 07-19.БР | Аркуш |
| Зм. | Аркуш | № докум | Підпис | Дата | | 16 |

РОЗДІЛ 2

ВИБІР ПРОГРАМНИХ ЗАСОБІВ ТА ПРОЄКТУВАННЯ КЛІЄНТ-СЕРВЕРНОГО ДОДАТКУ

1. Архітектура додатку

1.1. Use-case Діаграма

На рисунку 1 зображена Use-case діаграма. Вона описує поведінку системи з точки зору її користувачів та їх взаємодії з системою. У ній вказуються всі можливі сценарії взаємодії між користувачем та системою, а також дії, які виконуються в процесі взаємодії.

1. Не зареєстрований користувач: це користувач, який відвідує сайт без реєстрації або входу в систему. Він може переглядати каталог товарів, шукати за ключовими словами та фільтрувати за характеристиками товарів.

2. Зареєстрований користувач: це користувач, який має обліковий запис на сайті. Він може робити те ж саме, що і гість, але також може редагувати свій профіль, додавати товари в кошик та розміщувати замовлення.

3. Адміністратор сайту: це користувач з правами доступу до адміністративної панелі сайту. Він може додавати та видаляти товари з каталогу, редагувати їх характеристики, встановлювати ціни, керувати замовленнями, а також стежити за аналітикою сайту.

| | | | | | | | | |
|------------|-----------------|-----------------|---------------|-------------|--------------------------|--|--------------|----------------|
| | | | | | <i>ДТЕУ 121 07-19.БР</i> | | | |
| <i>Зм.</i> | <i>Аркуш</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> | <i>Розділ 2</i> | <i>Стадія</i> | <i>Аркуш</i> | <i>Аркушів</i> |
| Зав. каф. | Криворучко О.В. | | | 05.01.23 | | <i>P2</i> | <i>17</i> | <i>51</i> |
| Керівник | Власенко Л. О. | | | 05.01.23 | | Факультет інформаційних технологій <i>4 курс, 7 група</i> | | |
| Гарант | Рзаєва С.Л. | | | 05.01.23 | | | | |

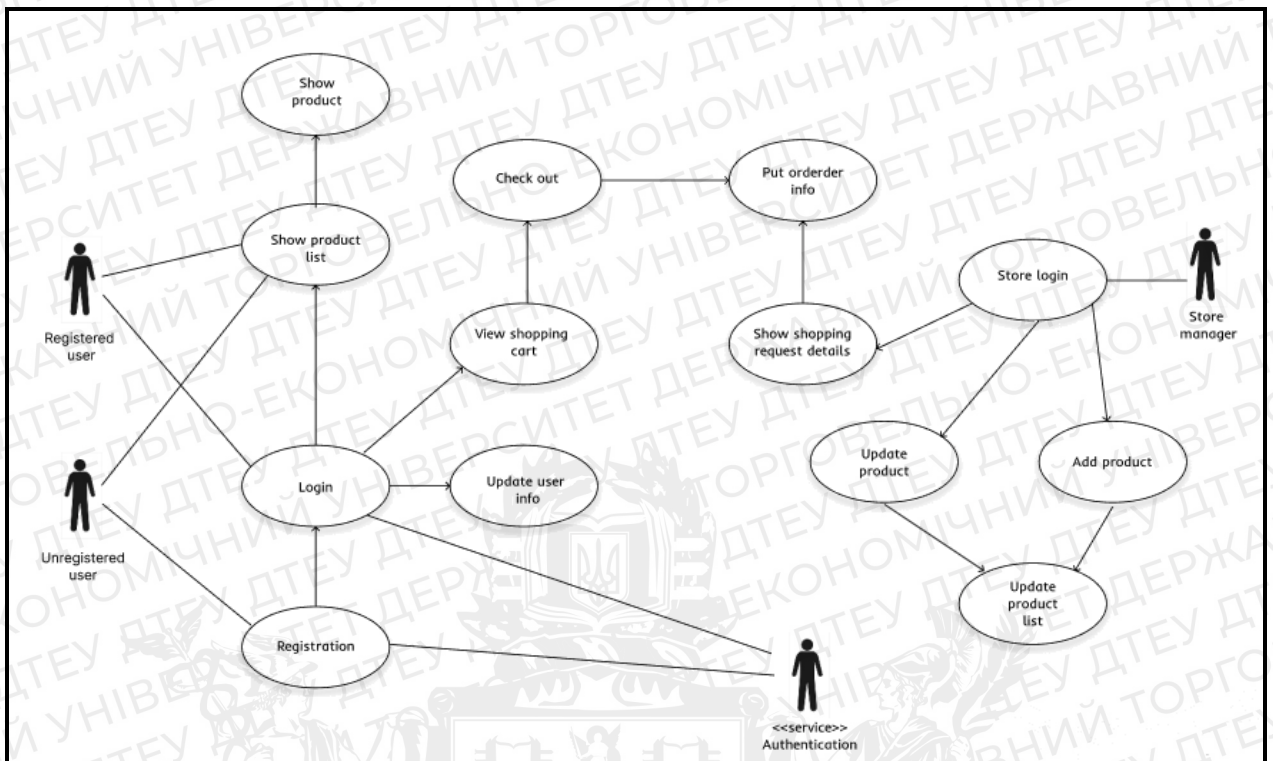


Рисунок 1. Use case діаграма

1.2. Еталонна модель

На зображенні, що позначене як Рисунок 2, зображена типова модель веб-сайту. Ця модель демонструє архітектурні особливості, пов'язані з проектуванням та розробкою веб-додатків. Користувач взаємодіє з інтерфейсом та логікою, а найбільш важливою є захист даних за допомогою протоколу передачі HTTPS, що використовується з додатковим шаром шифрування та автентифікації між HTTP і TCP. Також веб-сайт повинен мати сервер з налаштованою базою даних та SSL-сертифікатом, що забезпечує безпечне з'єднання між клієнтом та сервером за допомогою криптографічного протоколу SSL

| | | | | | | |
|-----|-------|---------|--------|------|-------------------|-------|
| | | | | | | Аркуш |
| | | | | | ДТЕУ 121 07-19.БР | 18 |
| Зм. | Аркуш | № докум | Підпис | Дата | | |

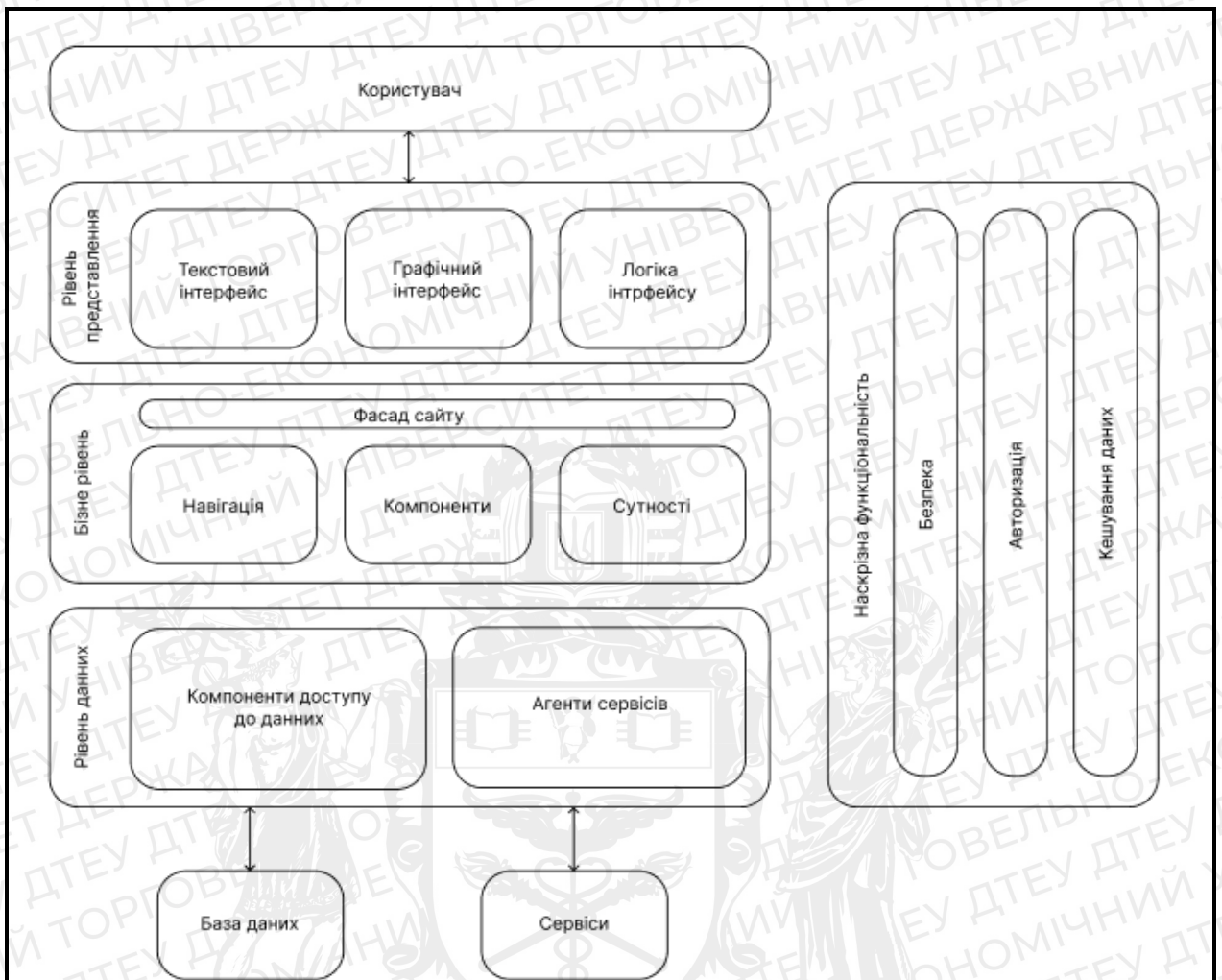


Рисунок 2. Еталонна модель веб сайту

Еталонна модель веб-сайту визначає стандарти якості для веб-сайту, такі як швидкість завантаження сторінок, частота відвідування, рівень задоволення користувачів тощо. Це допомагає визначити, наскільки добре веб-сайт працює в порівнянні з конкурентами і зробити відповідні вдосконалення. Також, веб-сайт можна оптимізувати, використовуючи еталонну модель. Наприклад, можна змінити розташування рекламних блоків на веб-сайті або змінити зображення, щоб забезпечити кращу привабливість веб-сайту для відвідувачів.

1.3. Блок схема



Рисунок 3. Блок схема веб-сайту

На рисунку Рис. 3 зображено блок схему веб-сайту. Блок-схема - це графічне зображення послідовності кроків або процесу, яке використовується для подання алгоритму. Блок схема клієнт-серверного додатку складається із початку, кінця та алгоритму за яким користувачі будуть робити замовлення.

Блок-схема допомагає визначити логічну послідовність переходу від однієї сторінки до іншої та встановити ієрархію сторінок. Також, вона

| | | | | | | | |
|-----|-------|---------|--------|------|--|-------------------|-------|
| | | | | | | ДТЕУ 121 07-19.БР | Аркуш |
| | | | | | | | 20 |
| Зм. | Аркуш | № докум | Підпис | Дата | | | |

дозволяє розробникам більш чітко розуміти функціональні вимоги до веб-сайту та планувати необхідні ресурси для його розробки.

1.4. Діаграма послідовності

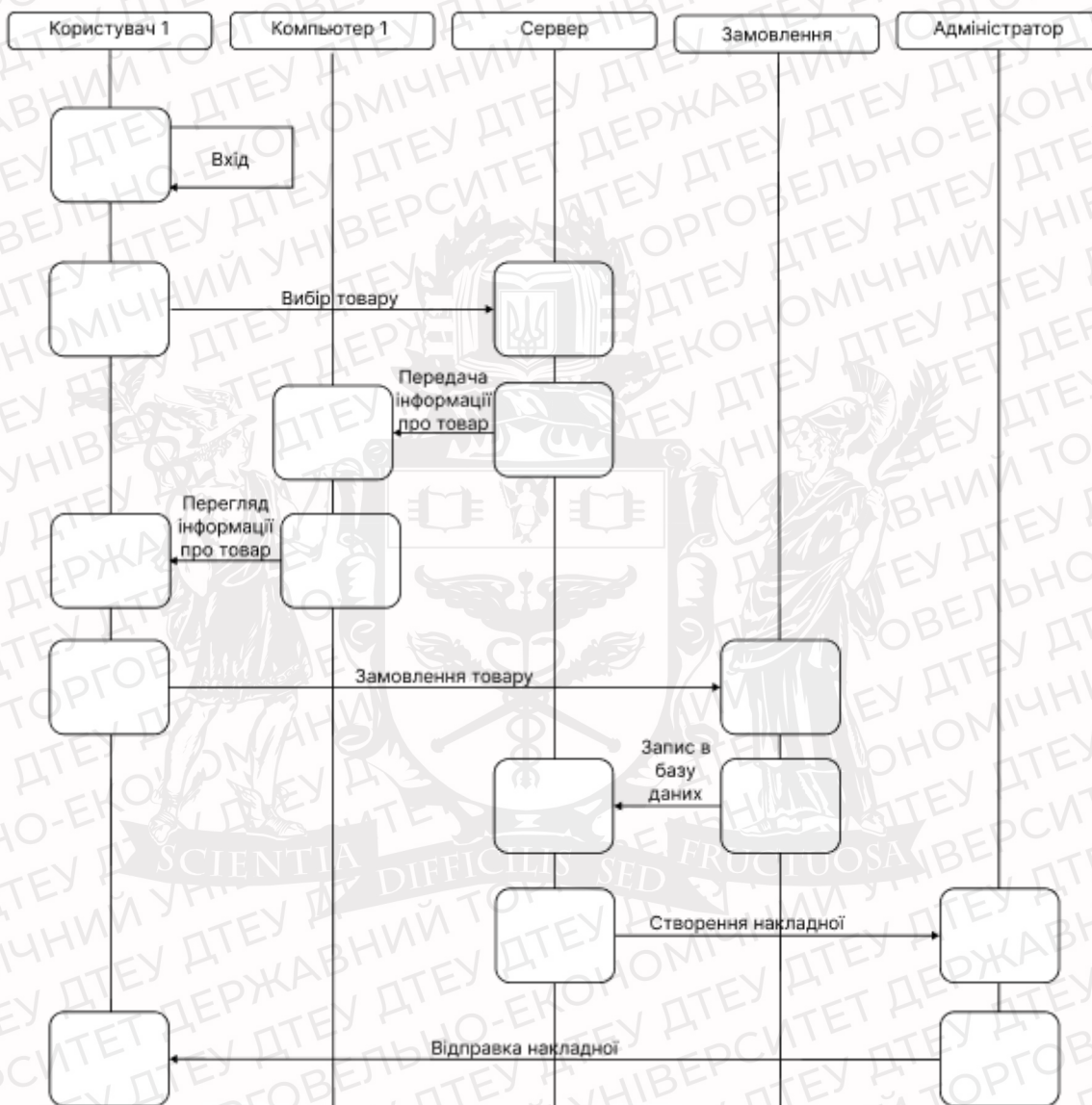


Рисунок 4. Діаграма послідовності

На рисунку 4 зображена діаграма послідовності. Вона допомагає розуміти, які дії можуть відбуватися на веб-сайті та в якому порядку. Діаграма послідовності візуалізує процеси, які відбуваються на веб-сайті, і показує, кроки потрібні для досягнення певної мети.

1.5. Діаграма стану



Рисунок 5 Діаграма стану замовлення

Діаграма стану веб-сайту є корисним інструментом для планування та проектування веб-сайту. Вона дозволяє візуалізувати структуру веб-сайту та взаємозв'язки між його різними частинами.

Діаграма стану відображує структуру веб-сайту в логічному та ієрархічному порядку, що допомагає організувати контент та навігацію.

Також вона дозволяє виявити пропущені сторінки або розділи веб-

| | | | | | | | |
|-----|-------|---------|--------|------|--|-------------------|-------|
| | | | | | | ДТЕУ 121 07-19.БР | Аркуш |
| Зм. | Аркуш | № докум | Підпис | Дата | | | 22 |

Також вона дозволяє виявити пропущені сторінки або розділи веб-сайту. Це допомагає забезпечити повноту та коректність контенту.

На рисунку 5 зображено діаграму стану замовлення, вона наглядно показує логіку та послідовність дій, які користувач повинен мати можливість зробити, якщо вирішиться зробити замовлення.

1.6. Діаграма класів

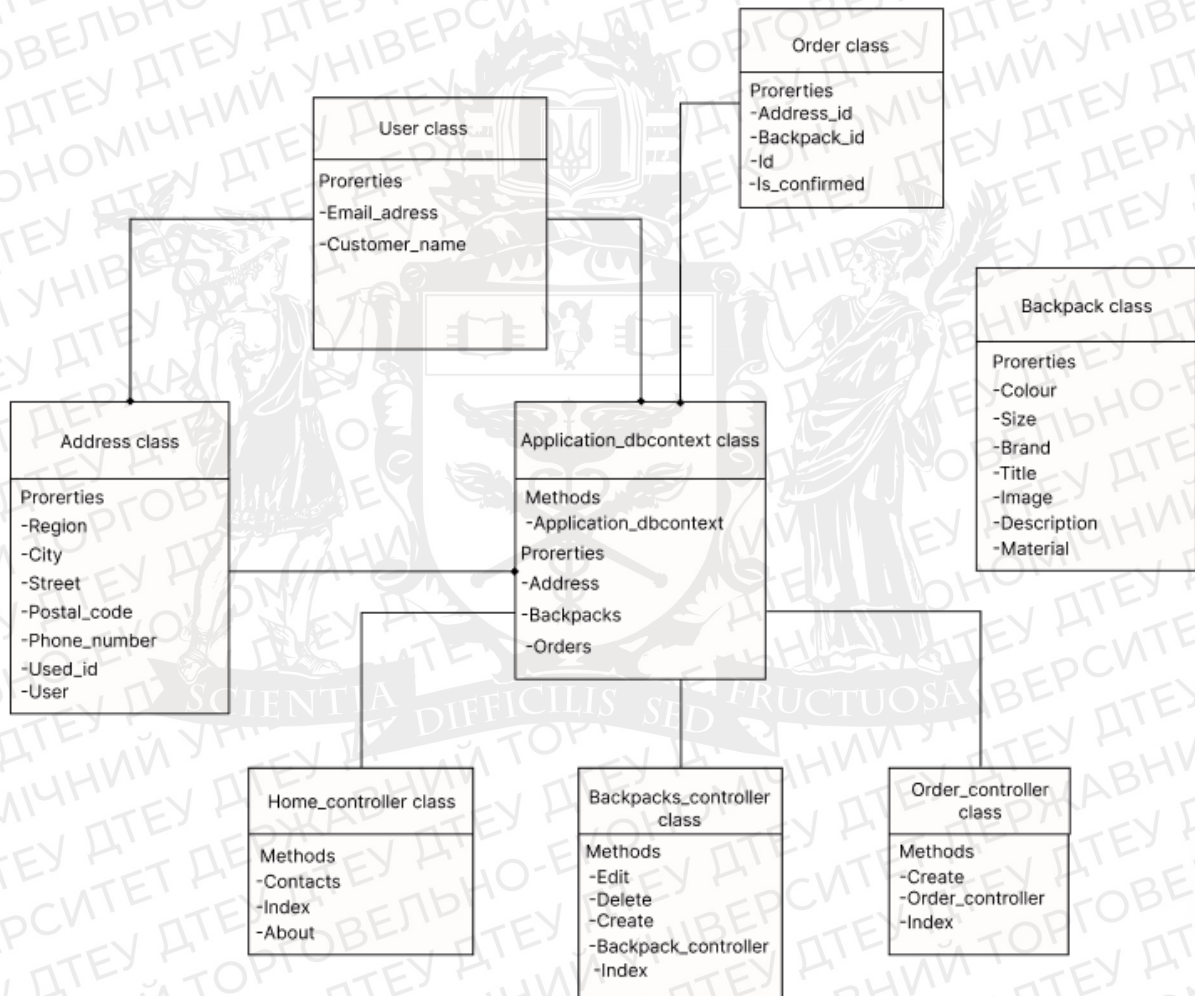


Рисунок 6. Діаграма класів

На рисунку Рис. 6 зображено діаграму класів. Діаграма класів - це вид діаграми в UML, який представляє структуру класів, їх атрибутів та методів, а також взаємозв'язки між ними. Вона служить для представлення статичної структури моделі системи на основі класів об'єктно-орієнтованого

програмування. На ній зображають класи, об'єкти, інтерфейси, кооперації із зазначенням відношень між ними. Зв'язки між класами показуються лініями з різними знаками та стрілками, які вказують на тип взаємозв'язку.

1.7. Діаграма розгортання

На рисунку Рис. 7 зображено діаграму розгортання.

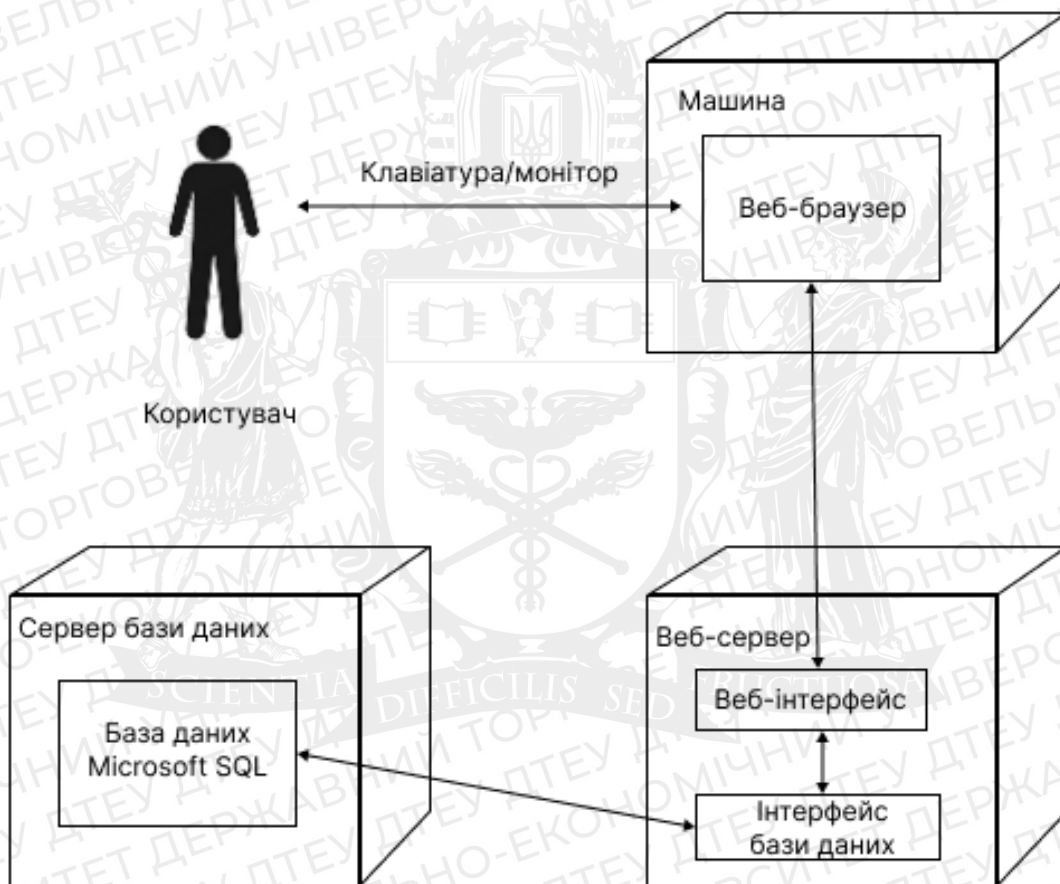


Рисунок 7. Діаграма розгортання

Діаграма розгортання веб-сайту є важливим інструментом для планування та розгортання веб-додатку або веб-сайту. Вона допомагає визначити, які компоненти системи повинні бути розгорнуті на сервері та як вони будуть взаємодіяти між собою. Вона допомагає визначити, які сервери

бази даних та інші компоненти повинні бути розгорнуті для забезпечення стабільної та безперебійної роботи веб-сайту. Також, діаграма розгортання веб-сайту дозволяє легко відслідковувати конфігурацію компонентів системи, визначити їх властивості та взаємодію між ними

2. Бази даних

База даних - це структуроване збереження інформації, яке дозволяє збирати, організувати, зберігати і отримувати доступ до даних. Вона використовується для ефективного управління великим обсягом інформації і дозволяє виконувати різні операції з даними, такі як додавання, редагування, вилучення та пошук.

2.1 Логічна модель бази даних

Логічна модель бази даних є концептуальним відображенням структури бази даних, яке використовує спеціальні поняття, які дозволяють описати дані та взаємозв'язки між ними. Це абстрактний рівень моделі, який не залежить від конкретної системи управління базами даних .

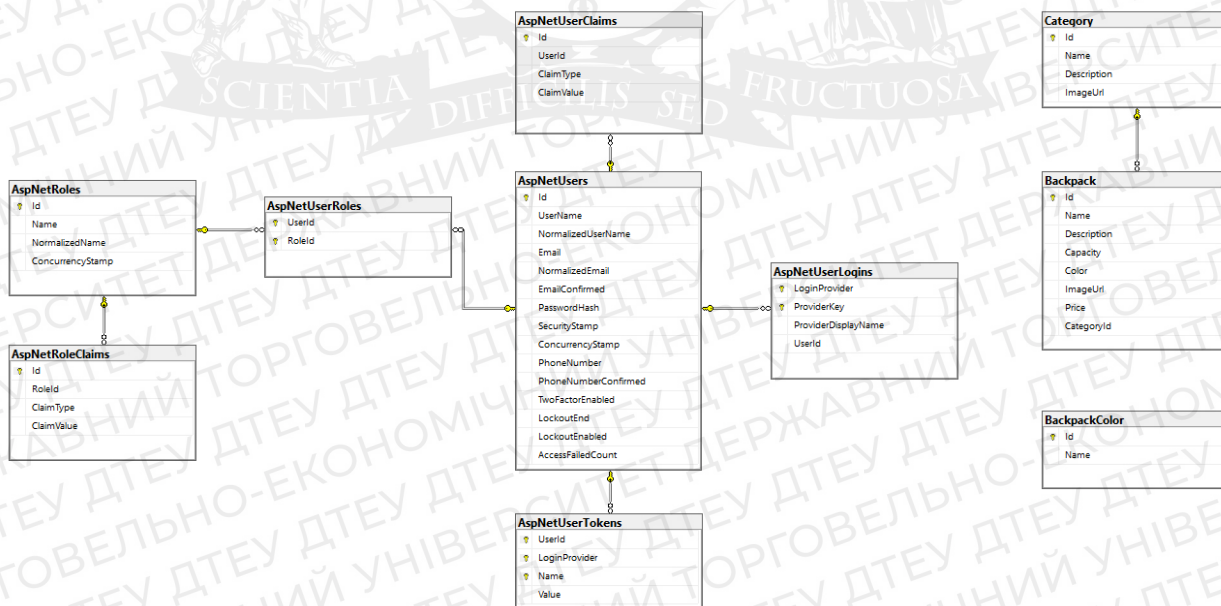


Рисунок 8. Логічна модель бази даних

2.2 Фізична модель бази даних

Фізична модель бази даних є важливою складовою веб-сайту з продажу рюкзаків. Ця модель визначає структуру і організацію даних, що використовуються для збереження і керування інформацією про продукти, замовлення, користувачів та інше.

Одним з ключових елементів фізичної моделі є таблиці бази даних, які представляють різні сутності, такі як продукти, замовлення, клієнти тощо. Наприклад, може бути створена таблиця "Продукти" з колонками, що включають назву продукту, опис, ціну, кількість на складі тощо. Такі таблиці використовуються для збереження даних про рюкзаки, які продаються на сайті.

Окрім цього, фізична модель включає зв'язки між таблицями, що дозволяють пов'язувати дані між різними сутностями. Наприклад, може бути створений зв'язок між таблицями "Замовлення" і "Продукти", щоб відстежувати, які продукти були замовлені кожним клієнтом. Це дозволяє зручно керувати замовленнями і знаходити відповідні продукти для кожного замовлення. Крім основних таблиць і зв'язків, фізична модель також враховує індекси для оптимізації швидкості пошуку та запитів до бази даних. Індекси дозволяють швидко знаходити потрібні записи за конкретними полями, наприклад, індекс за назвою продукту для швидкого пошуку конкретного рюкзака.

Таблиця 1
AspNetUsers

| Назва поля | Ключ | Індексоване поле | Тип даних | Null-значення |
|------------|------|------------------|---------------|---------------|
| Id | PK | Так | nvarchar(450) | Не допустимі |

| | | | | | | | |
|-----|-------|---------|--------|------|-------------------|--|-------|
| | | | | | | | Аркуш |
| | | | | | | | 26 |
| Зм. | Аркуш | № докум | Підпис | Дата | ДТЕУ 121 07-19.БР | | |

| | | | | |
|----------------------|--|--|-------------------|--------------|
| UserName | | | nvarchar(256) | |
| NormalizedUserName | | | nvarchar(256) | |
| Email | | | nvarchar(256) | Не допустимі |
| NormalizedEmail | | | nvarchar(256) | |
| EmailConfirmed | | | Bit | Не допустимі |
| PasswordHash | | | nvarchar(MAX) | |
| SecurityStamp | | | nvarchar(MAX) | |
| ConcurrencyStamp | | | nvarchar(MAX) | |
| PhoneNumber | | | nvarchar(MAX) | Не допустимі |
| PhoneNumberConfirmed | | | Bit | |
| TwoFactorEnabled | | | Bit | Не допустимі |
| LockoutEnd | | | datetimeoffset(7) | |
| LockoutEnabled | | | Bit | |
| AccessFailedCount | | | Int | |

| | | | | | | | | | |
|-----|-------|---------|--------|------|--------------------------|--|--|--|-------|
| | | | | | | | | | Аркуш |
| | | | | | | | | | 27 |
| Зм. | Аркуш | № докум | Підпис | Дата | <i>ДТЕУ 121 07-19.БР</i> | | | | |

Таблиця 2

Address

| Назва поля | Ключ | Індексоване поле | Тип даних | Null-значення |
|-------------|------|------------------|---------------|---------------|
| Id | PK | Так | Int | Не допустимі |
| UserId | FK | | nvarchar(450) | |
| Region | | | nvarchar(MAX) | |
| City | | | nvarchar(MAX) | |
| Street | | | nvarchar(MAX) | |
| PhoneNumber | | | nvarchar(MAX) | |
| PostalCode | | | nvarchar(MAX) | |

| | | | | | | | | | |
|-----|-------|---------|--------|------|--|--|--|--|-------------------|
| | | | | | | | | | Аркуш |
| | | | | | | | | | ДТЕУ 121 07-19.БР |
| Зм. | Аркуш | № докум | Підпис | Дата | | | | | 28 |

Таблиця 3

Orders

| Назва поля | Ключ | Індексоване поле | Тип даних | Null-значення |
|------------|------|------------------|-----------|---------------|
| Id | PK | Так | Int | Не допустимі |
| BackpackId | FK | | Int | Не допустимі |
| AddressId | FK | | Int | Не допустимі |

Таблиця 4

Backpacks

| Назва поля | Ключ | Індексоване поле | Тип даних | Null-значення |
|-------------|------|------------------|----------------|---------------|
| Id | PK | Так | Int | Не допустимі |
| Id | | | Int | |
| Name | | | nvarchar(MAX) | |
| Description | | | nvarchar(MAX) | |
| Capacity | | | Int | |
| Color | | | nvarchar(MAX) | |
| ImageUrl | | | nvarchar(MAX) | |
| Price | | | decimal(18, 2) | |

| | | | | | | | |
|-----|-------|---------|--------|------|-------------------|--|-------|
| | | | | | | | Аркуш |
| | | | | | | | 29 |
| Зм. | Аркуш | № докум | Підпис | Дата | ДТЕУ 121 07-19.БР | | |

Таблиця 5
AspNetUserTokens

| Назва поля | Ключ | Індексоване поле | Тип даних | Null-значення |
|---------------|------|------------------|---------------|---------------|
| UserId | FK | | nvarchar(450) | Не допустимі |
| LoginProvider | | | nvarchar(128) | |
| Name | | | nvarchar(128) | |
| Value | | | nvarchar(MAX) | |

Таблиця 6
AspNetUserLogins

| Назва поля | Ключ | Індексоване поле | Тип даних | Null-значення |
|---------------------|------|------------------|---------------|---------------|
| LoginProvider | | | nvarchar(128) | Не допустимі |
| ProviderKey | | | nvarchar(128) | Не допустимі |
| ProviderDisplayName | | | nvarchar(MAX) | |
| UserId | FK | Так | nvarchar(450) | Не допустимі |

Таблиця 7
AspNetUserClaims

| Назва поля | Ключ | Індексоване поле | Тип даних | Null-значення |
|------------|------|------------------|---------------|---------------|
| Id | PK | Так | Int | Не допустимі |
| UserId | FK | Так | nvarchar(450) | Не допустимі |
| ClaimType | | | nvarchar(MAX) | |
| ClaimValue | | | nvarchar(MAX) | |

Таблиця 8
AspNetUserRoles

| Назва поля | Ключ | Індексоване поле | Тип даних | Null-значення |
|------------|------|------------------|---------------|---------------|
| Id | FK | | Int | Не допустимі |
| RoleId | | | nvarchar(450) | Не допустимі |
| ClaimType | | | nvarchar(MAX) | |
| ClaimValue | | | nvarchar(MAX) | |

Таблиця 11

Category

| Назва поля | Ключ | Індексоване поле | Тип даних | Null-значення |
|-------------|------|------------------|---------------|---------------|
| Id | PK | Так | Int | Не допустимі |
| Name | FK | Так | nvarchar(MAX) | Не допустимі |
| Description | | | nvarchar(MAX) | |
| ImageUrl | | | nvarchar(MAX) | |

Висновки до розділу 2

У розділі "Архітектура додатку" було представлено декілька діаграм та моделей, які допомагають у розумінні та проектуванні архітектури веб-сайту.

У Use-case діаграмі було описано різні ролі користувачів, такі як не зареєстрований користувач, зареєстрований користувач та адміністратор сайту, а також їх можливі взаємодії з системою.

Еталонна модель веб-сайту була використана для визначення стандартів якості, що дозволило порівняти додаток з конкурентами та внести відповідні вдосконалення.

Блок-схема веб-сайту була використана для визначення послідовності переходу між сторінками та встановлення ієрархії сторінок.

Діаграма послідовності була використана для візуалізації процесів, що відбуваються на веб-сайті, та визначення кроків, необхідних для досягнення певної мети.

Діаграма стану була використана для візуалізації структури веб-сайту

| | | | | | | | |
|-----|-------|---------|--------|------|--|-------------------|-------|
| | | | | | | ДТЕУ 121 07-19.БР | Аркуш |
| | | | | | | | 33 |
| Зм. | Аркуш | № докум | Підпис | Дата | | | |

та взаємозв'язків між його частинами, а також для організації контенту та навігації.

Діаграма класів була використана для представлення структури класів, інтерфейсів, об'єктів та їх взаємозв'язків.

Діаграма розгортання була використана для планування та розгортання компонентів системи на сервері та визначення їх взаємодії.

У розділі «Бази даних» було розглянуто бази даних, як організовані колекції даних, призначені для ефективного доступу до інформації.

Описано логічну модель бази даних, яка є концептуальним відображенням структури бази даних. Це абстрактний рівень моделі, який не залежить від конкретної системи управління базами даних.

Також було представлено фізичну модель бази даних, яка відображає фактичну реалізацію системи бази даних. Фізична модель фокусується на фізичних аспектах структури бази даних, включаючи зберігання та організацію даних на пристроях зберігання.

У розділі наведено приклади таблиць з їхніми полями та типами даних, що використовуються для організації бази даних. Також вказано зв'язки між таблицями за допомогою первинних та зовнішніх ключів.

Фізична модель даних містить необхідні обмеження, такі як тригери та збережені процедури, що забезпечують цілісність та безпеку даних.

| | | | | | | | |
|-----|-------|---------|--------|------|--|-------------------|-------|
| | | | | | | ДТЕУ 121 07-19.БР | Аркуш |
| | | | | | | | 34 |
| Зм. | Аркуш | № докум | Підпис | Дата | | | |

РОЗДІЛ 3 РЕАЛІЗАЦІЯ КЛІЄНТ-СЕРВІСНОГО ДОДАТКУ

1. Встановлення Visual Studio

Visual Studio є інтегрованим середовищем розробки (IDE) для створення програмного забезпечення. Він має багатofункціональний редактор коду з підсвічуванням синтаксису та автоматичним завершенням коду.

Процес встановлення Visual Studio 2022 на Windows зазвичай включає наступні кроки:

Спочатку завантажено інсталяційний файл Visual Studio 2022 з офіційного сайту Microsoft.

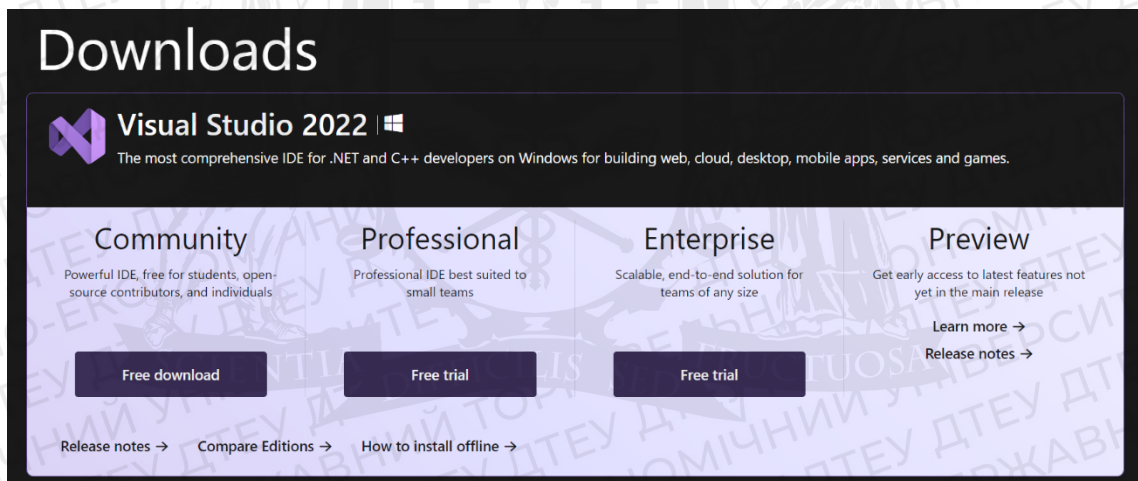


Рисунок 9. Частина веб сайту Visual Studio Installer

Після завантаження запущено інсталяційний файл.

При запуску з'явилось вікно встановлення Visual Studio 2022. На початку мене запитали про згоду з ліцензійними умовами Microsoft. прочитавши їх треба погодитись з ними.

| Зм. | Аркуш | № докум. | Підпис | Дата | ДТЕУ 121 07-19.БР | | | |
|-----------|------------------|----------|--------|----------|--|---------------------------------------|-------|---------|
| Зав. каф. | Криворучко О.В. | | | 10.01.23 | Клієнт-серверний додаток реалізації наплічників | Стадія | Аркуш | Аркушів |
| Керівник | Власенко Л. О. | | | 10.01.23 | | РЗ | 35 | 53 |
| Гарант | Рзаєва С.Л. | | | 10.01.23 | | Факультет інформаційних технологій | | |
| Розробив | Пересацько О. О. | | | 10.01.23 | | 4 курс, 7 група | | |
| | | | | | Розділ 3 | | | |

Далі мені було запропоновано вибрати набір компонентів, які треба встановити. Visual Studio 2022 надавала різні опції для розробки різних типів програм, обрано лише ті, що потрібні мені.

Після вибору компонентів мені запропонували встановити додаткові компоненти або інструменти, які покращують роботу з Visual Studio. Обрано необхідні додатки та інструменти.

Після встановлення обрано параметри, пов'язані з використанням телеметрії та повідомлень про помилки.

Натиснуто кнопку "Install" (Встановити), щоб розпочати процес встановлення Visual Studio 2022.

Після успішного встановлення запустила Visual Studio 2022 та розпочала розробку веб-сайту за допомогою цього середовища.

2. Встановлення SQL Server Express

Для роботи з базами даних було завантажено SQL Server Express, що є безкоштовною версією. Для завантаження SQL Server Express були виконані наступні кроки:

1. Було відкрито браузер і перейшли на веб-сторінку завантаження SQL Server Express на офіційному веб-сайті Microsoft.

Microsoft® SQL Server® 2019 Express

Important! Selecting a language below will dynamically change the complete page content to that language.

Select Language:

English

Download

Microsoft® SQL Server® 2019 Express is a powerful and reliable free data management system that delivers a rich and reliable data store for lightweight Web Sites and desktop applications.

Рисунок 10. Частина сторінки з вибором мови та кнопкою «Завантажити»

| | | | | | |
|-----|-------|---------|--------|------|-------|
| | | | | | Аркуш |
| | | | | | 36 |
| Зм. | Аркуш | № докум | Підпис | Дата | |

ДТЕУ 121 07-19.БР

2. На сторінці завантаження була обрана бажана версія SQL Server Express.

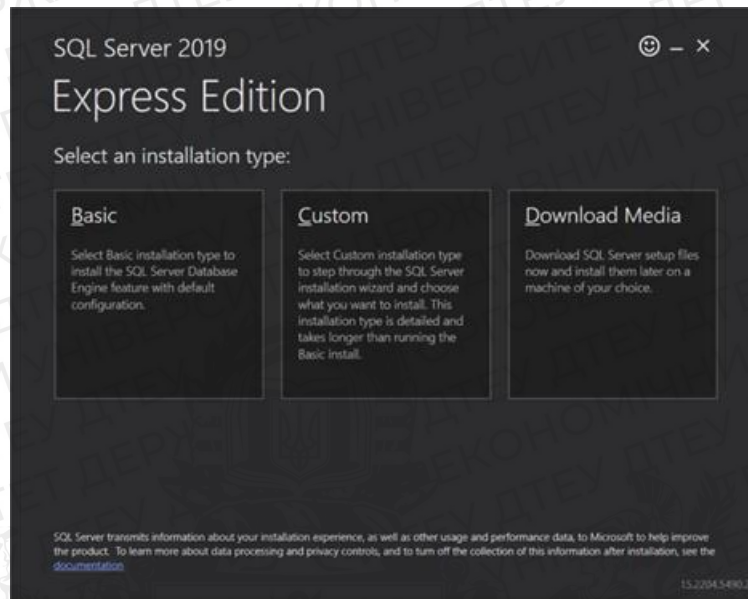


Рисунок 11. Початкова сторінка SQL Server 2019

3. Було натиснуто посилання для завантаження обраної версії SQL Server Express, що спричинило початок завантаження виконуваного файлу інсталяції.

4. Після завершення завантаження було відкрито виконуваний файл інсталяції, це був файл з розширенням ".exe".

5. Було уважно прочитано та прийнято ліцензійну угоду.

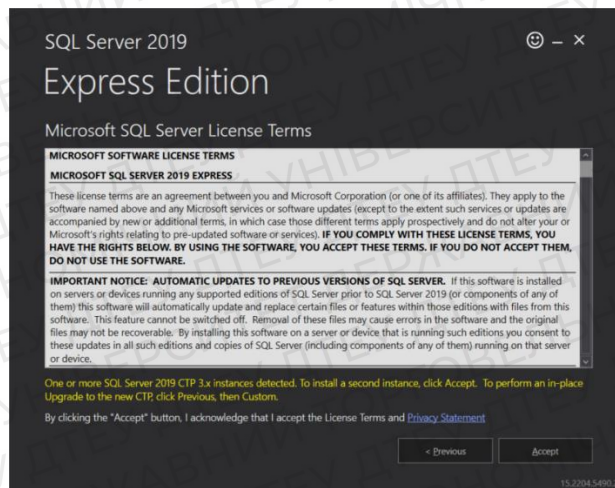


Рисунок 12. Прийняття ліцензійних угод SQL Server 2019 Express

| | | | | | |
|-----|-------|---------|--------|------|-------|
| | | | | | Аркуш |
| | | | | | 37 |
| Зм. | Аркуш | № докум | Підпис | Дата | |

ДТЕУ 121 07-19.БР

6. На екрані було обрано тип встановлення, залишаючи його за замовчуванням.

7. Були обрані компоненти для встановлення, залишаючи параметри за замовчуванням, якщо не було специфічних потреб.

8. Був обраний шлях для встановлення SQL Server Express, використовуючи рекомендований шлях за замовчуванням.

9. Було вказано ім'я інстанції SQL Server, а також можливо було обрати режим аутентифікації, якщо потрібно було використовувати інший режим.

10. Були налаштовані ролі та права доступу до сервера SQL Server, залишаючи параметри за замовчуванням, якщо не було певних потреб.

Тепер наявний успішно встановлений SQL Server Express на комп'ютері. Можна відкрити SQL Server Management Studio або будь-який інший інструмент для адміністрування SQL Server, щоб працювати з базами даних.

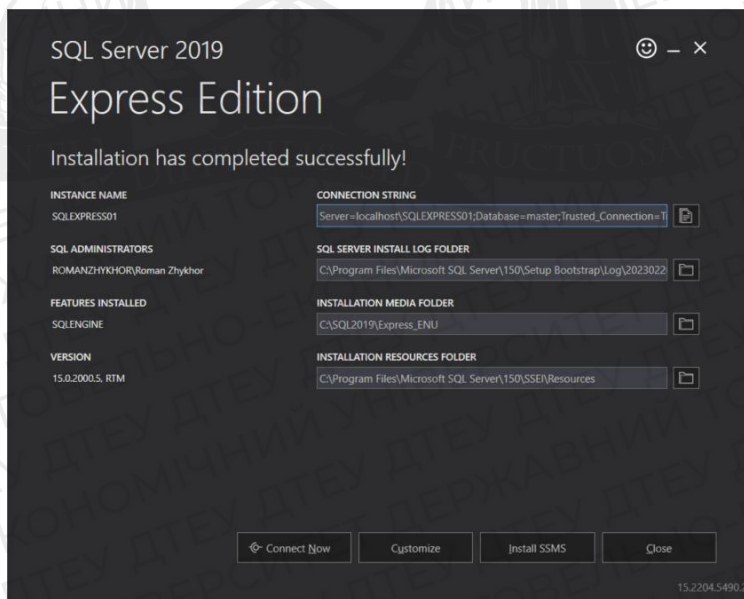


Рисунок 13. Вікно завершення встановлення SQL Server 2019 Express

| | | | | | |
|-----|-------|---------|--------|------|-------|
| | | | | | Аркуш |
| | | | | | 38 |
| Зм. | Аркуш | № докум | Підпис | Дата | |

ДТЕУ 121 07-19.БР

3. Створення проекту Switch Book

В Visual Studio обрано шаблон із заготовленим кодом, це допомагає швидко почати створення логіки програмних продуктів. Для цього було зроблено наступні кроки:

1. На головному екрані Visual Studio натиснуто кнопку "Створити новий проект".
2. Відкрилось вікно з назвою "Створення проекту", і в ньому обрано шаблон "ASP.NET Core Web App (MVC)".
3. Названо, встановлені параметри та обрано місце збереження проекту. Далі натиснуто кнопку «Створити».

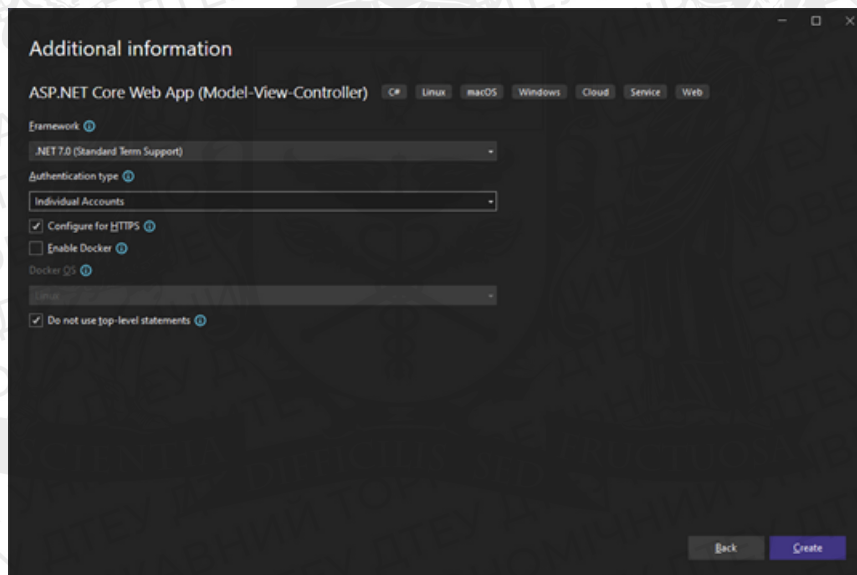


Рисунок 14. Створення проекту «SwitchBook»

Таким чином було створено проект «SwitchBook».

4. Підключення бази даних до проекту та налаштування Identity

Framework

Наступним кроком додано базу даних SQL Server Express до проекту використовуючи Entity Framework в Visual Studio, це зроблено наступним чином:

| | | | | | |
|-----|-------|---------|--------|------|-------|
| | | | | | Аркуш |
| | | | | | 39 |
| Зм. | Аркуш | № докум | Підпис | Дата | |

ДТЕУ 121 07-19.БР

1. В корені проекту знаходиться конфігураційний файл, в який був доданий рядок з'єднання. Названо його «DefaultConnection».

```
"ConnectionStrings": {  
  "DefaultConnection": "Server=DESKTOP-4ITITFF\\SQLEXPRESS;Database=BackpackShop.WebApplication;Trusted_Connection=True;MultipleActiveResultSets=true"  
},
```

Рисунок 15. Рядок з'єднання з базою даних

2. У папці «Data» створено файл контексту даних

```
16 references  
public class ApplicationDbContext : IdentityDbContext  
{  
  0 references  
  public ApplicationDbContext(DbContextOptions<ApplicationDbContext> options)  
    : base(options)  
  {  
  }  
  16 references  
  public DbSet<Models.Backpack> Backpack { get; set; }  
  25 references  
  public DbSet<Models.Category> Category { get; set; }  
}
```

Рисунок 16. Код контексту

3. Додано до проекту сервіси Entity Framework та Identity Framework.

```
// Add services to the container.  
var connectionString = builder.Configuration.GetConnectionString("DefaultConnection") ?? throw new InvalidOperationException("Connection string 'DefaultConnection' is not found.");  
builder.Services.AddDbContext<ApplicationDbContext>(options =>  
  options.UseSqlServer(connectionString));  
builder.Services.AddDatabaseDeveloperPageExceptionFilter();  
  
builder.Services.AddDefaultIdentity<IdentityUser>(options => options.SignIn.RequireConfirmedAccount = true)  
  .AddRoles<IdentityRole>(C)  
  .AddEntityFrameworkStores<ApplicationDbContext>(C);  
builder.Services.AddControllersWithViews();
```

Рисунок 17. Налаштування сервісів Entity та Identity.

5. Створення моделей даних, бази даних і таблиць

Обрано метод Code First для створення моделей даних

Моделі даних було створено наступним чином:

1. Першим створеним класом був клас Backpack, який включає такі моделі даних як:

| | | | | | | |
|-----|-------|---------|--------|------|-------------------|-------|
| | | | | | ДТЕУ 121 07-19.БР | Аркуш |
| | | | | | | 40 |
| Зм. | Аркуш | № докум | Підпис | Дата | | |

- Id: цілочисельне значення, що ідентифікує унікальний ідентифікатор рюкзака.
- Name: рядок, що містить назву рюкзака.
- Description: рядок, що містить опис рюкзака.
- Capacity: цілочисельне значення, що вказує вмістимість рюкзака.
- Color: рядок, що вказує колір рюкзака.
- ImageUrl: рядок, що містить URL-адресу зображення рюкзака.
- Price: десяткове значення, що вказує ціну рюкзака.
- CategoryId: цілочисельне значення, що ідентифікує категорію, до якої належить рюкзак.

• Category: посилання на об'єкт класу Category (категорія), до якої належить рюкзак.

```

19 references
public class Backpack
{
    13 references
    public int Id { get; set; }
    16 references
    public string Name { get; set; }
    13 references
    public string Description { get; set; }
    12 references
    public int Capacity { get; set; }
    10 references
    public string Color { get; set; }
    13 references
    public string ImageUrl { get; set; }
    14 references
    public decimal Price { get; set; }
    10 references
    public int CategoryId { get; set; }
    8 references
    public Category Category { get; set; }
}

```

Рисунок 18. Клас Backpack.

2. Наступним став клас Category, який містить наступні властивості:

- Id - цілочисельна властивість, що представляє ідентифікатор категорії.
- Name - рядкова властивість, що містить назву категорії.
- Description - рядкова властивість, що містить опис категорії.
- ImageUrl - рядкова властивість, що містить URL-адресу зображення для категорії.
- Backpacks - перерахування (IEnumerable) об'єктів типу Backpack, яке представляє рюкзаки, що належать до цієї категорії.

```
16 references
public class Category
{
    19 references
    public int Id { get; set; }
    20 references
    public string Name { get; set; }
    13 references
    public string Description { get; set; }
    12 references
    public string ImageUrl { get; set; }
    0 references
    public IEnumerable<Backpack> Backpacks { get; set; }
}
```

Рисунок 19. Клас Category.

Після створення всіх класів-моделей в контексті бази даних, було встановлено DbSet для кожного з них, щоб відтворити таблиці бази даних.

Далі створено клас ErrorViewModel, який використовується для представлення помилок у веб-додатку або на сторінці з помилками.

```

4 references
public class ErrorViewModel
{
    3 references
    public string? RequestId { get; set; }

    1 reference
    public bool ShowRequestId => !string.IsNullOrEmpty(RequestId);
}

```

Рисунок 20. Клас ErrorViewModel.

6. Створення контролерів

Для управління логікою веб-сайту були створені контролери, виконано наступні кроки:

1. Натиснуто правою кнопкою миші на папку "Controllers" та вибрано опцію "Add" → "Controller".

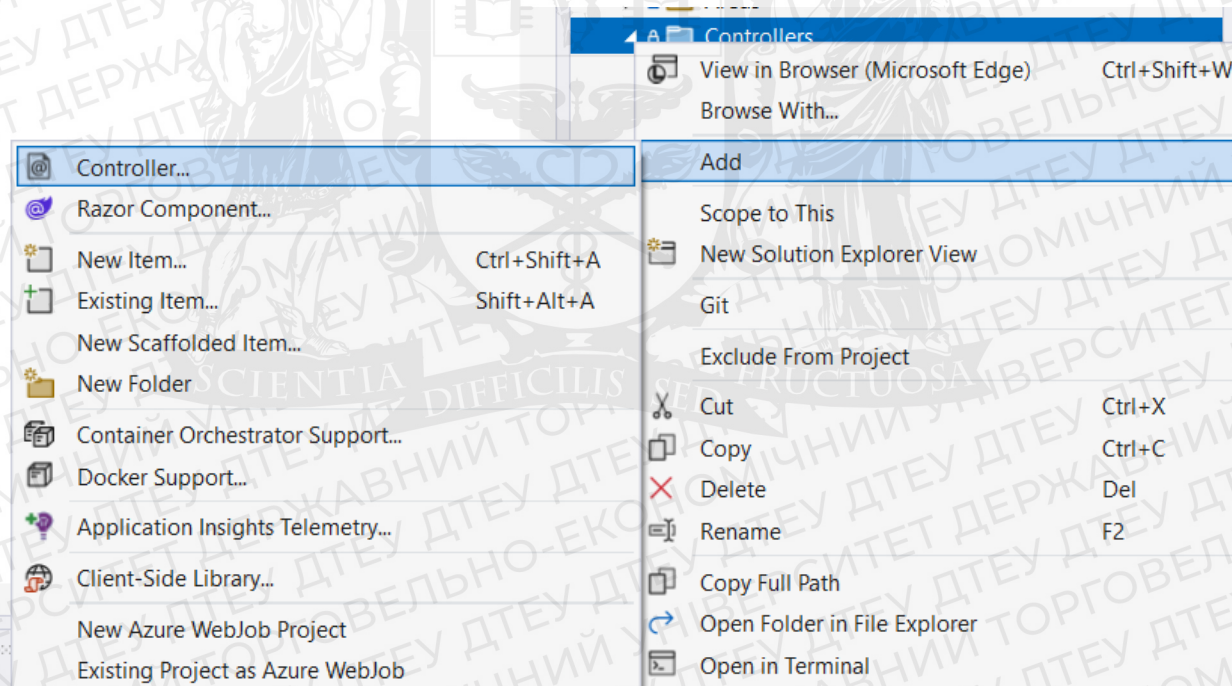


Рисунок 21. Додавання контролера через контекстне меню.

2. Обрано опцію "MVC Controller - Empty", а потім натиснуто кнопку "Add".

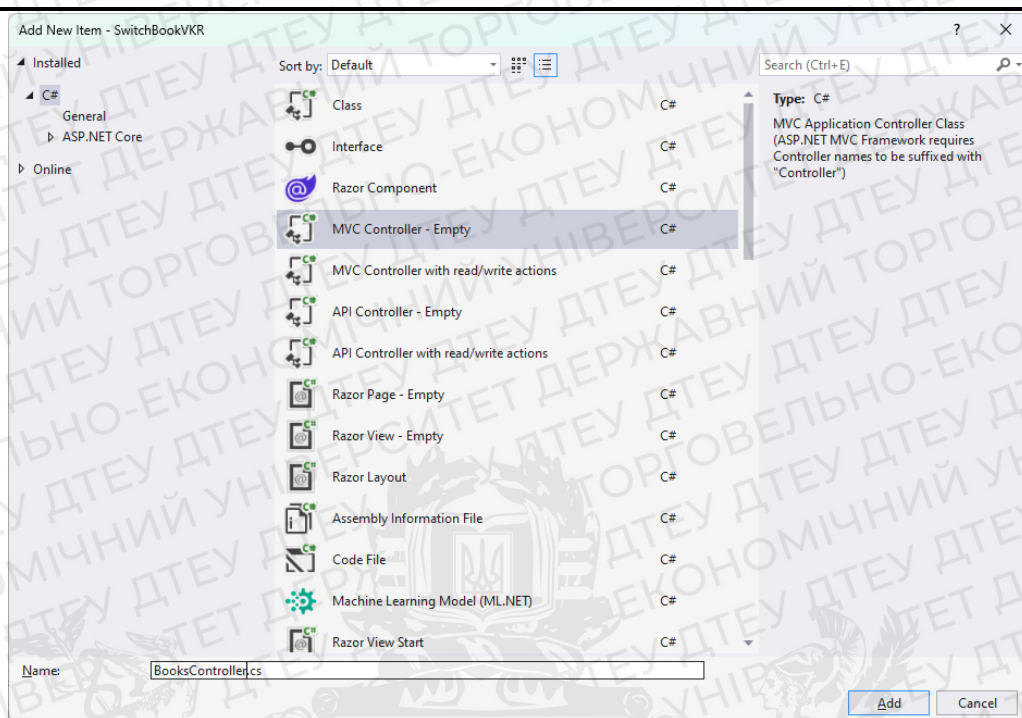


Рисунок 22. Клас контексту даних після зроблених кроків.

7 Розробка логіки та додавання контексту даних до контролерів

Для забезпечення контексту даних для виконання операцій з даними у таблицях бази даних було розроблено конструктор, який отримує клас контексту даних як параметр і ініціалізує приватну змінну в рамках цього конструктора.

```

1 reference
public class BackpacksController : Controller
{
    private readonly ApplicationDbContext _context;

0 references
    public BackpacksController(ApplicationDbContext context)
    {
        _context = context;
    }
}

```

Рисунок 23. Конструктор та змінна у конструкторі контролера «BooksController».

| | | | | | |
|-----|-------|---------|--------|------|-------|
| | | | | | Аркуш |
| | | | | | 44 |
| Зм. | Аркуш | № докум | Підпис | Дата | |

ДТЕУ 121 07-19.БР

Також створено методи додавання нового рюкзака та можливість маніпуляцій з наявними.

```
0 references
public async Task<IActionResult> Create([Bind("Id,Name,Description,Capacity,Color,Image,Price,CategoryId")] BackpackOperationViewModel backpack)
{
    if (!ModelState.IsValid) return View(backpack);

    var lastItemId = _context.Backpack.Count() + 1;

    var extension = Path.GetExtension(backpack.Image.FileName);
    if (extension != ".jpg" && extension != ".png" && extension != ".jpeg" && extension != ".gif" && extension != ".bmp" && extension != ".t")
        throw new Exception("Image must be jpg or png format");

    var filename = "Backpack" + lastItemId + extension;
    var combine = Path.Combine(Directory.GetCurrentDirectory(), "wwwroot/img/Backpacks", filename);
    await using var fileStream = new FileStream(combine, FileMode.Create);
    await backpack.Image.CopyToAsync(fileStream);

    var relativePath = "/img/Backpacks/" + filename;

    if (backpack.Image == null)
        relativePath = "/img/defaultBackpack.webp";

    var newBackpack = new Backpack
    {
        Name = backpack.Name,
        Description = backpack.Description,
        ImageUrl = relativePath
    };

    _context.Add(newBackpack);
    await _context.SaveChangesAsync();
    return RedirectToAction(nameof(Index));
}
```

Рисунок 24. Метод що опрацьовує Get запит (завантажує сторінку).

Видалення рюкзака з бази даних робить можливим наступна частина.

```
[Authorize(Roles = "Admin")]
0 references
public async Task<IActionResult> Delete(int? id)
{
    if (id == null || _context.Backpack == null)
    {
        return NotFound();
    }

    var backpack = await _context.Backpack
        .Include(b => b.Category)
        .FirstOrDefaultAsync(m => m.Id == id);
    if (backpack == null)
    {
        return NotFound();
    }

    return View(backpack);
}
```

Рисунок 25. Метод що опрацьовує Get запит (завантажує сторінку).

Далі додано метод "Create" контролера, який обробляє POST-запит на створення нової категорії.

| | | | | | |
|-----|-------|---------|--------|------|-------|
| | | | | | Аркуш |
| | | | | | 45 |
| Зм. | Аркуш | № докум | Підпис | Дата | |

ДТЕУ 121 07-19.БР

```

0 references
public async Task<IActionResult> Create([Bind("Id,Name,Description,Image")] CategoryOperationViewModel category)
{
    if (!ModelState.IsValid) return View(category);

    var lastItemId = _context.Category.Count() + 1;

    var extension = Path.GetExtension(category.Image.FileName);
    if (extension != ".jpg" && extension != ".png" && extension != ".jpeg" && extension != ".gif" && extension != ".bmp")
        throw new Exception("Image must be jpg or png format");

    var filename = "Category" + lastItemId + extension;
    var combine = Path.Combine(Directory.GetCurrentDirectory(), "wwwroot/img/Categories", filename);
    await using FileStream fileStream = new FileStream(combine, FileMode.Create);
    await category.Image.CopyToAsync(fileStream);

    var relativePath = "/img/Categories/" + filename;

    if (category.Image == null)
        relativePath = "/img/defaultCategory.jpg";

    var newCategory = new Category
    {
        Name = category.Name,
        Description = category.Description,
        ImageUrl = relativePath
    };

    _context.Add(newCategory);
    await _context.SaveChangesAsync();
    return RedirectToAction(nameof(Index));
}

```

Рисунок 26. Метод що опрацьовує Create запит (створення категорії)

Висновки до розділу 3

У рамках даної дипломної роботи було розроблено веб-сайт з реалізації наплічників, який має великий потенціал у задоволенні потреб споживачів у пошуку та придбанні якісних і стильних аксесуарів для повсякденного використання. Веб-сайт було створено з використанням сучасних технологій веб-розробки та врахуванням принципів зручності та ефективності користування.

У процесі розробки було проведено аналіз ринку, вивчено основні тенденції у сфері моди та аксесуарів, а також проаналізовано існуючі веб-сайти, які пропонують аналогічні товари. На основі цього аналізу було визначено основні вимоги та функціональні можливості веб-сайту з реалізації наплічників.

В результаті роботи було розроблено зручний та естетично привабливий веб-інтерфейс, який дозволяє користувачам швидко та зручно

| | | | | | |
|-----|-------|---------|--------|------|-------|
| | | | | | Аркуш |
| | | | | | 46 |
| Зм. | Аркуш | № докум | Підпис | Дата | |

ДТЕУ 121 07-19.БР

переглядати асортимент наплічників, здійснювати пошук за різними параметрами та здійснювати онлайн-замовлення.

Процес розробки веб-сайту включав в себе використання сучасних мов програмування та технологій, таких як C#, Visual Studio, ASP.NET, SQL Server, та використання баз даних для зберігання інформації про товари та користувачів.

Веб-сайт з реалізації наплічників має потенціал стати успішним комерційним проектом, оскільки відповідає потребам споживачів у якісних та стильних аксесуарах, а також має сучасний та зручний інтерфейс користувача. Продовження роботи над проектом може включати розширення асортименту товарів, покращення функціональних можливостей та розвиток маркетингових стратегій для залучення більшої кількості клієнтів.

Загалом, розробка веб-сайту з реалізації наплічників була цікавим та важливим процесом, що вимагав використання різноманітних навичок та знань у галузі веб-розробки. Результатом цієї роботи є функціональний та привабливий веб-сайт, який відповідає потребам споживачів та має потенціал для подальшого розвитку та успіху



Рисунок 27. Домашня сторінка.

| | | | | | |
|-----|-------|---------|--------|------|-------------------|
| | | | | | Аркуш |
| | | | | | ДТЕУ 121 07-19.БР |
| Зм. | Аркуш | № докум | Підпис | Дата | 47 |

Наш інтернет-магазин пропонує швидку та зручну доставку в будь-яку точку світу. Тож не вагайтеся - замовляйте рюкзаки у нас і отримуйте задоволення від покупки та використання!

Categories



Backpacks



Рисунок 28. Вибір категорій та товарів.

Hiking Backpack

The product page for the 'Hiking Backpack' features a large yellow backpack illustration in the center. Above it is the 'BackShop' logo. To the right, the product name 'Hiking Backpack' is displayed, followed by a description: 'Hiking Backpack for all ages and sizes'. Below this, the following details are listed: 'Category: Hiking', 'Price: 100,00 UAH', 'Color: Black', and 'Capacity: 30'. A green button labeled 'Contacts to buy!' is positioned at the bottom right of the product information.

Рисунок 29. Сторінка наплічника.

Contacts

Phone: +380 99 999 99 99
 Email: temp@TempData.com
 Address: 00000, Ukraine, Kyiv, 00000, 00000, 00000

Рисунок 30 Сторінка контактної інформації для замовлення.

| | | | | | |
|-----|-------|---------|--------|------|-------|
| | | | | | Аркуш |
| | | | | | 48 |
| Зм. | Аркуш | № докум | Підпис | Дата | |

ДТЕУ 121 07-19.БР

В цілому, розроблений клієнт-серверний додаток є функціональним прототипом, який може бути використаний в подальшому для розвитку та вдосконалення. Дана робота може служити основою для подальших досліджень у галузі розробки клієнт-серверних додатків та веб-програмування загалом.



| | | | | | | |
|-----|-------|---------|--------|------|-------------------|-------|
| | | | | | | Аркуш |
| | | | | | ДТЕУ 121 07-19.БР | 49 |
| Зм. | Аркуш | № докум | Підпис | Дата | | |

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

З розвитком технологій людям стало набагато легше здійснювати покупки онлайн, тому розробка веб-сайту з реалізації наплічників є необхідністю для бізнесу.

У процесі розробки було проведено аналіз ринку, вивчено основні тенденції у сфері моди та аксесуарів, а також проаналізовано існуючі веб-сайти, які пропонують аналогічні товари. На основі цього аналізу було визначено основні вимоги та функціональні можливості веб-сайту з реалізації наплічників.

В результаті роботи було розроблено зручний та естетично привабливий веб-інтерфейс, який дозволяє користувачам швидко та зручно переглядати асортимент наплічників, здійснювати пошук за різними параметрами та здійснювати онлайн-замовлення.

В майбутньому можна буде додати в функціонал 3д моделі рюкзаків, так у користувача буде відчуття реального магазину, і зробіть простішим вибір.

Загалом, розробка веб-сайту з реалізації наплічників була цікавим та важливим процесом, що вимагав використання різноманітних навичок та знань у галузі веб-розробки. Результатом цієї роботи є функціональний та привабливий веб-сайт, який відповідає потребам споживачів та має потенціал для подальшого розвитку та успіху Розроблена система повністю задовольняє вимоги до системи, програмного забезпечення та до технічного забезпечення.

| | | | | | <i>ДТЕУ 121 07-19.БР</i> | | | |
|------------|--------------|------------------|---------------|-------------|--|---|--------------|----------------|
| <i>Зм.</i> | <i>Аркуш</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> | | | | |
| Зав. каф. | | Криворучко О.В. | | 15.01.23 | Клієнт-серверний додаток реалізації наплічників | <i>Стадія</i> | <i>Аркуш</i> | <i>Аркушів</i> |
| Керівник | | Власенко Л.О. | | 15.01.23 | | <i>ВП</i> | <i>50</i> | <i>51</i> |
| Гарант | | Рзаєва С.Л. | | 15.01.23 | | <i>Факультет інформаційних технологій 4 курс, 7 група</i> | | |
| Розробив | | Пересадько О. О. | | 15.01.23 | | | | |
| | | | | | <i>Висновки та пропозиції</i> | | | |

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Веб сайт Bootstrap [Електронний ресурс]. – Режим доступу : <https://getbootstrap.com/>. Дата звернення: 21.02.2023 р.
2. Веб сайт Microsoft «ASP.NET» [Електронний ресурс]. – Режим доступу : <https://dotnet.microsoft.com/en-us/apps/aspnet>. Дата звернення: 21.02.2023 р.
3. Веб сайт W3Schools [Електронний ресурс]. – Режим доступу : <https://www.w3schools.com/>. Дата звернення: 21.02.2023 р.
4. Веб сайт MDN Web Docs [Електронний ресурс]. – Режим доступу : <https://developer.mozilla.org/>. Дата звернення: 21.02.2023 р.
5. Веб сайт Microsoft «C# programming guide» [Електронний ресурс]. – Режим доступу : <https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/>. Дата звернення: 21.02.2023 р.
6. Веб сайт Microsoft SQL Server Documentation [Електронний ресурс]. – Режим доступу : <https://docs.microsoft.com/en-us/sql/sql-server/>. Дата звернення: 21.02.2023 р.
7. Веб сайт Diagrams [Електронний ресурс]. – Режим доступу : <https://app.diagrams.net/>. Дата звернення: 21.02.2023 р.

| | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|--------------|------------------|---------------|-------------|--|--|--|--|--|--|---------------|--------------|----------------|--|--|------------|----|----|--|--|---|--|--|--|--|
| <i>ДТЕУ 121 07-19.БР</i> | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | |
| <i>Зм.</i> | <i>Аркуш</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> | | | | | | | | | | | | | | | | | | | | | |
| Зав. каф. | | Криворучко О.В. | | 20.01.23 | Клієнт-серверний додаток реалізації наплічників <i>Список використаних джерел</i> | | | | | | | | | | | | | | | | | | | | |
| Керівник | | Власенко Л.О. | | 20.01.23 | | | | | | | | | | | | | | | | | | | | | |
| Гарант | | Рзаєва С.Л. | | 20.01.23 | | | | | | | | | | | | | | | | | | | | | |
| Розробив | | Пересацько О. О. | | 20.01.23 | | | | | | | | | | | | | | | | | | | | | |
| | | | | | <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 20%;"></td> <td style="width: 20%;"></td> <td style="width: 20%;"></td> <td style="width: 20%;"></td> <td style="width: 20%;"></td> </tr> <tr> <td style="text-align: center;"><i>Стадія</i></td> <td style="text-align: center;"><i>Аркуш</i></td> <td style="text-align: center;"><i>Аркушів</i></td> <td colspan="2"></td> </tr> <tr> <td style="text-align: center;"><i>СВД</i></td> <td style="text-align: center;">51</td> <td style="text-align: center;">51</td> <td colspan="2"></td> </tr> <tr> <td colspan="5" style="text-align: center;"> Факультет інформаційних технологій 4 курс, 7 група </td> </tr> </table> | | | | | | <i>Стадія</i> | <i>Аркуш</i> | <i>Аркушів</i> | | | <i>СВД</i> | 51 | 51 | | | Факультет інформаційних технологій 4 курс, 7 група | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | |
| <i>Стадія</i> | <i>Аркуш</i> | <i>Аркушів</i> | | | | | | | | | | | | | | | | | | | | | | | |
| <i>СВД</i> | 51 | 51 | | | | | | | | | | | | | | | | | | | | | | | |
| Факультет інформаційних технологій 4 курс, 7 група | | | | | | | | | | | | | | | | | | | | | | | | | |

Додатки

Додаток А

Код класу Controller

```
using Microsoft.AspNetCore.Mvc;
using Microsoft.AspNetCore.Mvc.Rendering;
using Microsoft.EntityFrameworkCore;
using BackpackShop.WebApplication.Data;
using BackpackShop.WebApplication.Models;
using BackpackShop.WebApplication.ViewModel;
using Microsoft.AspNetCore.Authorization;

namespace BackpackShop.WebApplication.Controllers;

public class BackpacksController : Controller
{
    private readonly ApplicationDbContext _context;

    public BackpacksController(ApplicationDbContext context)
    {
        _context = context;
    }

    [Authorize(Roles = "Admin")]
    public async Task<IActionResult> Index()
    {
        var applicationDbContext = _context.Backpack.Include(b => b.Category);
        return View(await applicationDbContext.ToListAsync());
    }

    public async Task<IActionResult> Details(int? id)
    {
        if (id == null || _context.Backpack == null)
        {
            return NotFound();
        }

        var backpack = await _context.Backpack
            .Include(b => b.Category)
            .FirstOrDefaultAsync(m => m.Id == id);
        if (backpack == null)
        {
            return NotFound();
        }

        return View(backpack);
    }

    [Authorize(Roles = "Admin")]
    public IActionResult Create()
    {
        ViewData["CategoryId"] = new SelectList(_context.Category, "Id", "Name");
        return View();
    }

    [HttpPost]
    [ValidateAntiForgeryToken]
    [Authorize(Roles = "Admin")]
    public async Task<IActionResult> Create([Bind("Id,Name,Description,Capacity,Color,Image,Price,CategoryId")]
        BackpackOperationViewModel backpack)
```

```

    {
        if (!ModelState.IsValid) return View(backpack);

        var lastItemId = _context.Backpack.Count() + 1;

        var extension = Path.GetExtension(backpack.Image.FileName);
        if (extension != ".jpg" && extension != ".png" && extension != ".jpeg" && extension != ".gif" && extension != ".bmp" && extension != ".tiff")
            throw new Exception("Image must be jpg or png format");

        var filename = "Backpack" + lastItemId + extension;
        var combine = Path.Combine(Directory.GetCurrentDirectory(), "wwwroot/img/Backpacks", filename);
        await using var fileStream = new FileStream(combine, FileMode.Create);
        await backpack.Image.CopyToAsync(fileStream);

        var relativePath = "/img/Backpacks/" + filename;

        if (backpack.Image == null)
            relativePath = "/img/defaultBackpack.webp";

        var newBackpack = new Backpack
        {
            Name = backpack.Name,
            Description = backpack.Description,
            ImageUrl = relativePath
        };

        _context.Add(newBackpack);
        await _context.SaveChangesAsync();
        return RedirectToAction(nameof(Index));
    }

    [Authorize(Roles = "Admin")]
    public async Task<IActionResult> Edit(int? id)
    {
        if (id == null || _context.Backpack == null)
        {
            return NotFound();
        }

        var backpack = await _context.Backpack.FindAsync(id);
        if (backpack == null)
        {
            return NotFound();
        }
        ViewData["CategoryId"] = new SelectList(_context.Category, "Id", "Name", backpack.CategoryId);

        var backpackOperationViewModel = new BackpackOperationViewModel
        {
            Id = backpack.Id,
            Name = backpack.Name,
            Description = backpack.Description,
            Capacity = backpack.Capacity,
            Color = backpack.Color,
            Price = backpack.Price,
            CategoryId = backpack.CategoryId
        };

        return View(backpackOperationViewModel);
    }

    [Authorize(Roles = "Admin")]

```

```

[HttpPost]
[ValidateAntiForgeryToken]
public async Task<ActionResult> Edit(int id, [Bind("Id,Name,Description,Capacity,Color,Image,Price,CategoryId")]
BackpackOperationViewModel backpack)
{
    if (id != backpack.Id)
    {
        return NotFound();
    }

    if (ModelState.IsValid)
    {
        try
        {
            var backpackToUpdate = await _context.Backpack.FindAsync(id);

            backpackToUpdate.Name = backpack.Name;
            backpackToUpdate.Description = backpack.Description;
            backpackToUpdate.Capacity = backpack.Capacity;
            backpackToUpdate.Color = backpack.Color;
            backpackToUpdate.Price = backpack.Price;
            backpackToUpdate.CategoryId = backpack.CategoryId;

            if (backpack.Image != null)
            {
                var lastItemId = backpackToUpdate.Id;

                var extension = Path.GetExtension(backpack.Image.FileName);
                if (extension != ".jpg" && extension != ".png" && extension != ".jpeg" && extension != ".gif" && extension
                != ".bmp" && extension != ".tiff")
                    throw new Exception("Image must be jpg or png format");

                var filename = "Backpack" + lastItemId + extension;
                var combine = Path.Combine(Directory.GetCurrentDirectory(), "wwwroot/img/Backpacks", filename);
                await using FileStream fileStream = new FileStream(combine, FileMode.Create);
                await backpack.Image.CopyToAsync(fileStream);

                var relativePath = "/img/Backpacks/" + filename;

                backpackToUpdate.ImageUrl = relativePath;
            }

            _context.Update(backpack);
            await _context.SaveChangesAsync();
        }
        catch (DbUpdateConcurrencyException)
        {
            if (!BackpackExists(backpack.Id))
            {
                return NotFound();
            }
            else
            {
                throw;
            }
        }
        return RedirectToAction(nameof(Index));
    }
    ViewData["CategoryId"] = new SelectList(_context.Category, "Id", "Name", backpack.CategoryId);
    return View(backpack);
}

[Authorize(Roles = "Admin")]

```

```

public async Task<IActionResult> Delete(int? id)
{
    if (id == null || _context.Backpack == null)
    {
        return NotFound();
    }

    var backpack = await _context.Backpack
        .Include(b => b.Category)
        .FirstOrDefaultAsync(m => m.Id == id);
    if (backpack == null)
    {
        return NotFound();
    }

    return View(backpack);
}

[Authorize(Roles = "Admin")]
[HttpPost, ActionName("Delete")]
[ValidateAntiForgeryToken]
public async Task<IActionResult> DeleteConfirmed(int id)
{
    if (_context.Backpack == null)
    {
        return Problem("Entity set 'ApplicationDbContext.Backpack' is null.");
    }
    var backpack = await _context.Backpack.FindAsync(id);
    if (backpack != null)
    {
        _context.Backpack.Remove(backpack);
    }

    await _context.SaveChangesAsync();
    return RedirectToAction(nameof(Index));
}

private bool BackpackExists(int id)
{
    return (_context.Backpack?.Any(e => e.Id == id)).GetValueOrDefault();
}
}

```

```

using Microsoft.AspNetCore.Mvc;
using Microsoft.EntityFrameworkCore;
using BackpackShop.WebApplication.Data;
using BackpackShop.WebApplication.Models;
using BackpackShop.WebApplication.ViewModel;
using Microsoft.AspNetCore.Authorization;

namespace BackpackShop.WebApplication.Controllers;

```

```

public class CategoriesController : Controller
{
    private readonly ApplicationDbContext _context;

    public CategoriesController(ApplicationDbContext context)
    {
        _context = context;
    }
}

```

```

[Authorize(Roles = "Admin")]
public async Task<IActionResult> Index()
{
    return _context.Category != null ?
        View(await _context.Category.ToListAsync()) :
        Problem("Entity set 'ApplicationDbContext.Category' is null.");
}

public async Task<IActionResult> Details(int? id)
{
    if (id == null || _context.Category == null)
    {
        return NotFound();
    }

    var category = await _context.Category
        .FirstOrDefaultAsync(m => m.Id == id);

    var backpacks = await _context.Backpack
        .Where(b => b.CategoryId == id)
        .ToListAsync();

    var categoryViewModel = new CategoryDetailsViewModel()
    {
        Category = category,
        Backpacks = backpacks
    };

    if (category == null)
    {
        return NotFound();
    }

    return View(categoryViewModel);
}

[Authorize(Roles = "Admin")]
public IActionResult Create()
{
    return View();
}

[Authorize(Roles = "Admin")]
[HttpPost]
[ValidateAntiForgeryToken]
public async Task<IActionResult> Create([Bind("Id,Name,Description,Image")] CategoryOperationViewModel
category)
{
    if (!ModelState.IsValid) return View(category);

    var lastItemId = _context.Category.Count() + 1;

    var extension = Path.GetExtension(category.Image.FileName);
    if (extension != ".jpg" && extension != ".png" && extension != ".jpeg" && extension != ".gif" && extension !=
".bmp" && extension != ".tiff")
        throw new Exception("Image must be jpg or png format");

    var filename = "Category" + lastItemId + extension;
    var combine = Path.Combine(Directory.GetCurrentDirectory(), "wwwroot/img/Categories", filename);
    await using FileStream fileStream = new FileStream(combine, FileMode.Create);
    await category.Image.CopyToAsync(fileStream);

    var relativePath = "/img/Categories/" + filename;

```

Продовження дод. А

```

if (category.Image == null)
    relativePath = "/img/defaultCategory.jpg";

var newCategory = new Category
{
    Name = category.Name,
    Description = category.Description,
    ImageUrl = relativePath
};

_context.Add(newCategory);
await _context.SaveChangesAsync();
return RedirectToAction(nameof(Index));
}

[Authorize(Roles = "Admin")]
public async Task<IActionResult> Edit(int? id)
{
    if (id == null || _context.Category == null)
    {
        return NotFound();
    }

    var category = await _context.Category.FindAsync(id);
    if (category == null)
    {
        return NotFound();
    }

    var categoryOperationViewModel = new CategoryOperationViewModel
    {
        Id = category.Id,
        Name = category.Name,
        Description = category.Description
    };

    return View(categoryOperationViewModel);
}

[Authorize(Roles = "Admin")]
[HttpPost]
[ValidateAntiForgeryToken]
public async Task<IActionResult> Edit(int id, [Bind("Id,Name,Description,Image")] CategoryOperationViewModel
category)
{
    if (id != category.Id)
    {
        return NotFound();
    }

    if (ModelState.IsValid)
    {
        try
        {
            var categoryToUpdate = await _context.Category.FindAsync(id);
            categoryToUpdate.Name = category.Name;
            categoryToUpdate.Description = category.Description;

            if (category.Image != null)
            {
                var lastItemId = categoryToUpdate.Id;

```

Продовження дод. А


```

var extension = Path.GetExtension(category.Image.FileName);
if (extension != ".jpg" && extension != ".png" && extension != ".jpeg" && extension != ".gif" && extension
!= ".bmp" && extension != ".tiff")
    throw new Exception("Image must be jpg or png format");

var filename = "Category" + lastItemId + extension;
var combine = Path.Combine(Directory.GetCurrentDirectory(), "wwwroot/img/Categories", filename);
await using FileStream fileStream = new FileStream(combine, FileMode.Create);
await category.Image.CopyToAsync(fileStream);

var relativePath = "/img/Categories/" + filename;

categoryToUpdate.ImageUrl = relativePath;
}
_context.Update(categoryToUpdate);
await _context.SaveChangesAsync();
}
catch (DbUpdateConcurrencyException)
{
    if (!CategoryExists(category.Id))
    {
        return NotFound();
    }
    else
    {
        throw;
    }
}
return RedirectToAction(nameof(Index));
}
return View(category);
}

[Authorize(Roles = "Admin")]
public async Task<IActionResult> Delete(int? id)
{
    if (id == null || _context.Category == null)
    {
        return NotFound();
    }

    var category = await _context.Category
        .FirstOrDefaultAsync(m => m.Id == id);
    if (category == null)
    {
        return NotFound();
    }

    return View(category);
}

[Authorize(Roles = "Admin")]
[HttpPost, ActionName("Delete")]
[ValidateAntiForgeryToken]
public async Task<IActionResult> DeleteConfirmed(int id)
{
    if (_context.Category == null)
    {
        return Problem("Entity set 'ApplicationDbContext.Category' is null.");
    }
    var category = await _context.Category.FindAsync(id);

```

Продовження дод. А

```

    if (category != null)
    {
        _context.Category.Remove(category);
    }

    await _context.SaveChangesAsync();
    return RedirectToAction(nameof(Index));
}

private bool CategoryExists(int id)
{
    return (_context.Category?.Any(e => e.Id == id)).GetValueOrDefault();
}
}

using BackpackShop.WebApplication.Models;
using Microsoft.AspNetCore.Mvc;
using System.Diagnostics;
using BackpackShop.WebApplication.Data;
using BackpackShop.WebApplication.ViewModel;

namespace BackpackShop.WebApplication.Controllers;

public class HomeController : Controller
{
    private readonly ILogger<HomeController> _logger;
    private readonly ApplicationDbContext _context;

    public HomeController(ILogger<HomeController> logger, ApplicationDbContext context)
    {
        _logger = logger;
        _context = context;
    }

    public IActionResult Index()
    {
        var categories = _context.Category.ToList();
        var backpacks = _context.Backpack.ToList();

        var viewModel = new HomePageViewModel
        {
            Categories = categories,
            Backpacks = backpacks
        };

        return View(viewModel);
    }

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

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

    [ResponseCache(Duration = 0, Location = ResponseCacheLocation.None, NoStore = true)]
    public IActionResult Error()
    {

```

Продовження дод. А

```

    return View(new ErrorViewModel { RequestId = Activity.Current?.Id ?? HttpContext.TraceIdentifier });
}
}

```

```

using Microsoft.AspNetCore.Identity.EntityFrameworkCore;
using Microsoft.EntityFrameworkCore;

namespace BackpackShop.WebApplication.Data;

public class ApplicationDbContext : IdentityDbContext
{
    public ApplicationDbContext(DbContextOptions<ApplicationDbContext> options)
        : base(options)
    {
    }

    public DbSet<Models.Backpack> Backpack { get; set; }
    public DbSet<Models.Category> Category { get; set; }
}

```

Код класу Data

Додаток Б

```

using BackpackShop.WebApplication.Models;
using Microsoft.AspNetCore.Identity;

namespace BackpackShop.WebApplication.Data;

public static class Seed
{
    public static void SeedCategories(this IApplicationBuilder app)
    {

```

Продовження дод. Б

```

        using var serviceScope = app.ApplicationServices.CreateScope();
        var context = serviceScope.ServiceProvider.GetService<ApplicationDbContext>();
        if (context == null) return;
        if (context.Category.Any()) return;
        context.Category.AddRange(
            new Category
            {
                Name = "Hiking",
                Description = "Backpacks for hiking for all ages and sizes",
                ImageUrl = "/img/defaultCategory.jpg",
            },
            new Category
            {
                Name = "Camping",
                Description = "Backpacks for camping for all ages and sizes",
                ImageUrl = "/img/defaultCategory.jpg",
            },
            new Category
            {
                Name = "Traveling",
                Description = "Backpacks for traveling for all ages and sizes",
                ImageUrl = "/img/defaultCategory.jpg",
            },
            new Category
            {
                Name = "School",
                Description = "Backpacks for school for all ages and sizes",
            }
        );
    }
}

```

```

        imageUrl = "/img/defaultCategory.jpg",
    },
    new Category
    {
        Name = "Work",
        Description = "Backpacks for work for all ages and sizes",
        imageUrl = "/img/defaultCategory.jpg",
    }
);
context.SaveChanges();
}

public static void SeedBackpacks(this IApplicationBuilder app)
{
    using var serviceScope = app.ApplicationServices.CreateScope();
    var context = serviceScope.ServiceProvider.GetService<ApplicationDbContext>();
    if (context == null) return;
    context.Backpack.AddRange(
        new Backpack
        {
            Name = "Hiking Backpack",
            Description = "Hiking Backpack for all ages and sizes",
            Capacity = 30,
            Color = "Black",
            imageUrl = "/img/defaultBackpack.webp",
            Price = 100,
            CategoryId = context.Category.First(c => c.Name == "Hiking ").Id
        },
        new Backpack
        {
            Name = "Camping Backpack",
            Description = "Camping Backpack for all ages and sizes",
            Capacity = 30,
            Color = "Black",
            imageUrl = "/img/defaultBackpack.webp",
            Price = 100,
            CategoryId = context.Category.First(c => c.Name == "Camping ").Id
        },
        new Backpack
        {
            Name = "Traveling Backpack",
            Description = "Traveling Backpack for all ages and sizes",
            Capacity = 30,
            Color = "Black",
            imageUrl = "/img/defaultBackpack.webp",
            Price = 100,
            CategoryId = context.Category.First(c => c.Name == "Traveling ").Id,
        },
        new Backpack
        {
            Name = "School Backpack",
            Description = "School Backpack for all ages and sizes",
            Capacity = 30,
            Color = "Black",
            imageUrl = "/img/defaultBackpack.webp",
            Price = 100,
            CategoryId = context.Category.First(c => c.Name == "School ").Id
        },
        new Backpack
        {
            Name = "Work Backpack",
            Description = "Work Backpack for all ages and sizes",
            Capacity = 30,
            Color = "Black",

```

Продовження дод. Б

```

        ImageUrl = "/img/defaultBackpack.webp",
        Price = 100,
        CategoryId = context.Category.First(c => c.Name == "Work ").Id
    }
);

context.SaveChanges();

public static void SeedRoles(this ApplicationDbContext context)
{
    if (context.Roles.Any())
    {
        return;
    }

    context.Roles.AddRange(new List<IdentityRole>
    {
        new IdentityRole
        {
            Name = "Admin",
            NormalizedName = "ADMIN"
        },
        new IdentityRole
        {
            Name = "User",
            NormalizedName = "USER"
        }
    });

    context.SaveChanges();
}

public static void SeedUsers(this ApplicationDbContext context)
{
    if (context.Users.Any())
    {
        return;
    }

    var admin = new IdentityUser
    {
        UserName = "admin@admin.com",
        NormalizedUserName = "ADMIN@ADMIN.COM",
        Email = "admin@admin.com",
        NormalizedEmail = "ADMIN@ADMIN.COM",
        EmailConfirmed = true,
        LockoutEnabled = false,
        SecurityStamp = Guid.NewGuid().ToString("D")
    };

    var hasher = new PasswordHasher<IdentityUser>();
    admin.PasswordHash = hasher.HashPassword(admin, "Admin123!");

    context.Users.Add(admin);
    context.SaveChanges();

    context.UserRoles.Add(new IdentityUserRole<string>
    {
        UserId = admin.Id,
        RoleId = context.Roles.First(r => r.Name == "Admin").Id
    });

    context.SaveChanges();
}

```

Продовження дод. Б

```

var user = new IdentityUser
{
    UserName = "user",
    NormalizedUserName = "USER",
    Email = "user@user.com",
    NormalizedEmail = "USER@USER.COM",
    EmailConfirmed = true,
    LockoutEnabled = false,
    SecurityStamp = Guid.NewGuid().ToString("D")
};

user.PasswordHash = hasher.HashPassword(user, "User123!");

context.Users.Add(user);
context.SaveChanges();

context.UserRoles.Add(new IdentityUserRole<string>
{
    UserId = user.Id,
    RoleId = context.Roles.First(r => r.Name == "User").Id
});

context.SaveChanges();
}
}

```

Додаток В

Код класу Models

```
namespace BackpackShop.WebApplication.Models;
```

```

public class Backpack
{
    public int Id { get; set; }
    public string Name { get; set; }
    public string Description { get; set; }
    public int Capacity { get; set; }
    public string Color { get; set; }
    public string ImageUrl { get; set; }
    public decimal Price { get; set; }
    public int CategoryId { get; set; }
    public Category Category { get; set; }
}

```

```
namespace BackpackShop.WebApplication.Models;
```

```

public class Category
{
    public int Id { get; set; }
    public string Name { get; set; }
    public string Description { get; set; }
    public string ImageUrl { get; set; }
    public IEnumerable<Backpack> Backpacks { get; set; }
}

```

```

namespace BackpackShop.WebApplication.Models;
public class ErrorViewModel
{
    public string? RequestId { get; set; }
    public bool ShowRequestId => !string.IsNullOrEmpty(RequestId);
}

```

Код класу Views

Додаток Г

```
@model BackpackShop.WebApplication.ViewModel.BackpackOperationViewModel
```

```
@{
    ViewData["Title"] = "Create";
}
```

```
<h1>Create</h1>
```

```
<h4>Backpack</h4>
```

```
<hr />
```

```
<div class="row">
```

```
    <div class="col-md-4">
```

```
        <form asp-action="Create" enctype="multipart/form-data" method="post" class="form-horizontal">
```

```
            <div asp-validation-summary="ModelOnly" class="text-danger"></div>
```

```
            <div class="form-group">
```

```
                <label asp-for="Name" class="control-label"></label>
```

```
                <input asp-for="Name" class="form-control" />
```

```
                <span asp-validation-for="Name" class="text-danger"></span>
```

```
            </div>
```

```
            <div class="form-group">
```

```
                <label asp-for="Description" class="control-label"></label>
```

```
                <input asp-for="Description" class="form-control" />
```

```
                <span asp-validation-for="Description" class="text-danger"></span>
```

```
            </div>
```

```
            <div class="form-group">
```

```
                <label asp-for="Capacity" class="control-label"></label>
```

```
                <input asp-for="Capacity" class="form-control" />
```

```
                <span asp-validation-for="Capacity" class="text-danger"></span>
```

```
            </div>
```

```
            <div class="form-group">
```

```
                <label asp-for="Color" class="control-label"></label>
```

```
                <input asp-for="Color" class="form-control" />
```

```
                <span asp-validation-for="Color" class="text-danger"></span>
```

```
            </div>
```

```
            <div class="form-group">
```

```
                <label asp-for="Image" class="control-label"></label>
```

```
                <input type="file" asp-for="Image" class="form-control" />
```

```
                <span asp-validation-for="Image" class="text-danger"></span>
```

```
            </div>
```

```
            <div class="form-group">
```

```
                <label asp-for="Price" class="control-label"></label>
```

```
                <input asp-for="Price" class="form-control" />
```

```
                <span asp-validation-for="Price" class="text-danger"></span>
```

```
            </div>
```

```
            <div class="form-group">
```

```
                <label asp-for="CategoryId" class="control-label"></label>
```

```
                <select asp-for="CategoryId" class="form-control" asp-items="ViewBag.CategoryId"></select>
```

```
            </div>
```

Продовження дод. Г

```
<div class="form-group">
  <input type="submit" value="Create" class="btn btn-primary" />
</div>
</form>
</div>
</div>
```

```
@section Scripts {
  @{await Html.RenderPartialAsync("_ValidationScriptsPartial");}
}
```

```
@model BackpackShop.WebApplication.Models.Backpack
```

```
@{
  ViewData["Title"] = "Delete";
}
```

```
<h1>Delete</h1>
```

```
<h3>Are you sure you want to delete this?</h3>
```

```
<div>
  <h4>Backpack</h4>
  <hr />
```

```
<dl class="row">
  <dt class="col-sm-2">
    @Html.DisplayNameFor(model => model.Name)
  </dt>
  <dd class="col-sm-10">
    @Html.DisplayFor(model => model.Name)
  </dd>
  <dt class="col-sm-2">
    @Html.DisplayNameFor(model => model.Description)
  </dt>
  <dd class="col-sm-10">
    @Html.DisplayFor(model => model.Description)
  </dd>
  <dt class="col-sm-2">
    @Html.DisplayNameFor(model => model.Capacity)
  </dt>
  <dd class="col-sm-10">
    @Html.DisplayFor(model => model.Capacity)
  </dd>
  <dt class="col-sm-2">
    @Html.DisplayNameFor(model => model.ImageUrl)
  </dt>
  <dd class="col-sm-10">
    @Html.DisplayFor(model => model.ImageUrl)
  </dd>
  <dt class="col-sm-2">
    @Html.DisplayNameFor(model => model.Price)
  </dt>
  <dd class="col-sm-10">
    @Html.DisplayFor(model => model.Price)
  </dd>
  <dt class="col-sm-2">
    @Html.DisplayNameFor(model => model.Category)
  </dt>
  <dd class="col-sm-10">
    @Html.DisplayFor(model => model.Category.Id)
  </dd>
</dl>
```

```
<form asp-action="Delete">
```

Продовження дод. Г


```

<input type="hidden" asp-for="Id" />
<input type="submit" value="Delete" class="btn btn-danger" /> |
<a asp-action="Index">Back to List</a>
</form>
</div>

@model BackpackShop.WebApplication.Models.Backpack

@{
    ViewData["Title"] = "Details";
}

<h2>@Model.Name</h2>

<div style="display: flex; flex-direction: row; flex-wrap: wrap; justify-content: center;">
    
    <div>
        <h4>@Model.Name</h4>
        <hr />
        <p>@Model.Description</p>

        <p>Category: @Model.Category.Name</p>
        <p>Price: @Model.Price UAH</p>
        <p>Color: @Model.Color</p>
        <p>Capacity: @Model.Capacity</p>
        <a class="btn" style="background: aquamarine;" asp-controller="Home" asp-action="Contacts"> Contacts to buy!
    </a>
    </div>
</div>
@if (User.IsInRole("Admin"))
{
    <div>
        <a asp-action="Delete" asp-route-id="@Model?.Id">Delete</a> |
        <a asp-action="Edit" asp-route-id="@Model?.Id">Edit</a> |
        <a asp-controller="Categories" asp-action="Details" asp-route-id="@Model.CategoryId">Back to category</a>
    </div>
}

@model BackpackShop.WebApplication.ViewModel.BackpackOperationViewModel

@{
    ViewData["Title"] = "Edit";
}

<h1>Edit</h1>

<h4>Backpack</h4>
<hr />
<div class="row">
    <div class="col-md-4">
        <form asp-action="Edit" enctype="multipart/form-data" method="post" class="form-horizontal">
            <div asp-validation-summary="ModelOnly" class="text-danger"></div>
            <input type="hidden" asp-for="Id" />
            <div class="form-group">
                <label asp-for="Name" class="control-label"></label>
                <input asp-for="Name" class="form-control" />
                <span asp-validation-for="Name" class="text-danger"></span>
            </div>
            <div class="form-group">
                <label asp-for="Description" class="control-label"></label>
                <input asp-for="Description" class="form-control" />
                <span asp-validation-for="Description" class="text-danger"></span>
            </div>
        </form>
    </div>
</div>

```

Продовження под. Г

```

</div>
<div class="form-group">
  <label asp-for="Capacity" class="control-label"></label>
  <input asp-for="Capacity" class="form-control" />
  <span asp-validation-for="Capacity" class="text-danger"></span>
</div>
<div class="form-group">
  <label asp-for="Color" class="control-label"></label>
  <input asp-for="Color" class="form-control" />
  <span asp-validation-for="Color" class="text-danger"></span>
</div>
<div class="form-group">
  <label asp-for="Image" class="control-label"></label>
  <input type="file" asp-for="Image" class="form-control" />
  <span asp-validation-for="Image" class="text-danger"></span>
</div>
<div class="form-group">
  <label asp-for="Price" class="control-label"></label>
  <input asp-for="Price" class="form-control" />
  <span asp-validation-for="Price" class="text-danger"></span>
</div>
<div class="form-group">
  <label asp-for="CategoryId" class="control-label"></label>
  <select asp-for="CategoryId" class="form-control" asp-items="ViewBag.CategoryId"></select>
  <span asp-validation-for="CategoryId" class="text-danger"></span>
</div>
<div class="form-group">
  <input type="submit" value="Save" class="btn btn-primary" />
</div>
</form>
</div>
</div>
<a asp-action="Index">Back to List</a>
</div>
@section Scripts {
  @{await Html.RenderPartialAsync("_ValidationScriptsPartial");}
}

@model IEnumerable<BackpackShop.WebApplication.Models.Backpack>
@{
  ViewData["Title"] = "Index";
}

<h1>Backpack admin panel</h1>
<p>
  <a asp-action="Create">Create New</a>
</p>
<table class="table">
  <thead>
    <tr>
      <th>
        @Html.DisplayNameFor(model => model.Name)
      </th>
      <th>
        @Html.DisplayNameFor(model => model.Description)
      </th>
      <th>
        @Html.DisplayNameFor(model => model.Capacity)
      </th>
    </tr>
  </thead>
</table>

```

Продовження дод. Г

```

</th>
<th>
    @Html.DisplayNameFor(model => model.Color)
</th>
<th>
    Image
</th>
<th>
    @Html.DisplayNameFor(model => model.Price)
</th>
<th>
    @Html.DisplayNameFor(model => model.Category)
</th>
</tr>

```

Продовження дод. Г

```

</thead>
<tbody>
@foreach (var item in Model) {
    <tr>
        <td>
            @Html.DisplayFor(modelItem => item.Name)
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.Description)
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.Capacity)
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.Color)
        </td>
        <td>
            
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.Price)
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.Category.Id)
        </td>
        <td>
            <a asp-action="Edit" asp-route-id="@item.Id">Edit</a> |
            <a asp-action="Details" asp-route-id="@item.Id">Details</a> |
            <a asp-action="Delete" asp-route-id="@item.Id">Delete</a>
        </td>
    </tr>
}
</tbody>
</table>

```