

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ХАРЧОВИХ ТЕХНОЛОГІЙ

Інститут (факультет) автоматизації і комп'ютерних системКафедра Інформаційних технологій, штучного інтелекту і кібербезпекиОсвітній ступінь бакалаврСпеціальність 122 «Комп'ютерні науки»
(код і назва)Освітньо-професійна програма Комп'ютерні науки
(назва)**ЗАТВЕРДЖУЮ**

Завідувач

Кафедри Інформаційних технологій,штучного інтелекту і кібербезпекиСергій ГРИБКОВ«28» Квітня 2025 року**З А В Д А Н Н Я****НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧА**Гриб Ангеліна Сергіївна

(прізвище, ім'я, по батькові)

1. Тема роботи: Розроблення інтернет-магазину взуття ФОП "Синчук П.П."керівник роботи Грибков Сергій Віталійович, д.т.н, проф.,

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від «28» Квітня 2025 р. № 254-КС2. Строк подання здобувачем роботи: 30.05.20253. Вихідні дані до роботи: Інформація про ФОП «Синчук П.П.», посадові інструкції, каталог продукції, методи розробки інтернет магазину, скріни структури проєкту.

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити):

Дослідження діяльності ФОП «Синчук П.П.», технічне завдання на проєктування, розробка інтернет – магазину для продажу взуття

5. Перелік графічного матеріалу:

Скріни форм інтерфейсу користувача, структура бази даних, структура інтернет – магазину, функціональні моделі, моделі організаційних структур

6. Консультанти розділів роботи:

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1	Грибков С.В., професор	28.04.25	12.05.2025
2	Грибков С.В., професор	28.04.25	19.05.2025
3	Грибков С.В., професор	28.04.25	26.05.2025

7. Дата видачі завдання: _____

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів виконання кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Дослідження діяльності підприємства	29.04.2025	Виконано
2	Огляд існуючих методів	30.04.2025	Виконано
3	Розробка бази даних	05.05.2025	Виконано
4	Розробка інтернет-магазину	20.05.2025	Виконано
5	Оформлення роботи	28.05.2025	Виконано
6	Розробка презентації	30.05.2025	Виконано

Здобувач _____ Ангеліна ГРИБ

Керівник роботи _____ Сергій ГРИБКОВ

АНОТАЦІЯ

Метою кваліфікаційної роботи є створення інтернет-магазину взуття ФОП Синчук П.П. Для розроблення інтернет-магазину використано наступні технології та інструментарій: MS SQL 2019, .NET 8, MS SQL Server Management Studio 2019 та Visual Studio 2022. Створений інтернет-магазин дозволить спростити продаж взуття, додавати нову продукцію, нові категорії, займатись управлінням замовленнями. Інтернет-магазин дає можливість ефективно керувати продукцією, що в свою чергу забезпечує високий рівень продуктивності. Система передбачає розмежування доступу між адміністративною частиною та користувачем. Впроваджено механізм авторизації, управління замовленнями та валідацію введених даних. Було проведено тестування функціоналу з метою перевірки коректної роботи ключових модулів системи.

Обсяг дипломної роботи: 84 сторінок, 9 таблиць, 58 рисунків.

КЛЮЧОВІ СЛОВА: ІНТЕРНЕТ-МАГАЗИН, ВЕБ-ЗАСТОСУНОК, БАЗА ДАНИХ, .NET, ПРОДАЖ ВЗУТТЯ, ТЕСТУВАННЯ.

ABSTRACT

The purpose of the qualification work is to create an online shoe store for the individual entrepreneur Sinchuk P.P. The following technologies and tools were used to develop the online store: MS SQL 2019, .NET 8, MS SQL Server Management Studio 2019 and Visual Studio 2022. The created online store will simplify the sale of shoes, add new products, new categories, and manage orders. The online store makes it possible to effectively manage products, which in turn ensures a high level of productivity. The system provides for the separation of access between the administrative part and the user. An authorization mechanism, order management, and validation of entered data have been implemented. Functional testing was conducted to verify the correct operation of key system modules.

Thesis volume: 84 pages, 9 tables, 58 figures.

KEYWORDS: ONLINE STORE, WEB APPLICATION, DATABASE, .NET, FOOTWEAR SALES, TESTING.

ЗМІСТ

ВСТУП.....	8
РОЗДІЛ 1. ДОСЛІДЖЕННЯ ДІЯЛЬНОСТІ ФОП «СИНЧУК П.П.».....	10
1.1 Загальна характеристика ФОП «Синчук П.П.»	10
1.2 Організаційна структура ФОП «Синчук П.П.»	12
1.3 Стан автоматизації на ФОП «Синчук П.П.»	14
1.4 Розробка функціональної моделі існуючих бізнес – процесів.....	16
1.5 Огляд існуючих рішень для розв’язання виявлених проблем.....	17
1.6 Огляд майбутньої системи	20
1.7 Розрахунок економічного ефекту впровадження інтернет – магазину.....	21
РОЗДІЛ 2. ТЕХНІЧНЕ ЗАВДАННЯ НА ПРОЄКТУВАННЯ	28
2.1. Загальні положення	28
2.2. Призначення і цілі створення системи	28
2.3. Характеристика об’єкта автоматизації	28
2.4. Вимоги до системи	29
2.5. Склад і зміст робіт по створенню системи.....	38
2.6. Порядок контролю і приймання системи	39
2.7. Вимоги до складу і змісту робіт із підготовки до введення системи в дію.....	40
2.8. Вимоги до документації.....	40
2.9. Джерела розробки	41
РОЗДІЛ 3 ПРОЄКТУВАННЯ, СТВОРЕННЯ ТА АПРОБАЦІЯ ІНТЕРНЕТ – МАГАЗИНУ ДЛЯ ПРОДАЖУ ВЗУТТЯ	42
3.1 Аналіз мов програмування для розробки систем	42
3.2 Інформаційне забезпечення системи.....	43
3.3 Реалізація функцій системи	46
3.3 Інструкція користувача.....	55
3.4 Управління контентом інтернет – магазину.....	59
3.5 Тестування інтернет – магазину	64
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	83
ДОДАТКИ.....	85

Додаток А. Код програми.....	85
------------------------------	----

ВСТУП

Сучасний бізнес неможливо уявити без інформаційних технологій: інформатизація підприємства підвищує його конкурентоспроможність і ефективність. Особливо це актуально для малого бізнесу, який зазвичай має обмежені ресурси для розвитку [24]. Як зазначають дослідники, «цифрова трансформація малих і середніх підприємств (SME) в Україні є критичним пріоритетом, що сприяє підвищенню конкурентоспроможності, ефективності та стійкості бізнесу в сучасних умовах».

Впровадження цифрових рішень дозволяє підприємцям автоматизувати операції, централізовано зберігати інформацію про товари та клієнтів, редагувати дані в реальному часі і оперативно приймати управлінські рішення.

У разі ФОП «Синчук П.П.», який займається роздрібною торгівлею взуття, це надає можливість значно розширити канали продажів і підвищити якість обслуговування клієнтів. Інтернет-магазин як цифровий канал реалізації продукції стає особливо актуальним для малого бізнесу: він дає змогу працювати 24/7, залучити ширшу аудиторію та знизити витрати на традиційні магазини [19]. Згідно з аналітикою, ринок електронної торгівлі України демонструє стійке зростання: у 2023 році понад 9 млн українців здійснили покупки онлайн, а обсяг продажів перевищив 182 млрд гривень. Навіть за складних умов воєнного часу кількість замовлень у вітчизняних інтернет-магазинах суттєво перевищувала допандемічний рівень – наприклад, весною 2023-го на платформі Хорошоп було майже на чверть більше, ніж до повномасштабного вторгнення.

Ці тенденції підтверджують, що для ФОП «Синчук П.П.» створення власного онлайн-магазину взуття є логічним кроком. Онлайн-платформа дозволить підприємцю вирішити такі основні задачі: демонстрацію повного асортименту товарів, прийом та обробку замовлень через Інтернет, інтеграцію з платіжними та логістичними сервісами, а також збір аналітичних даних про потреби клієнтів. В умовах цифровізації малий бізнес отримує низку суттєвих переваг: за даними економічних досліджень, «основні переваги цифрової трансформації – зниження транзакційних витрат, збільшення обсягів товарообігу та покращення взаємодії з

клієнтами й постачальниками». Онлайн-торгівля суттєво зменшує витрати на утримання фізичних магазинів і персоналу, оскільки бізнес може працювати з меншими накладними витратами.

Система інтернет-магазину створює єдину електронну базу інформації про товари, ціни та запаси, що дозволяє власнику оперативно оновлювати дані в режимі реального часу. Застосування технологій ASP.NET та Microsoft SQL Server забезпечує надійну платформу для веб-розробки: як відзначається в технічній літературі, «ASP.NET – це безплатна веб-платформа від Microsoft для створення інтерактивних вебзастосунків на базі .NET».

Застосування цих інструментів у поєднанні з сучасним середовищем розробки Visual Studio дозволяє створювати безпечні, масштабовані і зручні у подальшій експлуатації рішення. Для ФОП «Синчук П.П.» розробка інтернет-магазину взуття є актуальною з огляду на загальнонаціональні тренди цифровізації та державну підтримку. В Україні існує офіційна нормативно-правова база для електронної комерції (зокрема, Закон «Про електронну комерцію» №675-VIII від 03.09.2015), а держава через платформу «Дія.Бізнес» сприяє цифровізації підприємництва. Таким чином, впровадження інформаційної системи для онлайн-торгівлі взуттям забезпечить ФОП вихід на нові ринки збуту, підвищить ефективність обліку та управління бізнесом, а отже – сприятиме зростанню його конкурентоспроможності в цілому

РОЗДІЛ 1. ДОСЛІДЖЕННЯ ДІЯЛЬНОСТІ ФОП «СИНЧУК П.П.»

1.1 Загальна характеристика ФОП «Синчук П.П.»

ФОП Синчук П.П. засноване у 2019 році, орієнтується на широке коло споживачів, та продає стильне та доступне взуття. В асортимент включено як і крафтове взуття, так і відомі бренди. Основна ідея компанії поєднати сучасний дизайн та зручність продукції.

Магазин зосереджувався на продажі класичного шкіряного взуття. Завдяки високій якості продукції та доступним цінам Синчук П.П. швидко завоював популярність серед покупців. В 2020 році компанія розширила приміщення, а також почала співпрацювати з українськими та зарубіжними компаніями. Підприємство розширило свій асортимент, додавши до асортименту спортивне та дитяче взуття.

На сьогоднішній день – це сучасний магазин з розвиненою клієнтською базою. Компанія пропонує асортимент взуття для чоловіків, жінок і дітей, а також постійно оновлює колекції відповідно до трендів.

За 6 років магазин зарекомендувало себе як надійний продавець, який піклується про комфорт своїх клієнтів. Компанія не зупиняється на досягнутому і продовжує розвиватися.

ФОП Синчук П.П. - це сучасний український магазин, який спеціалізується на продажу взуття для різних категорій споживачів. Заснований у 2019 році, підприємство пройшло шлях від невеликого магазинчика до повноцінного магазину.

Основна мета діяльності підприємства - це продаж комфортного та якісного взуття, яке відповідає сучасним трендам та тенденціям, яке враховує побажання споживачі. У своїй роботі компанія орієнтується на такі ключові принципи:

- **Висока якість** – співпраця з перевіреними постачальниками, та зарубіжними компаніями;
- **Доступність** – оптимальне співвідношення ціни та якості;
- **Клієнтоорієнтованість** – задоволення потреб покупців завдяки широкому асортименту та якісному обслуговуванню.

На сьогоднішній день підприємство ФОП «Синчук П.П.» продає сучасне взуття, яке відповідає міжнародним стандартам. Співпрацює з зарубіжними постачальниками у яких виробничі лінії оснащені новітнім обладнанням, що дозволить забезпечити високу продуктивність та якість виготовлення продукції.

Основні матеріали, що використовуються у виробництві, включають натуральну та еко-шкіру, міцну гумову та поліуретанову підошву, високоякісні текстильні матеріали.

Підприємство продає свою продукцію через власний магазин. ФОП Синчук П.П. має клієнтів по всій території України та активно працює над розширенням.

Компанія продовжує динамічно розвиватися, впроваджуючи нові технології та розширюючи асортимент. Основними стратегічними напрямками на найближчі роки є:

- Розширення та збільшення обсягів продажу продукції;
- Вихід на міжнародні ринки;
- Співпраця з постачальниками, які базують виробництво на нових матеріалах та екологічних технологіях у виробництві;
- Посилення онлайн-продажів та цифровізація бізнес-процесів;
- Розширення мережі фірмових магазинів у великих містах України.

Підприємство ФОП «Синчук П.П.» продовжує розвиватись, та зміцнювати свої позиції на ринку, пропонуючи якісну та доступну продукцію, яка відповідає вимогам споживачів.

Підприємство ФОП Синчук П.П. пропонує широкий вибір взуття для всіх вікових категорій та сезонів. Асортимент продукції включає:

- **Чоловіче взуття;**
- **Жіноче взуття;**
- **Дитяче взуття;**
- **Спеціалізоване взуття;**

В асортименті чоловічого та жіночого взуття у пропонуються різні стилі, які будуть відповідати сучасним трендам. Продукція виготовлюється з натуральної шкіри, еко-шкіри, текстилю та інших матеріалів, що забезпечить комфорт і

довговічність. Компанія регулярно оновлює сезонні колекції, орієнтуючись запити споживачів.

Дитяче взуття закупляється з урахуванням анатомічних особливостей дитячої стопи. Воно виготовляється з безпечних матеріалів, що сприяє здоровому розвитку стопи дитини. Асортимент включає дитячого взуття включає зимове взуття, літнє взуття та демісезонне.

1.2 Організаційна структура ФОП «Синчук П.П.»

Організаційна структура по своїй суті досить проста. Головним в підприємстві є **Генеральний директор**, який здійснює керівництво, приймає управлінські рішення та координує роботу всіх підрозділів. Структура зображена на рисунку 1.1.



Рисунок 1.1 – Верхній рівень організаційної структури

Під керівництвом генерального директора працюють такі основні відділи:

- **Бухгалтерія**, яку очолює **Головний бухгалтер**, відповідає за фінансовий облік, контроль витрат та податкову звітність.
- **Відділ продажу**, який відіграє ключову роль.

Організаційна структура передбачає тісну співпрацю між відділами, що дозволяє ефективно реалізовувати бізнес-процеси та стратегічне планування. Відділ продажу співпрацює з маркетологами які допомагають аналізувати ринок, з ІТ-відділом для моніторингу продаж, а також із бухгалтерією для контролю фінансових операцій.

Можна побачити, що організаційна структура чітко визначає відділи, кожен з яких виконує свої обов'язки та функції для того щоб досягти загальних бізнес-цілей.

Відділ продажів відіграє дуже важливу роль у забезпеченні прибутку та розвитку, тим, що координує роботу маркетологів, менеджерів. Також оптимізує бізнес-процеси, підвищує ефективність управління та забезпечує якісне обслуговування клієнтів.

Відділ продаж

Відділ продажу грає ключову роль яка забезпечує прибуток компанії та взаємодії з клієнтами. Головним є **Начальник відділу по продажах**, який керує всіма процесами, а також розробляє стратегії продажів та координує роботу підлеглих. На рисунку 1.2 відображено організаційну структуру відділу продаж.



Рисунок 1.2 – Організаційна структура відділу продаж

Структура відділу включає такі посади:

- **Старший менеджер по маркетингу** – який відповідає за аналіз ринку, розробку маркетингових стратегій та рекламу продукції або послуг.
- **Маркетологи** – займаються вивченням споживчого попиту, просуванням товарів і послуг, організацією рекламних кампаній та PR-активностей.
- **Заступник начальника по продажах** – допомагає керівнику відділу у реалізації стратегії продажу, контролює виконання планів та підтримує комунікацію між підрозділами.
- **Менеджер по продажах** – безпосередньо займається продажами, працює з клієнтами, проводить переговори та укладає контракти.
- **Менеджер по сервісу** – відповідає за післяпродажне обслуговування клієнтів, вирішення претензій та підтримку довготривалих відносин із покупцями.

Сервіс-менеджер – контролює якість сервісних послуг, взаємодіє з технічними спеціалістами та координує роботу сервісного обслуговування.

1.3 Стан автоматизації на ФОП «Синчук П.П.»

Автоматизація відіграє ключову роль, також автоматизація сприяє підвищенню ефективності управління та оптимізує усі бізнес-процеси. Основні переваги автоматизації включають:

Збільшення продуктивності

- Виконання завдань без участі людини, що прискорить та покращить робочі процеси.

Скорочення витрат

- Використовуючи автоматизовані системи дозволить зменшити витрати на персонал, тим самим мінімізує виникнення помилок, які можуть призвести до фінансових втрат.

Оперативність у прийнятті рішень

- Автоматизовані системи, дозволяють швидко обробляти інформацію та надавати актуальні дані, що в свою чергу дозволяє керівництву оперативно ухвалювати рішення.

Підтримка аналітики

- Автоматизація також забезпечує ефективний збір та обробку великих обсягів даних, що в свою чергу дозволить полегшити аналіз бізнес-процесів та допоможе в прийнятті обґрунтовані рішення.

Зміцнення конкурентних позицій

- Компанії, які використовують автоматизовані системи, краще реагують на зміни ринку, а також мають конкурентні переваги.

Обмежені ресурси та обсяг роботи не дозволяли працювати над автоматизацією, але я працювала над парсингом даних, для оптимізації вибірки даних, пошуку, наявності. Тому було дано завдання дослідити рівень та методологію оптимізації підприємства.

Слід зазначити що підприємство ФОП Синчук П.П. використовує власне програмне забезпечення для автоматизації та виконання поставлених завдань.

Приклад того що використовує підприємство:

- ERP система яку використовує підприємство куди вносять інформацію про продажі, приходи, витрати в систему, програма - генерує результат, а власник бізнесу на підставі цієї інформації приймає управлінські рішення.
- Ubuntu дану систему використовують для того щоб організувати обмін файлами, а також для комунікації між персоналом, включаючи обробки документів, інформації та даних.
- Lotus Notes систему використовують для організації корпоративного документообороту. Її призначення – це робота з інформацією і документами, тут можна знайти специфічні бази, сюди також входить структура електронної пошти, передачі повідомлень і комунікацій.

Десктопні додатки використовуються в підприємстві для локальної обробки та роботи з даними, що дає співробітникам взаємодіяти з даними та виконувати завдання безпосередньо на своїх робочих місцях.

Завдяки використанню цих систем працівники працюють в комплексі, даючи підприємству можливість ефективно керувати бізнес – процесами, також мати швидкий доступ до даних, їхню надійність та оптимізацію робочих процесів.

1.4 Розробка функціональної моделі існуючих бізнес – процесів

На Рисунок 1.3 зображено функціональну схему продажу взуття. CASE-засобів та UML діаграм, дають можливість моделювати всі сфери діяльності будь яких організацій та аналізувати їх роботу.

За допомогою програмного засобу [10]Bizagi Modeler було розроблено функціональну модель бізнес-процесів компанії, що дозволило проаналізувати та виявити інформаційні потоки, етапи виконання завдань і ролі учасників процесу. Цей інструмент надав можливість чітко описати та візуалізувати бізнес-процеси, що сприяло ефективному управлінню та оптимізації робочих потоків.

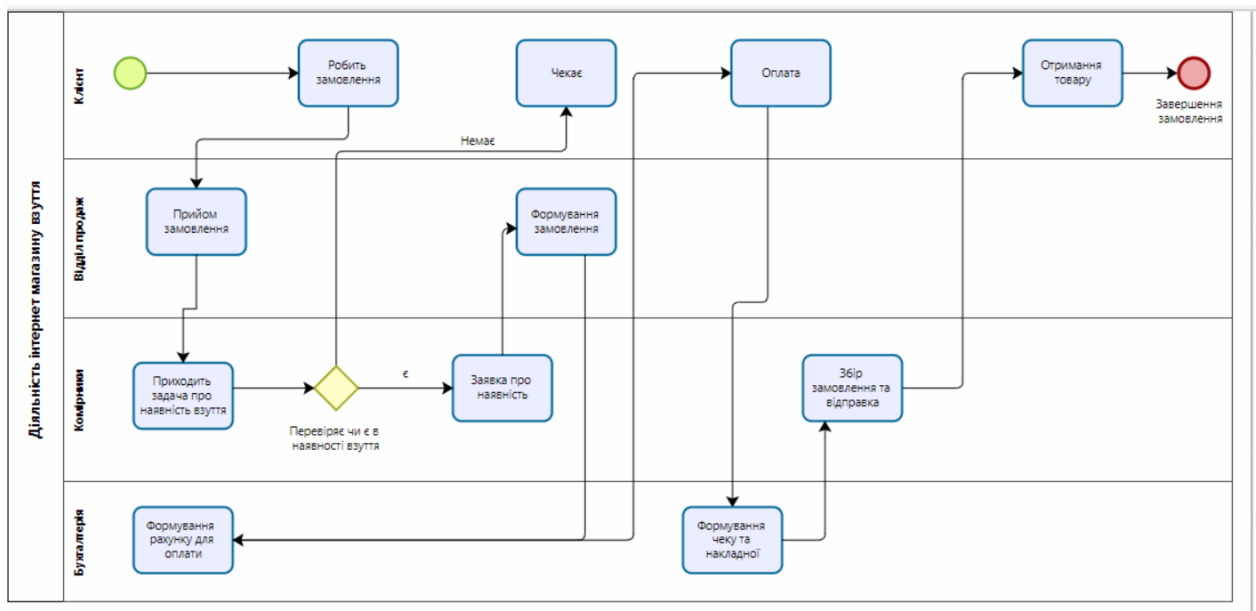


Рисунок 1.3 – Функціональна схема продажу взуття

1.4.1 Виявлені проблеми

У ФОП Синчук П.П. було виявлено відсутність автоматизованої системи управління продажами та обслуговування клієнтів. Головна проблема пов'язана з ручною обробкою замовлень, та зберіганням інформації про товари, клієнтів та продажі, це впливає на загальну організацію бізнес-процесів:

- прийом замовлень обробляється через телефонні дзвінки або месенджери;
- відсутній особистий кабінет клієнта - неможна переглянути історію замовлень;
- складно масштабувати бізнес — ручна робота обмежує кількість оброблених замовлень.

1.5 Огляд існуючих рішень для розв'язання виявлених проблем.

Для створення інтернет-магазину для продажу взуття було проаналізовано системи управління контентом, які підтримують функціонал електронної комерції. У цьому підрозділі розглянуто можливості платформ OpenCart, WooCommerce, PrestaShop та Magento.

1.5.1 OpenCart

OpenCart – безкоштовна CMS для інтернет-магазинів, що підходить для малого та середнього бізнесу [1].

- Дозволяє створювати необмежену кількість товарів і категорій.
- Інтеграція з популярними платіжними системами (PayPal, LiqPay, Stripe) та службами доставки.
- Великий вибір плагінів і тем для розширення функціоналу.
- SEO-інструменти для покращення видимості в пошукових системах.

- Регулярні оновлення та підтримка спільноти.

1.5.2 WooCommerce

WooCommerce – плагін для WordPress, що додає функції eCommerce [2].

- Простий у використанні та налаштуванні.
- Великий вибір безкоштовних і платних тем.
- Інтеграція з SEO-інструментами (Yoast SEO) та маркетинговими платформами (Google Ads, Facebook).
- Підтримка багатомовності та мультивалютності.
- Підключення платіжних сервісів (Monobank, Stripe, PayPal).

1.5.3 PrestaShop

PrestaShop – CMS для середніх магазинів із розширеним функціоналом [3].

- Гнучке управління товарами, варіаціями, знижками та акціями.
- Вбудована аналітика та фінансова звітність.
- Багатомовність та підтримка різних валют.
- Інтеграція з email-маркетингом, соціальними мережами та CRM.
- Підтримка HTTPS, SSL та регулярне оновлення безпеки.

1.5.4 Magento

Magento – потужна eCommerce-платформа для великих онлайн-магазинів [4].

- Висока продуктивність та масштабованість (100 000+ товарів).
- Гнучка кастомізація та інтеграція з ERP, CRM та API.

- Розширені SEO-налаштування та маркетингові функції.
- Потужна аналітика та звітність (Google Analytics, Power BI).
- Високий рівень безпеки та кіберзахисту.

1.5.5 Порівняльний аналіз платформ для розробки інтернет-магазинів

Зроблено порівняльну характеристику аналогів для розробки, що зображено в таблиці 1.1.

Таблиця 1.1 Порівняння систем-аналогів

Характеристика	OpenCart	WooCommerce	PrestaShop	Magento
Функціональність	Управління товарами, замовленнями	Інтеграція з WordPress, базовий eCommerce	Управління товарами, замовленнями користувачами	Потужний функціонал для великих магазинів
Інтеграції	Платіжні системи, маркетингові, CRM	WordPress плагіни, Google Analytics, Facebook Pixel	CRM, маркетинг-інструменти, платіжні системи	ERP, CRM, кастомні API, багатовалютність
Зручний інтерфейс	Частково	+	+	Частково
Аналітика та звітність	-	+	Частково	+
Гнучкість налаштувань	Частково	Частково	Частково	+
Безпека даних	+	+	+	+
Масштабованість	Низька	Середня	Низька	Висока
Вартість	\$30–50/місяць + \$50–300 (розширення) + \$0–500 (кастомізація)	\$20–30/місяць + \$50–200 (платні плагіни) + \$0–500 (налаштування)	\$30–50/місяць + \$100–500 (модулі та теми) + \$200–1000 (кастомізація)	\$50–500/місяць + \$500–5000 (розширення та розробка) + \$22 000+/рік (Adobe Commerce)

1.6 Огляд майбутньої системи

На основі аналізу конструкторів, мов програмування та витрат (таблиця 1.2), було прийняте рішення, розробити інтернет - магазин взуття ФОП Синчук П.П. на [5]C# ASP.NET Core MVC, який дозволить клієнтам переглядати каталог, оформлювати замовлення, а адміністраторам – керувати товарами та замовленнями.

Основні функції повинні включати:

- **Користувачі:** реєстрація, авторизація, перегляд каталогу, кошик, оформлення замовлення.
- **Адміністратори:** управління товарами, замовленнями, категоріями, акціями.
- **Додаткові функції:** SEO-оптимізація, адаптивний дизайн, інтеграція з платіжними та доставочними сервісами.
- **Відсутність платних плагінів** – у CMS (WooCommerce, OpenCart) доведеться купувати модулі (\$50–500).
- **Гнучкість та масштабованість** – можна додавати унікальні функції, інтегрувати CRM та API.
- **Вища продуктивність** – ASP.NET Core швидше обробляє запити, ніж PHP-основні CMS.
- **Безпека** – вбудовані механізми захисту (аутентифікація, захист від атак).
- **Економія у довгостроковій перспективі** – відсутність щомісячних витрат на плагіни.

Порівняння витрат продемонстровано в таблиці 1.2.

Таблиця 1.2. Порівняння витрат

Витрати	OpenCart WooCommerce	ASP.NET Core MVC
Хостинг	\$5–50/місяць	\$10–30/місяць
Платні плагіни	\$100–500/рік	\$0 (кастомні рішення)
Оптимізація SEO	\$50–200	Вбудовано
Безпека	\$100–300	Вбудована
Загальні витрати	\$500–2000/рік	\$300–1000

1.7 Розрахунок економічного ефекту впровадження інтернет – магазину

Розробка інтернет – магазину для продажу взуття дозволить автоматизувати процес продажу взуття, та керувати асортиментом, обробкою замовлень, доставку, що підвищить ефективність роботи підприємства.

Фактори економічного ефекту:

- Зниження витрат на управління даними про асортимент і замовлення.
- Скорочення часу на обробку замовлень та логістику.
- Підвищення рівня прозорості у взаємодії із постачальниками.
- Зменшення витрат на складське управління.
- Покращення аналітики та звітності для планування продажів.
- Вихідні дані для розрахунку

Визначаємо ознаку – управління інтернет магазином взуття.

Ступінь новизни розроблюваних задач – «В» - використання типових проектних рішень за умови їх змін.

Група складності алгоритму 2.

Таблиця 1.3. Дані для вхідної та вихідної інформації інтернет - магазину взуття

Вид інформації	Позначення	К-сть наборів даних
Змінна інформація	ЗІ	m=10
Нормативно–довідкова інформація	НДІ	n=3
Банк(база) даних	БД	p=1
Обробка в режимі реального часу	РЧ	Так
Забезпечення телекомунікаційної обробки даних і управління віддаленими об'єктами	ТОУ	ні

Таблиця 1.4.Визначення витрат часу

Вид системи	Стадія розробки системи	
	Ескізний проект	Технічне завдання
	В	В
Управління взуттям. Управління технічним обслуговуванням, управління складом.	$T_1 = 80$	$T_2 = 50$

Визначаємо базове значення витрат часу для стадій «Технічний проект», «Робочий проект», «Впровадження».

Вхідними даними для визначення є :

- кількість форм вхідної інформації - 10,
- кількість форм вихідної інформації - 3,
- Базове значення витрат часу для стадій «Технічний проект»: $T_{B3} = 150$
- Базове значення витрат часу для стадій «Робочий проект»: $T_{B4} = 220$
- Базове значення витрат часу для стадій «Впровадження»: $T_{B5} = 65$

Розрахунок витрат часу для стадії «Технічний проект»(T_3):

Коефіцієнт трудомісткості робіт k_{Π} визначається за формулою:

$$k_{\Pi} = \frac{k_1 * m + k_2 * n + k_3 * p}{m + n + p}$$

де

k_1, k_2, k_3 – поправочні коефіцієнти, що наведені в табл.3,

m – кількість видів змінної інформації (ЗІ),

n – кількість видів нормативно-довідкової інформації (НДІ),

p – кількість видів бази даних (БД),

m, n, p наведені у таблиці 1.5.

Таблиця 1.5. Коефіцієнти k_1, k_2, k_3 для стадії «Технічний проект»

Вид використаної інформації	Ступінь новизни
	В
k_1 (ЗІ)	1.0
k_2 (НДІ)	0.8
k_3 (БД)	2.0

Таблиця 1.6. Коефіцієнти ступеню новизни проекту

Стадія розробки системи	Вид обробки	Ступінь новизни
		В
Технічний проект	РЧ	1.26
Робочий проект	РЧ	1.32
Впровадження	РЧ	1.21

Розрахуємо $k_{\Pi} = (1 * 10 + 0.8 * 3 + 2.0 * 1) / (10 + 3 + 1) = 1.057$

Витрати часу для стадії «Технічний проект» T_3 розраховуються за формулою:

$$T_3 = T_{Б3} * k_{\Pi} * k_o$$

$$T_3 = 150 * 1.057 * 1.26 = 199 \text{ люд.-днів.}$$

Розрахунок витрат часу для стадії «Робочий проект (T_4)»

Для розрахунку витрат часу на стадії «Робочий проект використовують формулу: $T_4 = T_{Б4} * k_{\Pi} * k_o * k_c$

Таблиця 1.7. Коефіцієнти k_1, k_2, k_3 для стадії “Робочий проект”

Вид використаної інформації	Ступінь новизни
$k_1(ЗІ)$	1.1
$k_2(НДІ)$	0.7
$k_3(БД)$	0.6

Розрахуємо $k_{\Pi} = (1.1 * 10 + 0.7 * 3 + 0.6 * 1) / (10 + 3 + 1) = 1.04$

Отже вирахуємо T_4 за формулою (3)

$T_4 = 220 * 1.04 * 1.32 * 1.6 = 484$ люд.-днів.

Розрахунок витрат часу для стадії “Впровадження (T_5)”

Поправочні коефіцієнти мають такі ж значення як і при обрахунку T_4 . Для стадії визначення загальних витрат часу на “Впровадження” T_5 (люд.-днів).

$T_5 = 65 * 1.04 * 1.21 * 1.6 = 130$ люд.-днів.

Таким чином, загальні витрати людської праці на проектування системи складають : $T_{\text{заг}} = 80 + 50 + 199 + 484 + 130 = 943$ люд.-днів.

Розрахунок загальних витрат

Для курсового проекту кількість робочих годин складає 530 із 7 годинним робочим днем , тому на розробку проекту виділено Φ днів: $\Phi = 530 / 7 = 75$

Кількість місяців на розробку , M : $M = \Phi / 25 = 3$

Отже для виконання такого проекту потрібно таку чисельність виконавців \mathcal{C} , яка обрховується за формулою : $\mathcal{C} = T_{\text{заг}} / \Phi = 943 / 75 = 13$

Якщо прийняти, що оплата програміста здійснюється в розмірі 25 000 грн, то оплата праці всіх виконавців складе : $V_1 = \mathcal{C} * M * Z_{\text{пр}} = 13 * 3 * 25000 = 975 000$ грн.

Витрати пов’язані з розробкою програми на ПК.

Розрахунок річного фонду часу роботи ПК

Дісний річний фонд часу Пк у годинах дорівнює числу робочих годин у році для оператора , за винятком часу на технічне обслуговування і ремонт ПК (в середньому 5 год /міс + 6 роб.днів/рік)

$$T_{\text{ПК}} = T_{\text{оп}} - (6 \cdot 8 + 5 \cdot 12)$$

$$T_{\text{ПК}} = 2000 - (6 \cdot 8 + 5 \cdot 12) = 1892 \text{ год}$$

Оскільки під час виконання курсового проекту студент витрачає 450 год. машинного часу , то величина фонду часу ПК : $T'_{\text{ПК}} = T_{\text{ПК}} \cdot K / T_{\text{оп}} = 1892 \cdot 450 / 2000 = 452.7$.

Балансована вартість ПК вираховується за формулою

$$Ц_{\text{ПК}} = Ц_{\text{р}} \cdot (1 + k_{\text{ун}})$$

$Ц_{\text{р}}$ ринкова вартість ПК, орієнтовно складає 40000 грн , $k_{\text{ун}}$ — коефіцієнт , що враховує витрати на установку і налагодження ПК і дорівнює 0.12. Отже $Ц_{\text{ПК}} = Ц_{\text{р}} \cdot (1 + k_{\text{ун}}) = 40000 \cdot (1 + 0.12) = 44800$ грн

Амортизаційні відрахування використання ПК, $З_{\text{ам}}$, обчислюється за формулою : $З_{\text{ам}} = Ц_{\text{ПК}} / N_{\text{а}}$, де $N_{\text{а}}$ - норма амортизаційних відрахувань , яка для ПК = 5. $З_{\text{ам}} = 44800 / 5 = 8960$ грн.

Витрати на електроенергію , споживану ПК , визначають за формулою :

$$З_{\text{ел}} = P_{\text{ПК}} \cdot T_{\text{ПК}} \cdot Ц_{\text{ел}} \cdot A$$

де $P_{\text{ПК}}$ – потужність ПК;

$T_{\text{ПК}}$ – фонд корисного часу роботи ПК;

$Ц_{\text{ел}}$ – вартість 1кВт електроенергії;

A – коефіцієнт інтенсивного використання ПК.

$$P_{\text{ПК}} = 0.5 \text{ кВт},$$

$$T_{\text{ПК}} = 435.16 \text{ год}$$

$$Ц_{\text{ел}} = 1.68 \text{ грн /кВт}$$

$$A = 0.9$$

$$\text{Отже } З_{\text{ел}} = 329 \text{ грн}$$

$З_{\text{р}}$ — витрати на поточний ремонт і тех обслуговування ПК визначаються як 6% від балансової вартості ПК ($Ц_{\text{ПК}}$): $З_{\text{р}} = Ц_{\text{ПК}} \cdot 0.06 = 44800 \cdot 0.06 = 2688$ грн

З_{мат} — непрямі витрати, пов'язані з експлуатацією Пк, визначаються як 5 % від балансової вартості Пк Цпк $Z_{\text{мат}} = 44800 * 0.05 = 2240$ грн

Заробітна плата обслуговуючого персоналу складає в середньому – 10000 грн.

Поточні витрати на експлуатацію V1'' визначаються за формулою :

$$V''_1 = Z_{\text{ОП}} + Z_{\text{АМ}} + Z_{\text{ЕЛ}} + Z_{\text{Р}} + Z_{\text{МАТ}}$$

Отже $V1'' = 10000 + 8960 + 329 + 2688 + 2240 = 24\ 217$ грн.

Отже загальні витрати на розробку програмного забезпечення комп'ютерної системи розраховуються за формулою $V1 = V'1 + V1'' = 975\ 000 + 24\ 217 = 999\ 217$ грн.

Витрати на придбання і установку Пк V2 .

Витрати на придбання і установку ПК визначаються за формулою $V2 = \text{Цпк} = 50000$ грн.

Витрати на підготовку приміщення V3

Ці витрати залежать від стану приміщення, де буде встановлюватись Пк. Так як пристосоване приміщення є то $V3 = 0$;

Витрати на навчання персоналу V4 :

В середньому навчання персоналу триватиме 1 місяць, тому можна вважати, що $V4 = 4500$

Загальна вартість розробки і впровадження системи.

Загальна вартість розробки і впровадження системи вираховується за формулою : $V_{\text{заг}} = V1 + V2 + V3 + V4 = 999217 + 50000 + 0 + 4500 = 1\ 053\ 717$ грн.

Оскільки норма амортизаційних втрат для комп'ютерних систем $N_a = 5$, то для обрахування річного економічного ефекту слід брати до розгляду величину $V_p = V_{\text{заг}} / N_a = 1\ 053\ 717 / 5 = 210\ 743$ грн.

Річний прибуток P_p від впровадження системи буде досягнуто за рахунок підвищення якості виробів і орієнтовно складатиме 80 000 грн на рік.

Коефіцієнт економічної ефективності розробки вираховується за формулою : $K_{\text{еф}} = P_p / V_p = 80\ 000 / 210743 = 0.37$

Термін окупності розробки визначається за формулою: $T_{\text{ок}} = 1 / K_{\text{еф}} = 1 / 0.37 = 2,7$ роки.

Таким чином, термін окупності інформаційної системи буде 2,7 роки.

1.8 Обґрунтування й доцільності проєктування й розроблення інтернет – магазину для продажу взуття

На основі проведеного аналізу конструкторів, функціональних схем, економічного ефекту, діаграм, мов програмування та витрат, було прийняте рішення, розробити інтернет - магазин взуття ФОП Синчук П.П. на C# ASP.NET Core MVC, який дозволить клієнтам переглядати каталог, оформлювати замовлення, а адміністраторам – керувати товарами та замовленнями.

Основні функції повинні включати:

- Користувачі: реєстрація, авторизація, перегляд каталогу, кошик, оформлення замовлення.
- Адміністратори: управління товарами, замовленнями, категоріями.
- SEO-оптимізація, адаптивний дизайн.

РОЗДІЛ 2. ТЕХНІЧНЕ ЗАВДАННЯ НА ПРОЄКТУВАННЯ

2.1. Загальні положення

2.1.1. Найменування веб-магазину «Інтернет – магазин взуття ФОП «Синчук П.П.»

2.1.2. Результати робіт зі створення Інтернет – магазин оформлюються згідно з вимогами ДСТУ на відповідні етапи розробки. Порядок оформлення і передачі результатів визначається змістом і календарним планом виконання розробки.

2.1.3. У випадку необхідності на наступних стадіях робіт по створенню інтернет – магазин окремі положення можуть уточнюватися і розвиватися.

2.2. Призначення і цілі створення системи

2.2.1 Призначення системи

Інтернет – магазин взуття ФОП Синчук П.П. призначений для автоматизації та покращення процесів роздрібного продажу взуття і обслуговування клієнтів. Інтернет – магазин покликана забезпечити ефективне управління товарним асортиментом, обробкою замовлень клієнтів та супутніми операціями (оплата, доставка), а також підвищити оперативність і точність виконання торговельних процесів.

2.2.2. Цілі створення системи

Основною метою створення інтернет – магазину є підвищення ефективності продажів та якості обслуговування клієнтів інтернет-магазину взуття ФОП Синчук П.П. шляхом автоматизації і оптимізації бізнес-процесів. повинен забезпечувати швидке та точне оформлення замовлень через веб-сайт, ефективне управління товарними запасами, формування аналітичних звітів для підтримки управлінських рішень, а також розширення клієнтської бази за рахунок онлайн-каналу продажів. Важливим аспектом є також забезпечення безпеки даних клієнтів та покращення загального досвіду користувачів.

2.3. Характеристика об'єкта автоматизації

2.3.1. Короткі відомості про об'єкт автоматизації

Об'єктом автоматизації є підприємство ФОП Синчук П.П., яке здійснює роздрібну торгівлю взуттям (чоловічим, жіночим та дитячим) у місті Київ.

Діяльність ФОП «Синчук П.П.» спеціалізується на продажу якісного взуття різних виробників та забезпеченні високого рівня обслуговування клієнтів. Завдяки розширенню асортименту і зростанню обсягу продажів, виникла необхідність впровадження інформаційної системи (інтернет-магазину) для автоматизації та покращення торговельних процесів на підприємстві.

2.3.2. Основні характеристики діяльності ФОП Синчук П.П.:

- ФОП Синчук П.П. здійснює роздрібний продаж взуття різних типів та брендів, включаючи чоловіче, жіноче й дитяче взуття.
- Продаж товарів наразі відбувається у фізичній торговельній точці, проте планується розширення діяльності через інтернет-магазин.
- Асортимент продукції регулярно оновлюється відповідно до попиту та містить взуття для різних сезонів і випадків.
- Магазин працює щоденно (наприклад, з 9:00 до 19:00), забезпечуючи доступність товарів для клієнтів у будні та вихідні дні.
- Персонал включає власника (адміністратора системи) та продавців-консультантів; бухгалтерський облік може виконуватися власником або окремим бухгалтером.
- Облік товарних запасів та продажів досі вівся вручну або за допомогою електронних таблиць, що ускладнює отримання актуальної інформації про залишки на складі.
- Підприємство прагне підвищити конкурентоспроможність і охопити ширшу аудиторію шляхом впровадження онлайн-продажів, розширюючи географію збуту поза межі фізичного магазину.

2.4. Вимоги до системи

2.4.1. Вимоги до системи в цілому

2.4.1.1. Вимоги до структури і функціонування системи

2.4.1.1.1. Система повинна мати клієнт-серверну архітектуру з використанням єдиної централізованої бази даних. В контексті інтернет-магазину взуття це означає, що інформаційна система включає сервер бази даних (наприклад, MS SQL Server), до якого підключаються різні компоненти через мережу: веб-

інтерфейс для клієнтів (онлайн-вітрина магазину) та адміністративний інтерфейс для персоналу магазину. Така архітектура забезпечить узгоджений доступ до даних товарів, замовлень і клієнтів в реальному часі з різних робочих місць (комп'ютер адміністратора, пристрої клієнтів тощо).

2.4.1.1.2. Функціонування системи має передбачати діагностику відхилень від нормального режиму роботи, виявлення помилок програмного забезпечення або збоїв технічних засобів. Про виявлені збої чи помилки система повинна сповіщати адміністратора для оперативного втручання та усунення проблем. Взаємодія між підсистемами (клієнтською частиною сайту та серверною частиною) повинна здійснюватися через спільну базу даних і стандартні мережеві протоколи обміну інформацією.

2.4.1.1.3. Розвиток і модернізація системи повинні проводитися з урахуванням змін у функціональних вимогах та оновлення технічних і програмних засобів. Архітектура та програмні рішення системи мають бути гнучкими й масштабованими, здатними до розширення функціоналу і обробки зростаючих обсягів інформації. Забезпечення сумісності програмного та інформаційного середовища має здійснюватися за рахунок використання стандартизованих протоколів передачі даних та єдиних систем класифікації і кодування.

2.4.1.1.4. Система повинна забезпечувати можливість діалогової та розподіленої обробки даних у режимі реального часу. Це дозволить персоналу магазину і клієнтам одночасно взаємодіяти з системою (наприклад, оформлювати замовлення, оновлювати інформацію про товари) без конфліктів і затримок, з оперативним оновленням даних на всіх робочих місцях.

2.4.1.2. Вимоги до чисельності і кваліфікації персоналу

2.4.1.2.1. Персонал, що використовує інформаційну систему, повинен мати навички роботи на персональному комп'ютері та пройти навчання щодо використання системи. Користувачі системи повинні дотримуватися технологічних інструкцій і правил безпеки при роботі з програмним забезпеченням.

2.4.1.2.2. У ФОП Синчук П.П. користувачами системи можуть бути власник

(адміністратор системи), менеджери з продажу або товарознавці, а також бухгалтер (за потреби). Кожному користувачу внутрішньої системи призначається унікальний обліковий запис з паролем, який визначає рівень його доступу. Залежно від ролі (адміністратор, менеджер тощо) користувач матиме повний або обмежений доступ до функцій системи. Покупці (клієнти) інтернет-магазину взаємодіятимуть із системою через публічний веб-інтерфейс, маючи доступ лише до власних даних та функцій оформлення замовлення.

2.4.1.3. Показники призначення

2.4.1.3.1. Показники призначення системи повинні характеризувати ступінь та якість автоматизації торговельної, облікової та управлінської діяльності магазину взуття для його оптимального функціонування. Перелік і допустимі значення таких показників, при яких зберігається цільове призначення системи, визначаються на стадії технічного проектування.

2.4.1.3.2. Система повинна мати можливості налаштування параметрів підприємства та інтеграції з периферійним обладнанням (наприклад, сканерами штрих-кодів, платіжними терміналами) при модернізації та розвитку, а також зміни бізнес-процесів та методів організаційного управління.

2.4.1.4. Вимоги до надійності

2.4.1.4.1. Система є багатофункціональною і повинна працювати безперервно (онлайн-частина – 24/7 для клієнтів, адміністративна частина – протягом робочого дня підприємства). Всі функції системи мають виконуватися надійно і коректно. Надійність роботи оцінюється для кожної основної функції окремо; ключовими показниками надійності є безвідмовність виконання операцій та доступність системи для користувачів у потрібний час.

2.4.1.4.2. Технічне забезпечення системи повинно передбачати можливість запуску та виконання функціональних задач з різних пристроїв користувачів (робочих станцій адміністратора, комп'ютерів чи смартфонів клієнтів). Рекомендується використання джерела безперебійного живлення і стабільного інтернет-з'єднання для серверного обладнання, щоб мінімізувати простой системи у разі перебоїв електропостачання або зв'язку.

2.4.1.5. Вимоги до безпеки

Для забезпечення безпеки при експлуатації, налагодженні, монтажі, обслуговуванні і ремонті технічних засобів системи необхідно дотримуватися вимог відповідних державних стандартів та норм. Робочі місця та обладнання, задіяні в роботі системи, повинні відповідати чинним санітарним, електротехнічним та протипожежним нормам.

2.4.1.6. Вимоги з ергономіки та технічної естетики

Загальні ергономічні та естетичні вимоги до системи повинні відповідати державним стандартам. Освітленість і організація робочого місця адміністратора, розміщення дисплеїв та іншого обладнання мають відповідати нормам, забезпечуючи зручність і безпеку роботи. Інтерфейс програми (веб-сайту та адміністраторської панелі) повинен бути інтуїтивно зрозумілим, забезпечувати легку навігацію, привабливий дизайн та єдину стилістику оформлення, що сприяє зручності використання системи користувачами.

2.4.1.7. Вимоги до експлуатації, технічного обслуговування, ремонту і зберігання компонентів системи

2.4.1.7.1. Система повинна експлуатуватися та обслуговуватися відповідно до вимог, визначених чинними стандартами (наприклад, ДСТУ 3576-97 щодо експлуатації автоматизованих систем).

2.4.1.7.2. Розміщення технічних засобів системи (серверного обладнання, мережевих пристроїв) повинно відповідати вимогам будівельних норм і правил (ДБН В.2.2-9-2009). Для серверного обладнання слід передбачити стабільне електроживлення: напруга 220/380 В, частота (50 ± 1) Гц, допустиме відхилення напруги від +10% до -15%. Рекомендується використовувати джерело безперебійного живлення (UPS) для забезпечення роботи серверу при короткочасних перебоях (до 0,001 с).

2.4.1.7.3. Кількість, кваліфікація та режим роботи персоналу, який обслуговує систему, повинні відповідати вимогам, зазначеним у технічних умовах та інструкціях з експлуатації (тобто персонал технічної підтримки має володіти необхідними навичками і працювати за встановленим графіком, достатнім для

підтримки безперервної роботи системи).

2.4.1.7.4. Склад, розміщення і умови зберігання компонентів системи (комп'ютерного обладнання, носіїв даних, резервних копій) повинні відповідати рекомендаціям документації на ці компоненти. Обладнання серверу слід розташувати у провітрюваному приміщенні з належним рівнем температури і вологості, а резервні носії інформації зберігати у захищених умовах, що унеможливають їх пошкодження.

2.4.1.7.5. Регламент технічного обслуговування системи повинен враховувати рівень і умови роботи системи та забезпечувати, у разі відмови, можливість аварійного (резервного) режиму роботи до повного відновлення функціонування.

2.4.1.8. Вимоги до захисту інформації від несанкціонованого доступу

- Для забезпечення безпеки та цілісності даних необхідно використовувати комплекс засобів захисту:

- операційну систему серверу з вбудованими засобами захисту (контроль доступу, шифрування тощо) та міжмережеві екрани (фаєрволи) для захисту локальної мережі;

- засоби захисту, що надаються системою управління базами даних (наприклад, MS SQL Server), з можливістю використання тригерів, процедур, функцій для контролю цілісності даних, а також механізмів розмежування доступу (створення користувачьких ролей і прав).

- Кожен сеанс роботи з системою повинен починатися з введення користувачем свого індивідуального логіна і пароля. Система автентифікації повинна забезпечувати перевірку прав доступу користувача до відповідних функцій.

- Система керування паролями повинна підтримувати вимоги щодо надійності паролів та, за можливості, їх періодичної зміни (можна використовувати вбудовані механізми платформи .NET для автентифікації і авторизації користувачів).

- Кожному співробітнику підприємства надається персональний

обліковий запис; доступ до критично важливих даних (наприклад, таблиць бази даних з фінансовою інформацією) має бути обмежений і захищений від несанкціонованого редагування чи видалення.

2.4.1.9. Вимоги щодо збереження інформації при аварійних ситуаціях

2.4.1.9.1. Слід передбачити засоби резервного копіювання бази даних після критичних оновлень інформації та можливість відновлення даних з резервної копії у разі збою або пошкодження основної БД. В системі повинна бути реалізована регулярна процедура створення резервних копій (наприклад, щоденно або щотижнево, залежно від інтенсивності оновлення даних).

2.4.1.9.2. Резервні копії бази даних рекомендується зберігати на різних носіях чи в різних місцях (наприклад, на окремому фізичному носії, хмарному сховищі або віддаленому сервері) для підвищення надійності збереження даних. У разі виходу з ладу основного накопичувача це забезпечить можливість відновлення інформації з альтернативного джерела.

2.4.1.10. Вимоги до захисту від зовнішніх впливів

2.4.1.10.1. Електромагнітні випромінювання від обладнання системи повинні відповідати встановленим нормам: рівень електромагнітного поля на робочих місцях не має перевищувати $0,3 \text{ В/м}^2$ у діапазоні частот від 0,15 до 300 МГц. Для захисту технічних засобів від зовнішніх електромагнітних завад слід застосовувати екранування та фільтри, передбачені стандартами.

2.4.1.10.2. Необхідно врахувати вплив інших зовнішніх шкідливих факторів (пилу, вологості, вібрації) на обладнання системи. Засоби захисту від цих факторів повинні бути застосовані згідно з чинними нормами і правилами. Обладнання та захисні пристрої повинні відповідати стандартам стійкості до зовнішніх впливів (наприклад, ДСТУ 2506-94), що гарантує надійну роботу системи в типових умовах експлуатації.

2.4.1.11. Вимоги до патентної чистоти

Система повинна бути патентно чистою, тобто її реалізація не порушуватиме чинних патентів та авторських прав. При розробленні слід використовувати ліцензоване програмне забезпечення та відкриті бібліотеки з дотриманням умов їх

ліцензій. Це забезпечить відсутність претензій з боку правовласників на використанні технології або компоненти системи.

2.4.2. Вимоги до функцій системи

Нижче наведені основні функції, які повинна виконувати інформаційна система інтернет-магазину взуття, із зазначенням їхніх входів, виходів та опису логіки роботи:

Управління каталогом товарів:

- Вхідна інформація: дані про товари, що вводяться адміністратором (найменування, категорія, бренд, розміри, ціна, кількість на складі, описи, фотографії тощо).

- Вихідна інформація: оновлений каталог товарів, доступний покупцям на веб-сайті (список товарних позицій з актуальними характеристиками та станом наявності); підтвердження успішного додавання або редагування товару для адміністратора.

- Опис: Дана функція дозволяє адміністратору системи додавати нові товарні позиції, редагувати інформацію про існуючі товари або вилучати товари з асортименту. Система зберігає всі зміни в базі даних. Після внесення змін відповідна інформація одразу відображається в онлайн-каталозі для користувачів. Система повинна забезпечувати валідацію введених даних (наприклад, контроль обов'язкових полів, недопущення введення некоректних значень цін) та підтримувати прикріплення зображень товарів.

Пошук і перегляд товарів:

- Вхідна інформація: запит користувача у вигляді ключових слів для пошуку або вибрані фільтри (категорія взуття, розмір, ціновий діапазон тощо).

- Вихідна інформація: список товарів, що відповідають критеріям пошуку/фільтрації, відображений на веб-сторінці; детальна інформація про обраний товар (опис, характеристики, фото, ціна, наявність).

- Опис: Ця функція реалізує інтерфейс інтернет-магазину для покупців, дозволяючи їм знаходити потрібні товари. Система обробляє введений текст пошукового запиту або обрані параметри фільтра та виконує вибірку відповідних

товарів з бази даних. Результати пошуку відображаються у вигляді каталогу товарів. Покупець може переглянути детальну сторінку товару, яка містить повну інформацію про продукт. Функція повинна працювати швидко навіть при великій кількості товарів, щоб забезпечити користувачеві зручність і оперативність у виборі продукції.

Оформлення замовлення покупцем:

- Вхідна інформація: дані, що вводяться клієнтом при оформленні замовлення: вибрані товарні позиції та їх кількість (формується у «кошик»), контактна інформація покупця (ім'я, адреса доставки, номер телефону, email), спосіб оплати (онлайн-оплата карткою, післяплата тощо) та спосіб доставки.

- Вихідна інформація: сформоване замовлення в базі даних (зі статусом «нове» або «очікує підтвердження»), унікальний номер замовлення; повідомлення користувачу про успішне прийняття замовлення (відображення сторінки підтвердження, можливо відправка підтвердження на email).

- Опис: Ця функція забезпечує процес онлайн-купівлі. Клієнт, вибравши товари, переходить до оформлення замовлення, заповнює необхідні дані та підтверджує замовлення. Система перевіряє коректність введеної інформації (наприклад, чи заповнені усі обов'язкові поля, чи є товари наявними у зазначеній кількості). Після підтвердження система реєструє замовлення: генерує йому унікальний ідентифікатор, фіксує склад замовлення (список товарів і кількість), обраховує суму до сплати, зберігає дані клієнта та вибрані методи доставки/оплати. Якщо обрано онлайн-оплату, система перенаправляє клієнта на захищену сторінку платіжного шлюзу для введення платіжних реквізитів. Після успішної обробки замовлення користувачу відображається повідомлення з деталями замовлення і подальшими інструкціями (наприклад, щодо відстеження доставки).

Обробка замовлень адміністратором:

- Вхідна інформація: дані нових замовлень, що зберігаються в системі і очікують обробки; дії адміністратора щодо кожного замовлення (підтвердження наявності, зміна статусу, внесення інформації про відправку, номер накладної тощо).

- Вихідна інформація: оновлений статус замовлення (наприклад, «підтверджено», «виконано», «відправлено») у базі даних; повідомлення клієнту про зміну статусу замовлення (може реалізуватися через email або особистий кабінет користувача на сайті).

- Опис: Ця функція виконується працівниками підприємства (адміністратором або менеджером) і охоплює весь цикл обробки отриманих через сайт замовлень. Адміністратор через інтерфейс системи переглядає список нових замовлень, перевіряє деталі кожного з них. Система повинна дозволяти змінювати статус замовлення (напр., «в обробці», «відправлено», «завершено»), редагувати інформацію про замовлення при необхідності (напр., якщо клієнт попросив змінити адресу доставки), друкувати супровідні документи (накладні, рахунки) та реєструвати інформацію про доставку (напр., номер поштової декларації). Після відправлення товару адміністратор змінює статус замовлення на «відправлено», і система може автоматично сповістити клієнта. Функція повинна забезпечувати зручний контроль за всіма відкритими замовленнями і дозволяти отримувати актуальну інформацію про їх виконання.

Управління даними клієнтів:

- Вхідна інформація: реєстраційні дані від нового клієнта (ім'я, електронна пошта, пароль тощо); дані для входу (логін/пароль) від існуючого клієнта; зміни у профілі клієнта (адреса, телефон, пароль) або дії адміністратора щодо облікового запису (блокування/розблокування, зміна статусу).

- Вихідна інформація: створений запис клієнта в системі (при реєстрації) або результат автентифікації (успішний вхід чи повідомлення про помилку); оновлені дані профілю клієнта; список замовлень клієнта в особистому кабінеті; для адміністратора – актуальний реєстр клієнтів.

- Опис: Система дозволяє новим користувачам зареєструватися, створивши особистий кабінет, де зберігатиметься їхня контактна інформація та історія замовлень. При реєстрації система перевіряє унікальність логіну (e-mail) і надійність паролю, зберігає пароль у зашифрованому вигляді. Авторизація клієнтів забезпечує захищений доступ до персонального розділу (за допомогою платформи

.NET Identity або аналогічної). Клієнти можуть переглядати і оновлювати свої дані (адреси доставки, контактні телефони) та переглядати статус і історію власних замовлень. Адміністратор системи має можливість переглядати список зареєстрованих клієнтів, актуалізувати їх статус (наприклад, призначити рівень знижки або заблокувати обліковий запис у разі порушень) відповідно до політики підприємства.

Формування аналітичної звітності:

- Вхідна інформація: параметри запиту звіту, що вводяться адміністратором (тип звіту, період часу, фільтри за категоріями товарів чи клієнтами тощо).

- Вихідна інформація: згенерований звіт, що відображається на екрані або експортується у файл (таблиця або діаграма з ключовими показниками: загальна сума продажів за період, кількість замовлень, найпопулярніші товари, залишки товарів на складі, фінансові показники тощо).

- Опис: Система повинна надавати засоби для отримання управлінської інформації. Адміністратор має можливість формувати різноманітні звіти для аналізу діяльності магазину. При виборі типу звіту і параметрів система виконує вибірку потрібних даних з бази та обчислює показники. Наприклад, звіт з продажів за місяць може містити сумарний дохід, кількість проданих пар взуття, структуру продажів за категоріями товарів тощо. Звіт зі стану складу може показувати залишки товарів та товари, що потребують поповнення. Генерація звітів має відбуватися протягом прийняттого часу і не повинна сильно навантажувати систему при великому обсязі даних. Отримані звіти можуть використовуватися керівництвом для прийняття рішень (закупівля нового товару, планування акцій, оцінка попиту).

2.5. Склад і зміст робіт по створенню системи

2.5.1. Стадії створення системи і терміни виконання робіт (табл. 2.1).

Таблиця 2.1 Стадії створення системи

№ п/п	Найменування робіт	Строки виконання робіт
1	Системний аналіз торгової діяльності підприємства	28.04.2025
2	Розроблення функціональної моделі системи	03.05.2025
3	Аналіз систем-аналогів (існуючих інтернет-магазинів взуття)	05.05.2025
4	Вибір технологій та середовища розробки системи (.NET, MS SQL Server, Visual Studio 2022)	08.05.2025
5	Розробка логічної та фізичної моделі бази даних	10.05.2025
6	Створення бази даних в середовищі MS SQL Server	15.05.2025
7	Розробка інтерфейсу користувача (веб-сайту і адмін-панелі)	20.05.2025
8	Написання інструкції користувача	30.05.2025

2.6. Порядок контролю і приймання системи

2.6.1. Введення в дію інтернет – магазину на ФОП Синчук П.П.:

При введенні в дію система повинна пройти приймальні випробування відповідно до вимог ДСТУ 3974-2000.

2.6.2. Випробування працездатності та рішення про прийняття системи в дослідну експлуатацію:

Розробник разом із замовником проводять випробування для визначення працездатності системи на реальних даних підприємства.

Програму і методику випробувань складає розробник та затверджує замовник перед початком тестування.

2.6.3. Дослідна експлуатація та формування переліку доробок:

Запуск системи в дослідну експлуатацію здійснюється на основі технічного завдання та інструкції користувача.

Після проведення дослідної експлуатації формується перелік необхідних доробок (модифікацій або виправлень), які потрібно виконати для досягнення повної відповідності системи поставленим вимогам, із зазначенням рекомендованих строків їх виконання.

2.6.4. Акт здачі-приймання:

Введення системи в промислову експлуатацію оформлюється актом здачі-приймання, який підтверджує передачу розробленої системи від розробника

замовнику. Наявність такого акта засвідчує, що замовник приймає систему і вона належним чином впроваджена та забезпечена усіма необхідними ресурсами для роботи.

2.7. Вимоги до складу і змісту робіт із підготовки до введення системи в дію

2.7.1. Укомплектування технічних засобів:

Замовник (ФОП Синчук П.П.) забезпечує оснащення всіма необхідними технічними засобами для функціонування системи. Це включає в себе сервер або надійний хостинг для розміщення бази даних і веб-застосунку, засоби мережевого зв'язку (інтернет-підключення достатньої швидкості), а також комп'ютери або інші пристрої для роботи адміністратора і персоналу з системою.

2.7.2. Навчання користувачів системи:

Замовник організовує навчання співробітників, які будуть користуватися системою. Навчання повинно включати освоєння роботи на персональних комп'ютерах (при необхідності підвищення цифрової грамотності) та детальне ознайомлення з інструкцією користувача і функціями нової інформаційної системи. Після проходження навчання персонал має впевнено виконувати операції в системі згідно зі своїми ролями (адміністрування, обробка замовлень тощо).

2.7.3. Дослідна експлуатація і введення системи в дію:

Замовник проводить дослідну (тестову) експлуатацію системи в реальних умовах підприємства з метою перевірки її працездатності та визначення готовності до повноцінного введення в експлуатацію. На цьому етапі виявляються можливі недоліки у практичному використанні системи, оцінюється зручність інтерфейсу, надійність роботи та відповідність усім бізнес-вимогам. За результатами дослідної експлуатації приймається рішення про введення системи в промислову експлуатацію.

2.8. Вимоги до документації

2.8.1. Склад документації:

Для інтернет – магазину розробляється комплект документації, що включає дане технічне завдання, проектну документацію (технічний проект системи, модель

бази даних, описи модулів програми) та інструкцію користувача. Уся документація повинна відображати актуальний стан системи та містити необхідні відомості для її підтримки і використання.

2.8.2. Відповідність стандартам:

Документація на інтернет – магазин розробляється відповідно до вимог державних стандартів серії 19 «Єдина система програмної документації» та серії 24 «Єдина система стандартів автоматизованих систем управління». Зміст і оформлення документів повинні відповідати встановленим стандартам структурам та правилам, зокрема дотримуватися ДСТУ 3008–2015 (щодо структури і оформлення звітів у сфері науки і техніки).

2.9. Джерела розробки

2.9.1. Використання документів:

При розробленні технічного завдання на інтернет – магазин використано такі нормативні та методичні документи:

- ДСТУ 3008:2015 «Документація. Звіти у сфері науки і техніки. Структура та правила оформлення».
- ДСТУ 3973–2000 «Система розроблення та поставлення продукції на виробництво. Загальні положення».
- ДСТУ Б В.2.5–82:2016 «Електробезпека в будівлях і спорудах. Вимоги до захисних заходів від ураження електричним струмом».

РОЗДІЛ 3 ПРОЄКТУВАННЯ, СТВОРЕННЯ ТА АПРОБАЦІЯ ІНТЕРНЕТ – МАГАЗИНУ ДЛЯ ПРОДАЖУ ВЗУТТЯ

3.1 Аналіз мов програмування для розробки систем

Для реалізації інтернет-магазину для продажу взуття ФОП Синчук П.П. також були розглянуті мови програмування, які найчастіше використовуються для створення інтернет - магазинів. Нижче наведено короткий аналіз таких мов програмування: С#, PHP, Python та JavaScript.

3.1.1 С#

С# разом із **ASP.NET Core MVC** – популярний вибір для розробки продуктивних та безпечних веб-магазинів [23].

- Використовує патерн **MVC**, що забезпечує розділення логіки, представлення та управління даними.
- Висока продуктивність та масштабованість завдяки оптимізації **ASP.NET Core**.
- Вбудовані механізми безпеки (захист від SQL-ін'єкцій, XSS, CSRF).
- Легка інтеграція з базами даних (**MSSQL**, **PostgreSQL**, **MySQL**) та зовнішніми API.
- Підходить для середніх і великих проєктів, потребує хостингу на **Windows** або **Linux**-серверах.

3.1.2 PHP

PHP – одна з найпопулярніших мов для створення веб-додатків, часто використовується з **Laravel** або **Symfony** [26].

- **Laravel** використовує патерн **MVC**, що спрощує структуру коду та підтримку проєкту.
- Велика спільнота та широкий вибір бібліотек для роботи з eCommerce.
- Добре підходить для малого та середнього бізнесу завдяки доступному хостингу.
- Потрібне додаткове налаштування для підвищення безпеки.

3.1.3 Python

Python разом із **Django** або **Flask** використовується для створення веб-додатків із потужною логікою [9].

- Django реалізує патерн **MVT**, що схожий на **MVC**.
- Добре підходить для інтеграції з машинним навчанням та аналітикою.
- Високий рівень безпеки завдяки вбудованим механізмам Django.
- Менша продуктивність порівняно з **C#**, але може бути достатньою для середніх магазинів.

3.1.4 JavaScript

Node.js разом із **Express** або **Next.js** дозволяє створювати швидкі та інтерактивні eCommerce-додатки [25].

- Використовує патерн **REST API** або **SSR** в **Next.js**.
- Одночасна обробка великої кількості запитів завдяки **асинхронній моделі**.
- Легко інтегрується з фронтенд-фреймворками (React, Vue.js, Angular).
- Підходить для великих проєктів із високим навантаженням.
- Вимагає досвіду в асинхронному програмуванні.

3.2 Інформаційне забезпечення системи

База даних розроблялась за допомогою міграцій з використанням бібліотеки `Microsoft.EntityFrameworkCore.Tools` в Visual Studio, на Рисунок 3.1 показано модель бази даних в MS SQL Server [12].

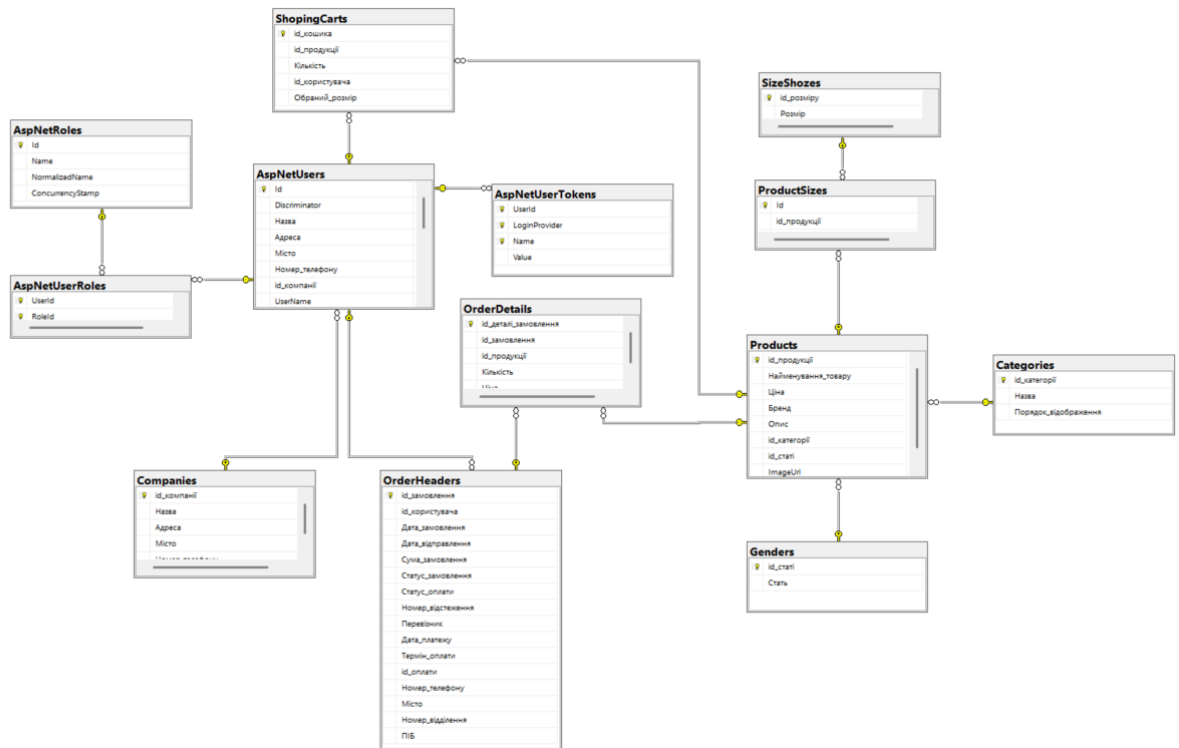


Рисунок 3.1 – Модель бази даних

Наступний етап, підключення бази даних до проекту, за допомогою рядка підключення [19].

Рядок підключення до БД:

```
"ConnectionStrings": {
  "DefaultConnection":
```

```
"Server=localhost;Database=VzutikDB;Trusted_Connection=True;TrustServerCertificate=True"
```

```
}
```

Підключення в Program.cs:

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

Та реалізація контексту бази даних в C#.

Фрагмент коду:

```
public class ApplicationDbContext : IdentityDbContext<IdentityUser>
```

```

{
    public ApplicationDbContext(DbContextOptions<ApplicationDbContext>
options) : base(options)
    {

    }

    public DbSet<Category> Categories { get; set; }
    public DbSet<Product> Products { get; set; }
    public DbSet<ApplicationUser> applicationUsers { get; set; }
    public DbSet<Company> Companies { get; set; }
    public DbSet<ShopingCart> ShopingCarts { get; set; }
    public DbSet<OrderHeader> OrderHeaders { get; set; }
    public DbSet<OrderDetail> OrderDetails { get; set; }
    public DbSet<SizeShoes> SizeShozes { get; set; }
    public DbSet<Gender> Genders { get; set; }
    public DbSet<ProductSize> ProductSizes { get; set; }
}

```

Для того щоб створити таблиці у базі даних, використовується система міграцій Entity Framework за допомогою команди [25]: *Add-Migration InitialCreate* буде створено міграцію. Після виконання команди *Update-Database* система створює відповідні таблиці у базі даних згідно з визначеннями в *ApplicationDbContext*. Це дозволяє легко синхронізувати модель проєкту з реальними структурами бази даних.

За стовпці в таблицях бази даних відповідають моделі, які визначаються у вигляді C# класів. Кожна властивість класу відповідає окремому стовпцю таблиці, а типи даних у моделях автоматично перетворюються у відповідні SQL-типи.

Фрагмент коду:

```

public class Category
{
    [Key]

```

```

public int id_категорії { get; set; }
[Required]
[MaxLength(30)]
public string Назва { get; set; }
public int Код_категорії { get; set; }
}
}

```

3.3 Реалізація функцій системи

Проект був побудований за принципом багат шарової архітектури та поділений на окремі підпроекти для забезпечення зручності підтримки, повторного використання коду та чіткого розмежування відповідальності, на рисунку 3.2 зображено структуру проекту.

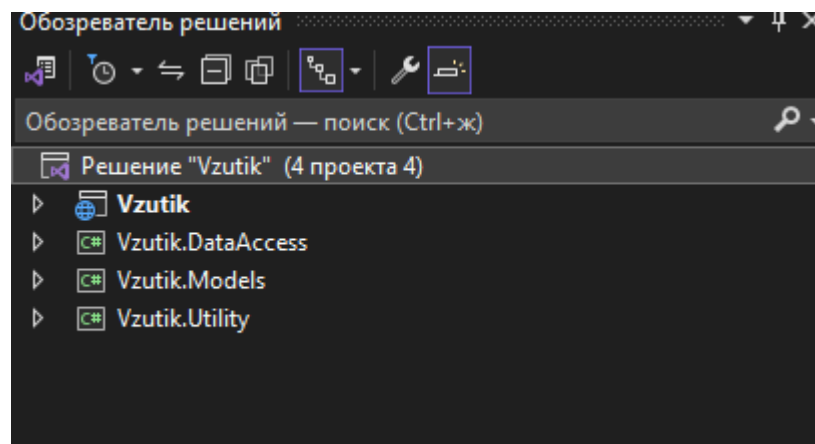


Рисунок 3.2 – Структура проекту

- Vzutik — основний веб-проект, який реалізує інтерфейс користувача та взаємодію з користувачами через ASP.NET Core MVC.
- Vzutik.Models — містить усі моделі даних, які відповідають таблицям у базі даних.
- Vzutik.DataAccess — відповідає за роботу з базою даних, включає контекст ApplicationDbContext і репозиторії.
- Vzutik.Utility — містить допоміжні класи, константи та конфігураційні елементи.

Наступним етапом стало створення системи автентифікації користувачів за допомогою ASP.NET Core Identity [27]. Для цього було використано стандартний шаблон аутентифікації у Visual Studio з інтеграцією Identity, який автоматично згенерував функціонал для реєстрації, входу, виходу з облікового запису, а також обробки помилок, , та було відредаговано для потреб інтернет – магазину, форма реєстрації представлена на рисунку 3.3.

Рисунок 3.3 – Форма реєстрації

Фрагмент коду відображення форми реєстрації:

```
<form id="registerForm" class="row" asp-route-returnUrl="@Model.ReturnUrl"
method="post">
```

```
<h4 class=" border-bottom pb-3 mb-4 text-secondary text-center">Зареєструвати
новий акаунт</h4>
```

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

```
<div class="form-floating mb-3 col-md-12">
```

```
<input asp-for="Input.Email" class="form-control" autocomplete="username"
aria-required="true" placeholder="name@example.com" />
```

```

<label asp-for="Input.Email" class="ms-2 text-muted">Електронна
адреса</label>
<span asp-validation-for="Input.Email" class="text-danger"></span>
</div>
<div class="form-floating mb-3 col-md-6">
<input asp-for="Input.Password" class="form-control" autocomplete="new-
password" aria-required="true" placeholder="password" />
<label asp-for="Input.Password" class="ms-2 text-muted">Пароль</label>
<span asp-validation-for="Input.Password" class="text-danger"></span>
</div>
<div class="form-floating mb-3 col-md-6">
<input asp-for="Input.ConfirmPassword" class="form-control"
autocomplete="new-password" aria-required="true" placeholder="password" />
<label asp-for="Input.ConfirmPassword" class="ms-2 text-muted">Підтвердити
пароль</label>
<span asp-validation-for="Input.ConfirmPassword" class="text-
danger"></span>
</div>
<div class="form-floating mb-3 col-md-6">
<input asp-for="Input.Місто" class="form-control" placeholder="Київ" />
<label asp-for="Input.Місто" class="ms-2 text-muted">Місто</label>
<span asp-validation-for="Input.Місто" class="text-danger"></span>
</div>
<div class="form-floating mb-3 col-md-6">
<input asp-for="Input.Назва" class="form-control" placeholder="Іванов Іван
Іванович" />
<label asp-for="Input.Назва" class="ms-2 text-muted">Прізвище Імя по
Батькові</label>
<span asp-validation-for="Input.Назва" class="text-danger"></span>
</div>

```

```

<div class="form-floating mb-3 col-md-6">
  <input asp-for="Input.Адреса" class="form-control" placeholder="Іванова 10"
/>
  <label asp-for="Input.Адреса" class="ms-2 text-muted">Адреса</label>
  <span asp-validation-for="Input.Адреса" class="text-danger"></span>
</div>

```

```

<div class="mb-3 col-md-6">
  <label asp-for="Input.Номер_телефону" class="form-label">Номер
телефону</label>
  <div class="input-group">
    <span class="input-group-text">+38</span>
    <input asp-for="Input.Номер_телефону" class="form-control" type="tel"
inputmode="numeric" placeholder="066XXXXXXX" pattern="\d{10}"
maxlength="10" />
  </div>
  <span asp-validation-for="Input.Номер_телефону" class="text-
danger"></span>
</div>

```

Обов'язкові поля на формі це:

- Електронна адреса.
- Пароль.
- Прізвище, Ім'я, по Батькові.

Необов'язкові поля:

- Місто
- Адреса
- Номер телефону

Якщо при реєстрації ввести всі поля на формі, то при замовленні продукції дані автоматично завантажуються на форму замовлення.

Форма для входу в інтернет – магазин зображено на рисунку 3.4.

Рисунок 3.4 – Форма «Вхід»

На рисунку 3.5 зображено головну сторінку інтернет – магазину. На центральній сторінці знаходиться каталог товарів, в якому реалізовано функціонал фільтрації за категорією, розміром та статтю. Також було розроблено пошук за назвою товару.

Кожна продукція відображається у вигляді картки з фотографією, назвою, брендом, ціною та кнопкою "Детальніше", яка веде на сторінку з описом товару.

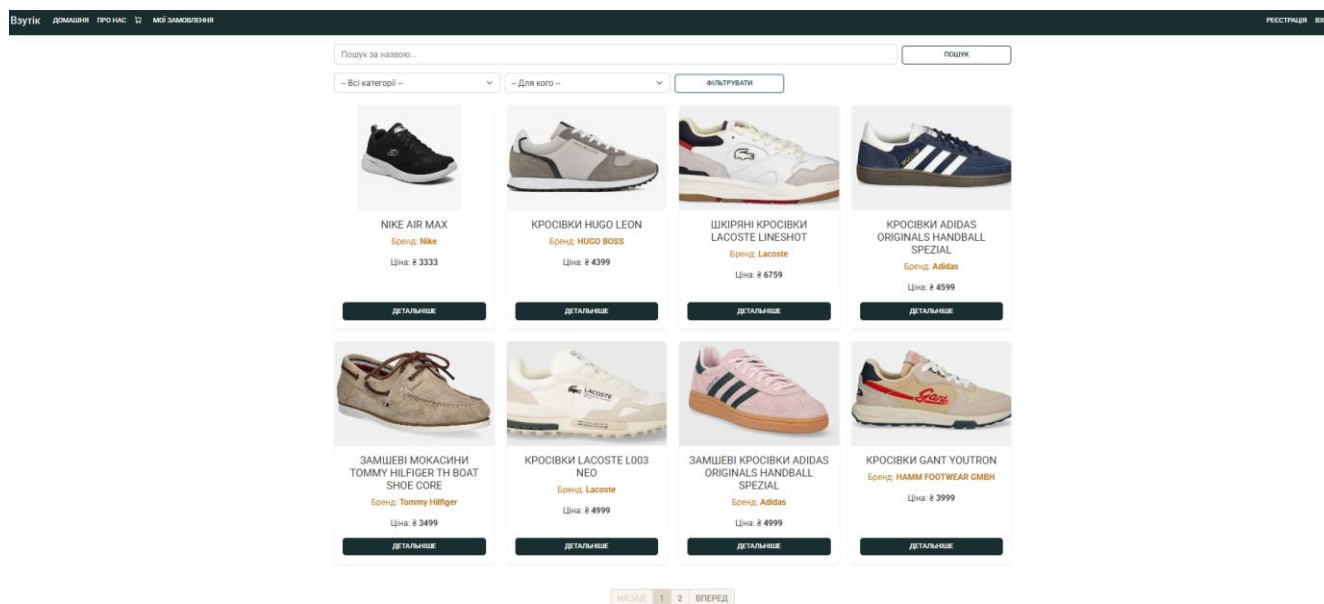


Рисунок 3.5 – Головна сторінка інтернет – магазину

Фрагмент коду головної сторінки для відображення продукції:

```

<div class="row g-4">
  @foreach (var product in Model.Products)
  {
    <div class="col-lg-3 col-md-4 col-sm-6">
      <div class="card h-100 border-0 shadow-sm">
        

        <div class="card-body d-flex flex-column">
          <h5 class="card-title text-center text-uppercase text-dark">
            @product.Найменування_товару
          </h5>
          <p class="card-text text-center text-warning">
            Бренд: <strong>@product.Бренд</strong>
          </p>
          <p class="card-text text-center text-dark">
            Ціна: ₪ <strong>@product.Ціна</strong>
          </p>
          <a asp-action="Details" asp-route-
productId="@product.id_продукції" class="mt-auto btn btn-primary btn-gradient
border-0">
            Детальніше
          </a>
        </div>
      </div>
    </div>
  }
</div>

```

На рисунку 3.6 зображено основну інформацію про компанію.

Про компанію Взутік

Ласкаво просимо до "Взутік" – вашого надійного партнера у світі якісного взуття!

Компанія "Взутік" займається продажем якісного та стильного взуття для всієї родини. Ми пропонуємо широкий асортимент моделей – від класичних та ділових варіантів до спортивного та повсякденного взуття.

Наша місія

Ми прагнемо зробити якісне та зручне взуття доступним для кожного. У нашому магазині ви знайдете товари від перевірених виробників, які поєднують у собі сучасний дизайн, надійність та комфорт.

Чому обирають "Взутік"?

- Широкий вибір взуття для жінок, чоловіків та дітей;
- Тільки якісні матеріали та перевірені бренди;
- Зручний онлайн-магазин з швидкою доставкою;
- Регулярні знижки та акції для наших клієнтів;
- Індивідуальний підхід до кожного покупця.

Контакти

Якщо у вас є запитання або пропозиції, зв'яжіться з нами:

Телефон: +380 (95) 222-11-11

Email: info@vzutik.com

Адреса: м. Київ, вул. Прикладна, 12

Дякуємо, що обираєте "Взутік"! Ми працюємо для вашого комфорту.

Рисунок 3.6 – інформаційна сторінка «Про нас»

Усередині проекту Vzutik створено три основні області:

- Admin який призначена для адміністратора. Містить функціонал управління категоріями, товарами, замовленнями, користувачами та звітами.

Для того щоб співробітник міг користуватись окремим функціоналом таким як: створення продукції та її редагування, обробка замовлень тощо, в контроллерах потрібно прописувати наступний код:

Фрагмент коду:

```
[Area("Admin")]
[Authorize(Roles = SD.Role_Admin)]
```

Для того щоб звичайному користувачу не відображались вкладки для управління контентом на головній сторінці був прописаний наступний код:

```
@if (User.IsInRole(SD.Role_Admin))
{
    <li class="nav-item">
        <a class="nav-link text-light" asp-area="Admin" asp-controller="Order"
asp-action="Index">Управління замовленнями</a>
    </li>
    <li class="nav-item">
        <a class="nav-link text-light" asp-area="Admin" asp-controller="User"
asp-action="Index">Управління користувачами</a>
    </li>
}
```

```

</li>
<li class="nav-item dropdown">
  <a class="nav-link dropdown-toggle text-light" href="#" role="button"
data-bs-toggle="dropdown" aria-expanded="false">
    Управління контентом
  </a>
}

```

- Customer логіка взаємодії з кінцевим користувачем. Тут реалізовано перегляд товарів, фільтрацію, кошик, оформлення замовлення та перегляд його замовлень.

Для того щоб взаємодіяти з інтерфейсом було створено контролери, які відповідають за обробку натискання або запиту до БД. Для прикладу наведено фрагмент коду ProductController:

```

public IActionResult Index(string? searchTerm, int? categoryId)
{
    var products = _unitOfWork.Product.GetAll(includeProperties:
"Category").AsQueryable();

    if (!string.IsNullOrEmpty(searchTerm))
    {
        products = products.Where(p =>
p.Найменування_товару.Contains(searchTerm.Trim(),
StringComparison.OrdinalIgnoreCase));
        categoryId = null;
    }
    else if (categoryId.HasValue)
    {
        products = products.Where(p => p.id_категорії == categoryId);
    }
}

```

```

ViewBag.SearchTerm = searchTerm;
ViewBag.CategoryId = categoryId;
ViewBag.Categories = _unitOfWork.Category.GetAll()
    .Select(c => new SelectListItem
    {
        Text = c.Назва,
        Value = c.id_категорії.ToString()
    });

return View(products.ToList());
}

```

На рисунку 3.7 зображено структуру папки Repository у проєкті Vzutik.DataAccess, де реалізовано шаблони репозиторію.

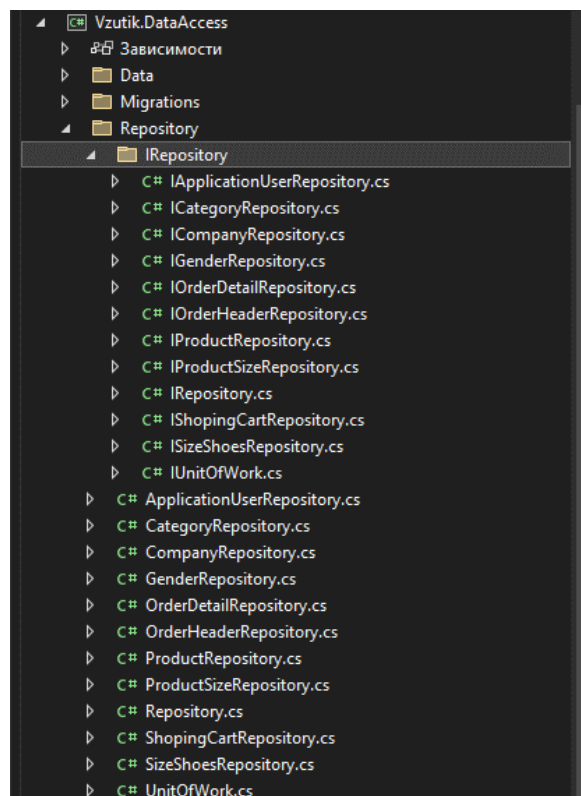


Рисунок 3.7 – Структура репозиторію

Кожна сутність, така як Product, OrderHeader, Category, має свій інтерфейс

наприклад, IProductRepository та реалізацію ProductRepository.

Це забезпечує зручну й централізовану роботу з базою даних, дозволяючи відокремити логіку доступу до даних від логіки.

Файл UnitOfWork.cs об'єднує всі репозиторії в єдиний об'єкт, що дозволяє зберігати зміни до бази даних.

Фрагмент коду ProductRepository:

```
public void Update(Product obj)
{
    var objFromDb = _db.Products.FirstOrDefault(u => u.id_продукції ==
obj.id_продукції);
    if (objFromDb != null)
    {
        objFromDb.Найменування_товару = obj.Найменування_товару;
        objFromDb.Ціна = obj.Ціна;
        objFromDb.Бренд = obj.Бренд;
        objFromDb.Опис = obj.Опис;
        objFromDb.id_категорії = obj.id_категорії;
        if(obj.ImageUrl != null)
        {
            objFromDb.ImageUrl = obj.ImageUrl;
        }
    }
}
```

3.3 Інструкція користувача

Після входу в інтернет – магазин на головні сторінці, зображено доступну продукцію, зображено на рисунку 3.8.

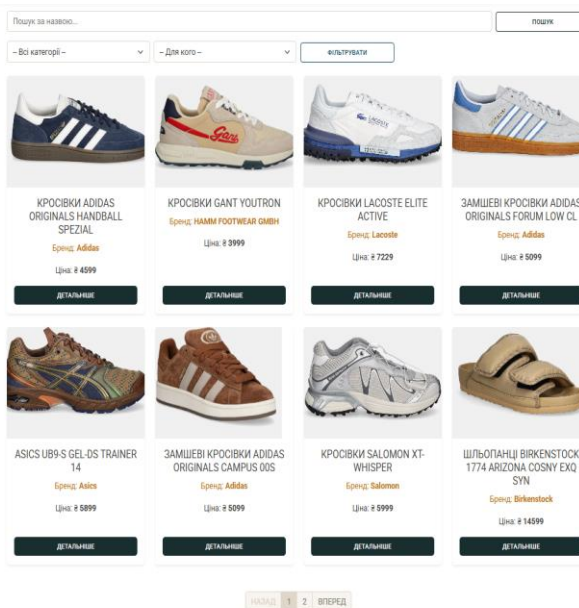


Рисунок 3.8 – Головна сторінка інтернет - магазину

Для того щоб зробити замовлення, потрібно бути зареєстрованим користувачем.

Для цього потрібно натиснути на кнопку «Реєстрація», та заповнити необхідні поля на формі, на рисунку 3.9 показано приклад заповнення форми.

3.9 – Форма «Реєстрація»

Для того щоб зробити замовлення, потрібно вибрати необхідну продукцію, натиснути «Детальніше», обрати розмір та кількість та додати до замовлення.

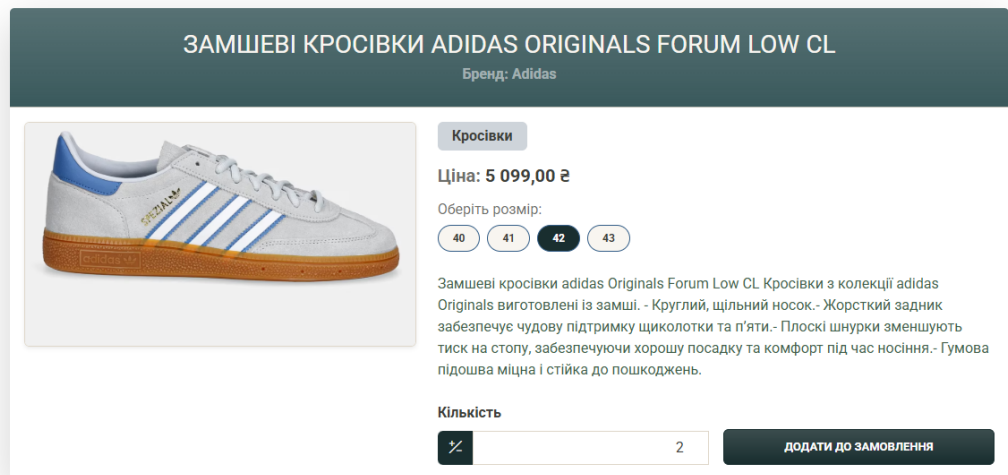


Рисунок 3.10 – Сторінка продукції

Після чого перейти до кошика та натиснути кнопку замовити, зображено на рисунку 3.11.

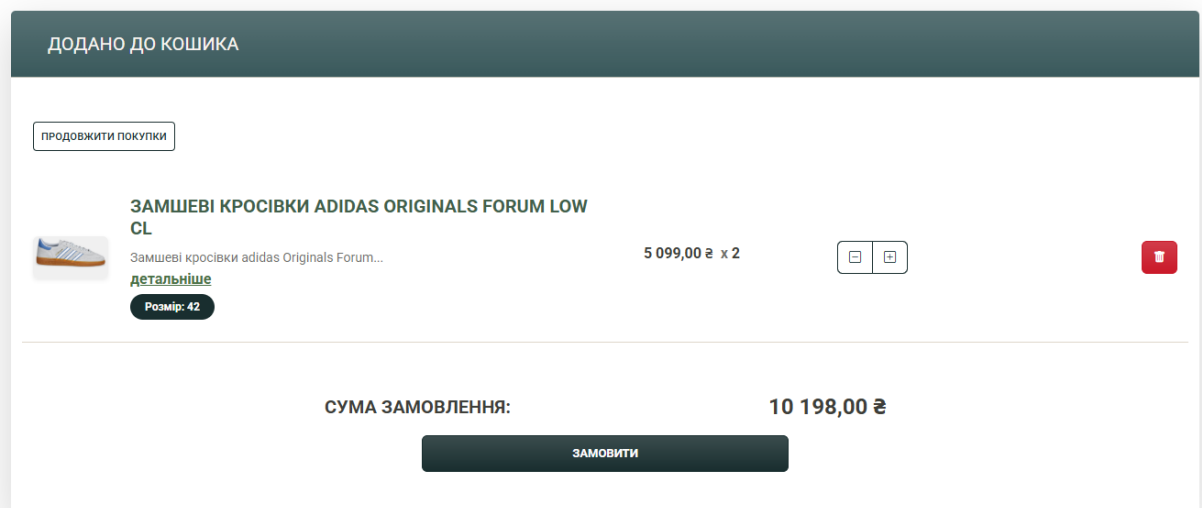


Рисунок 3.11 – Форма «Додано до кошика»

Після того як натиснути на замовити ви перейдете до деталей замовлення зображено на рисунку 3.12, де потрібно заповнити необхідні поля.

Деталі замовлення ПОВЕРНУТИСЬ ДО КОШИКА

Деталі доставки:

Прізвище, імя, по-батькові: Ivan Ivanov

Номер телефону: 0662223232

Місто: Київ

Спосіб доставки: Нова Пошта

Номер відділення: 33

Сума замовлення:

Замшеві кросівки adidas Originals Forum Low CL	10 198,00 ₴
Кількість: 2	
Ціна ГРН	10 198,00 ₴

Орієнтовна дата прибуття: 16.05.2025 - 20.05.2025 ЗРОБИТИ ЗАМОВЛЕННЯ

Рисунок 3.12 – Форма «Деталі замовлення»

Натиснувши кнопку «Зробити замовлення», буде вказано номер вашого замовлення зображено на рисунку 3.13.

Замовлення успішно оформлене!

Ваш номер замовлення:

13

ПРОДОВЖИТИ ПОКУПКИ

Рисунок 3.13 – Створене замовлення

Також можна переглянути замовлення натиснувши кнопку «Мої замовлення» зображено на рисунку 3.14.

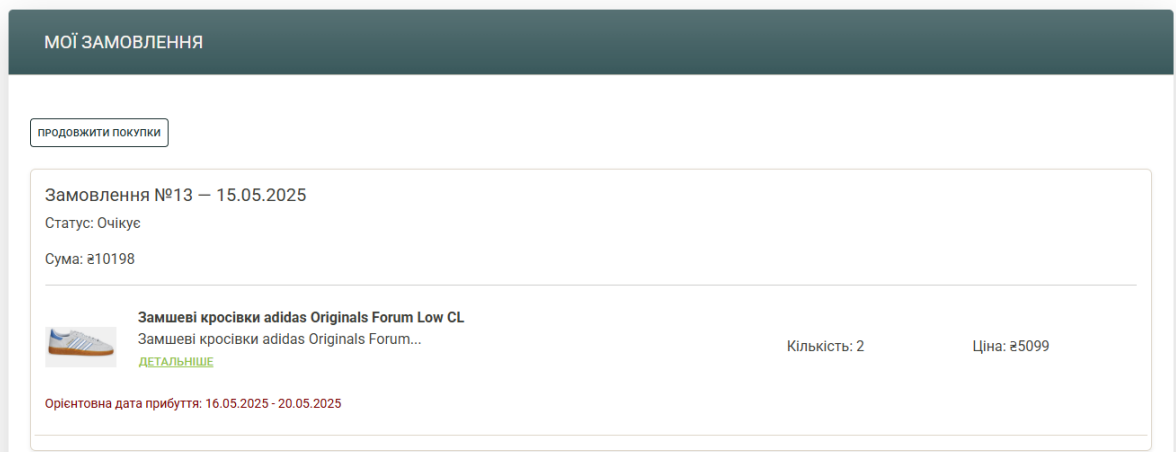


Рисунок 3.14 – Форма «Мої замовлення»

3.4 Управління контентом інтернет – магазину

Коли менеджер дає права доступу до управління контентом то в навігаційному меню появляться нові вкладки:

- Управління замовленнями
- Категорії
- Продукція
- Компанії
- Розмір взуття

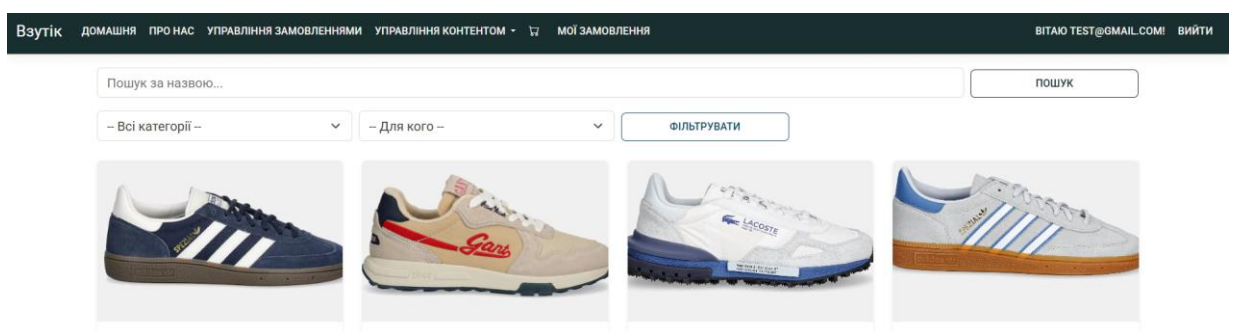


Рисунок 3.15 – Навігаційне меню для управління контентом

Натиснувши на кнопку «Управління замовленнями» ви попадете на форму де показано всі замовлення, також на формі реалізовано пошук замовлення по його номеру та по ПІБ клієнта, форму зображено на рисунку 3.16.

Список замовлень

Пошук за номером замовлення Пошук за ПІБ **ПОШУК** **ОЧИСТИТИ**

Номер	ПІБ	Місто	Телефон	Дата	Статус	Дії
10	Ivan Ivanov	Київ	0662223232	13.05.2025	Скасовано	ПЕРЕГЛЯНУТИ
11	Ivan Ivanov	Київ	0662223232	13.05.2025	Доставлено	ПЕРЕГЛЯНУТИ
12	Ivan Ivanov	Київ	0662223232	15.05.2025	Очікує	ПЕРЕГЛЯНУТИ
13	Ivan Ivanov	Київ	0662223232	15.05.2025	Очікує	ПЕРЕГЛЯНУТИ
14	Гриб Ангеліна	Київ	0663689802	15.05.2025	Опрацьовується	ПЕРЕГЛЯНУТИ

Рисунок 3.16 – Форма «Список замовлень»

Натиснувши кнопку «Переглянути», відкриється форма «Деталі замовлення», на якій можна змінити статус замовлення, та переглянути деталі замовлення, форму зображено на рисунку 3.17.

Взутік ДОМАШНЯ ПРО НАС УПРАВЛІННЯ ЗАМОВЛЕННЯМИ УПРАВЛІННЯ КОНТЕНТОМ - МОЇ ЗАМОВЛЕННЯ ВІТАЮ TEST@GMAIL.COM! ВИЙТИ

Деталі замовлення №14 Статус: Опрацьовується **ЗМІНИТИ**

ПІБ: Гриб Ангеліна
 Місто: Київ
 Телефон: 0663689802
 Номер відділення: 33

Товар	Кількість	Ціна	Сума
Asics UB9-S GEL-DS TRAINER 14	1	€5899,00	€5899,00
Кросівки Gant Youtron	1	€3999,00	€3999,00
Замшеві кросівки adidas Originals Campus 00S	2	€5099,00	€10198,00
Загальна сума:			€20096,00

[← НАЗАД ДО СПИСКУ ЗАМОВЛЕНЬ](#)

Рисунок 3.17 – Форма «Деталі замовлення»

Перейшовши на вкладку продукція, можна переглянути список доданої продукції також можна редагувати, видалити або створити нову продукцію на рисунку 3.16 зображено форму «Список продукції».

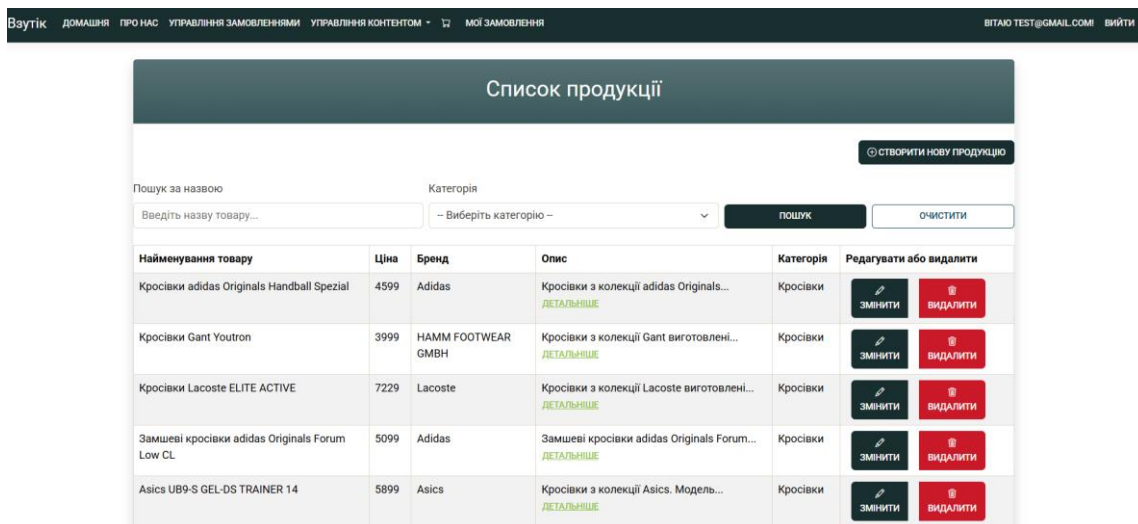


Рисунок 3.18 – Форма «Список продукції»

Натиснувши «Створити нову продукцію», відкриється форма «Створити продукцію», на рисунку 3.19 зображено дану форму.

Натиснувши кнопку «Видалити», можна видалити вибрану продукцію.

Створити продукцію

Назва товару

Ціна

Бренд

Доступні розміри

38 39 40 41 42 43 44

Опис

File Edit View Insert Format

← → Paragraph System Font 12pt **B** *I* U ↺ ...

Build with tinyMCE

id_категорії

Для кого

Фото взуття

Вибрати файлФайл не вибрано

СТВОРИТИ

ПОВЕРНУТИСЬ НАЗАД

Рисунок 3.19 – Форма «Створити продукцію»

Натиснувши кнопку «змінити», відкриється форма «Змінити продукцію», на рисунку 3.20 зображено як виглядає ця форма.

Змінити продукцію

Назва товару
Кросівки adidas Originals Handball Spezial

Ціна
4599

Бренд
Adidas

Доступні розміри
 38 39 40 41 42 43 44

Опис

File Edit View Insert Format
 Paragraph System Font 12pt B I U S ...

Кросівки з колекції adidas Originals виготовлені з поєднання замші, натуральної шкіри та текстилю. Легка та зручна модель ідеально підходить для щоденного носіння. - Круглий, м'який носок. - Жорстка п'ятка стабілізує каблук і утримує ногу у взутті, щоб вона не вислизала під час руху. - Класична шнурівка забезпечує індивідуальне регулювання стопи. - У комплекті додаткова пара шнурків. - Текстильна устілка комфортна для стопи та полегшує утримання взуття в чистоті. - Гумова підшова міцна і стійка до пошкоджень.

Build with [tinyMCE](#)

id_категорії
Кросівки

Для кого
Унісекс

Фото взуття
Вибрати файл | Файл не вибрано

ЗМІНИТИ **ПОВЕРНУТИСЬ НАЗАД**

Поточне зображення

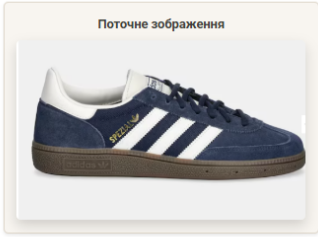


Рисунок 3.20 – Форма «Змінити продукцію»

Для форм категорії, компанії, розміри управління буде відбуватись в такому самому вигляді, як продемонстровано на формі «Продукція».

Якщо менеджер увійшов в систему, в нього відображається додаткова вкладка, а саме «Управління користувачами», в цій вкладці він може надавати права доступу для співробітників, для цього співробітнику потрібно зареєструватись спочатку, як користувач, після чого менеджер може дати йому права доступу, форму зображено на рисунку 3.21.

Взутік ДОМАШНЯ ПРО НАС УПРАВЛІННЯ ЗАМОВЛЕННЯМИ УПРАВЛІННЯ КОРИСТУВАЧАМИ УПРАВЛІННЯ КОНТЕНТОМ МОЇ ЗАМОВЛЕННЯ

Пошук за назвою...

-- Всі категорії -- -- Для кого -- **ФІЛЬТРУВА**

Рисунок 3.21 – Вкладка «Управління користувачами»

О НАС УПРАВЛІННЯ ЗАМОВЛЕННЯМИ УПРАВЛІННЯ КОРИСТУВАЧАМИ УПРАВЛІННЯ КОНТЕНТОМ - МОЇ ЗАМОВЛЕННЯ

Список користувачів

Введіть Email для пошуку

Email	Ім'я користувача (UserName)	Дія
TestCustomer@gmail.com	TestCustomer@gmail.com	<input type="button" value="ЗМІНИТИ РОЛЬ"/>
CustomerUser@gmail.com	CustomerUser@gmail.com	<input type="button" value="ЗМІНИТИ РОЛЬ"/>
test@gmail.com	test@gmail.com	<input type="button" value="ЗМІНИТИ РОЛЬ"/>
Angelina.Company@gmail.com	Angelina.Company@gmail.com	<input type="button" value="ЗМІНИТИ РОЛЬ"/>

Рисунок 3.22 – Форма «Список користувачів»

Натиснувши кнопку «Змінити роль», менеджер може змінити права доступу, форму зображено на рисунку 3.23.

УПРАВЛІННЯ ЗАМОВЛЕННЯМИ УПРАВЛІННЯ КОРИСТУВАЧАМИ УПРАВЛІННЯ КОНТЕНТОМ - МОЇ ЗАМОВЛЕННЯ

Зміна ролі для CustomerUser@gmail.com

Поточна роль

Нова роль

Рисунок 3.23 – Форма «Змінити роль»

3.5 Тестування інтернет – магазину

Після того як інтернет – магазин було розроблено потрібно провести тестування з метою перевірки коректності відображення всіх елементів та функцій ІС. Перевірка здійснювалась на основних формах інтернет – магазину, а саме: реєстрації, авторизації, кошику, фільтрації товарів, оформленню замовлень та доступу до адміністративної панелі. Також перевірялась безпека вводу даних, зокрема захист від SQL-ін'єкцій. В таблиці 3.1 наведено узагальнений план тестування із зазначенням основних тестових сценаріїв та очікуваних результатів.

Таблиця 3.1 План тестування

№	Назва тесту	Очікуваний результат
1	Реєстрація нового користувача	Користувача додано до бази, автоматичний вхід
2	Авторизація з правильними даними	Користувач потрапляє в особистий кабінет
3	Авторизація з неправильними даними	Повідомлення про помилку входу
4	Додавання товару в кошик	Товар з'являється у списку замовлень
5	Видалення товару з кошика	Товар зникає зі списку
6	Пошук за назвою	Виводяться релевантні результати
7	Фільтрація за категорією та статтю	Виводяться товари згідно з фільтрами
8	Оформлення замовлення	Створюється запис у таблицях OrderHeader/Detail
9	Вхід в адмін-панель неавторизованого	Попереджуваче повідомлення про права доступу
10	Перевірка відображення елементів з різним розширенням екрану	Елементи відображаються коректно

Тестування форми «реєстрація» якщо елементи пусті та натиснути кнопку «Зареєструвати», результатом буде (Рисунок 3.24).

The screenshot shows a registration form titled "Реєстрація" with the subtitle "Зареєструвати новий акаунт". The form contains several input fields: "Електронна адреса", "Пароль", "Підтвердити пароль", "Місто", "Прізвище Імя по Батькові", "Адреса", and "Номер телефону". The "Номер телефону" field has a dropdown for the country code (+38) and a text input for the number (066XXXXXXX). Below the "Електронна адреса" field, there is a red error message: "Поле 'Електронна пошта' обов'язкове для заповнення." Below the "Пароль" field, there is another red error message: "Поле «Пароль» обов'язкове для заповнення." At the bottom of the form, there is a dark button labeled "ЗАРЕЄСТРУВАТИСЯ".

Рисунок 3.24 – Результат натискання «Зареєструвати» при незаповнених елементах

Для перевірки реєстрації нового користувача буде введено тестову пошту та пароль. Для прикладу електронна адреса буде testuser@example.com а пароль «Qwerty1234.», результат успішна реєстрація.

Рисунок 3.25 – Тестування форми реєстрації

Також якщо спробувати зареєструватись повторно на ту саму електронну адресу, буде помилка (Рис. 3.26).

Рисунок 3.26 – Тестування повторної реєстрації

1	4681c403-cf79-4d38-96d4-2c3c20e22f27	ApplicationUser	Customer Test	Test Adres	Test City	1111111111	TestCustomer@gmail.com	TestCustomer@gmail.com	AQAAAAIAAYagAAAAEG52B9NOSzppzCbW78AiC+PyJzTwxjR0cz...
2	66e4918f-c86b-408b-9732-4839a294c055	ApplicationUser	Іван іванов	Проспект Науки 36	Київ	1231231231	CustomerUser@gmail.com	CustomerUser@gmail.com	AQAAAAIAAYagAAAAEG3WX1lo86bsubu1tb0nW5i2XZaBM3Ya6n...
3	6a0cd73c-a2b-46c4-87b8-6009d431c722	ApplicationUser	NULL	NULL	NULL	NULL	testuser@example.com	testuser@example.com	AQAAAAIAAYagAAAAEKY08wtTGVETydcxs57u0iUKQHLfKrBmH...
4	6ab1d917-4518-42bd-a50f-417e5b3272e5	ApplicationUser	Ivan Ivanov	Проспект Науки	Київ	0662223232	test@gmail.com	test@gmail.com	AQAAAAIAAYagAAAAEKfwwGbkKkuFFyqq+TMSQmyTcOlvooNvO...
5	e6474d2-7551-4cde-8559-23aaf30f667	ApplicationUser	Гриб Ангеліна	Проспект Науки 36	Київ	0663689802	Angelina.Company@gmail.com	Angelina.Company@gmail.com	AQAAAAIAAYagAAAAEN1u5uWUQ0XaQ0086z86S1S1SFwvwr...

Рисунок 3.27 – Відображення тестового користувача в БД

Після чого вийдемо з акаунту, та спробуємо увійти з недійсним паролем, результат зображено на рисунку 3.28.

Рисунок 3.28 – Авторизація з не дійсним паролем

Наступним кроком буде пошук товару, для прикладу візьмемо «Кросівки adidas». Результат зображено на рисунку 3.29.

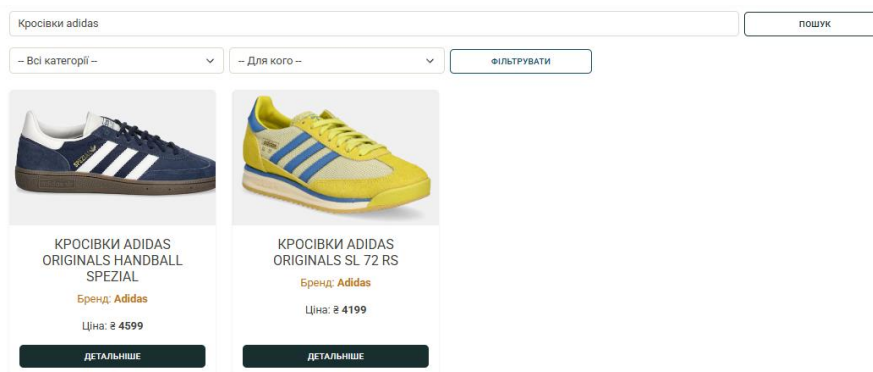


Рисунок 3.29 – Результат пошуку

Наступним кроком буде перевірка фільтрів для прикладу візьмемо категорію «Кросівки» та «Для чоловіків».

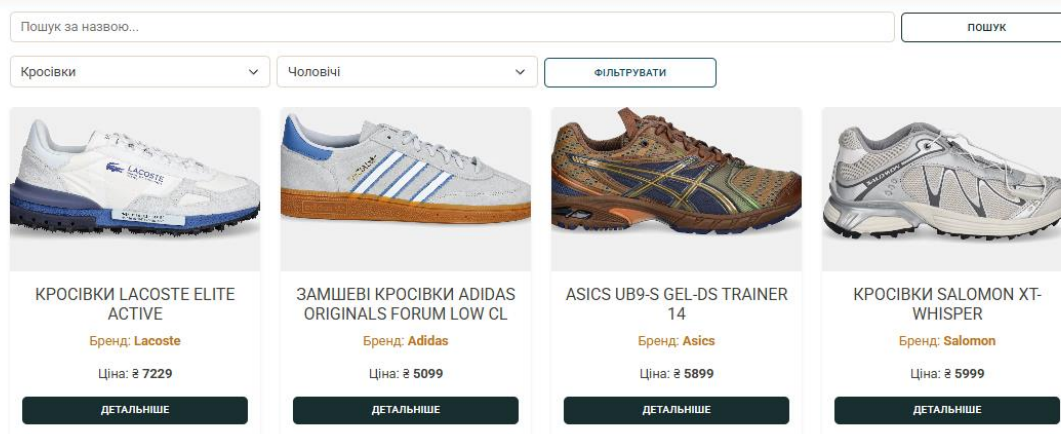


Рисунок 3.30 – Результат використання фільтру

Натиснувши «Детальніше» на формі спробуємо ввести кількість -1, та вибрати продукцію не вибравши розмір. Результатом буде, якщо не вибрати розмір, продукції то кнопка замовити неактивна, якщо ввести -1 в кількість то виводиться повідомлення яке зображено на рисунку 3.32



Рисунок 3.31 – Результат тестування форми «Деталі продукції»



Рисунок 3.32 – Результат тестування форми «Деталі продукції»

Коли кошик порожній на кнопці виводиться повідомлення «Кошик порожній» і вона неактивна доти доки не буде додано хоч одну позицію:

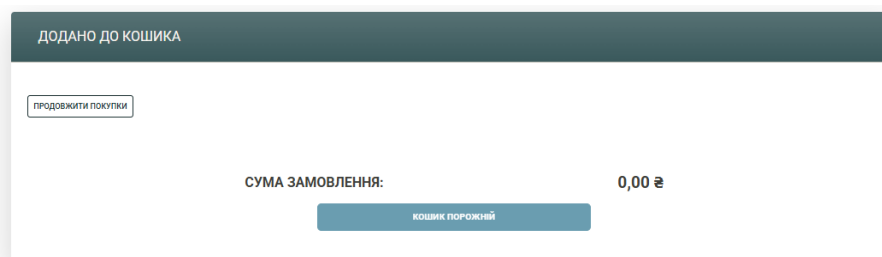


Рисунок 3.33 – Перевірка форма «Додано до кошика»

Додавши товар в кошик, товар редагується без проблем, Кнопки «продовжити покупки», додати кількість та відняти кількість також працює коректно, також кнопка видалити позицію також працює без помилок, коли кількість вибраної продукції менша 0 то продукція зникає, також всі дані відображаються коректно.

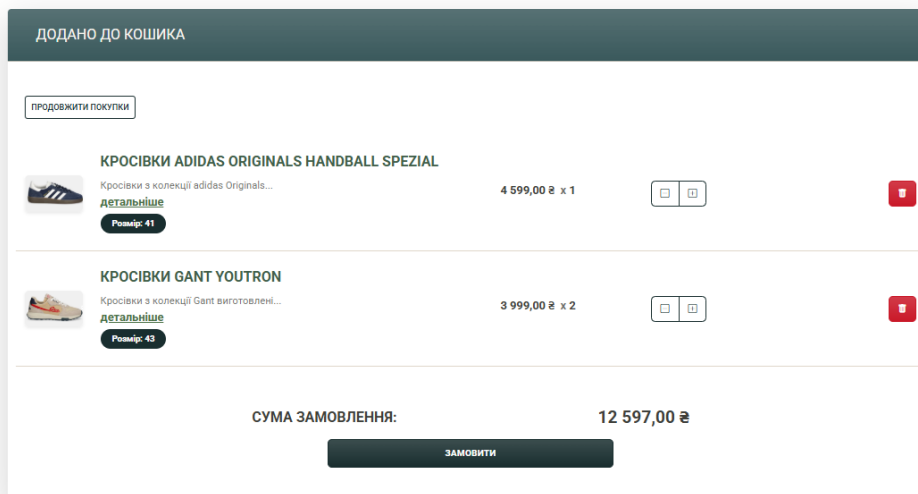


Рисунок 3.34 – Тестування форми «Додано до кошика»

Наступним кроком буде перевірка створення замовлення, яка зображена на рисунку 3.35. Натиснувши кнопку зробити замовлення без заповнених елементів на формі, виводиться повідомлення що поля є обов'язковими, кнопка «Повернутись до кошика», працює як і очікувалось, повертає в форму «Додано до кошика».

Деталі замовлення [ПОВЕРНУТИСЬ ДО КОШИКА](#)

Деталі доставки:

Прізвище, ім'я, по-батькові:
Введіть прізвище ім'я по-батькові

Номер телефону:
Введіть номер телефону

Місто:
Введіть місто

Спосіб доставки: Нова Пошта

Номер відділення:
Введіть номер відділення

Орієнтовна дата прибуття: 17.05.2025 - 21.05.2025

Сума замовлення:

Кросівки adidas Originals Handball Spezial	4 599,00 ₴
Кількість: 1	
Кросівки Gant Youtron	7 998,00 ₴
Кількість: 2	
Ціна ГРН	12 597,00 ₴

ЗРОБИТИ ЗАМОВЛЕННЯ

Рисунок 3.35 – Тестування форми «Деталі замовлення»

Заповнивши всі поля та натиснувши «Зробити замовлення», повинно відобразитись номер замовлення, та те що замовлення успішно створено, результат зображено на рисунку 3.36.

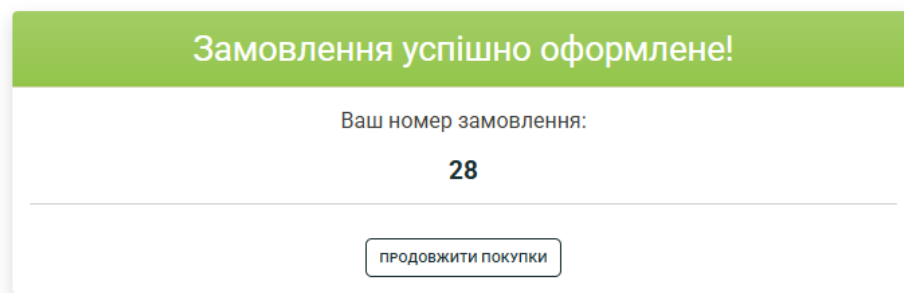


Рисунок 3.36 – Успішно створене замовлення

Натиснувши «Продовжити покупки», відкриється головна форма інтернет – магазину, також можна переглянути замовлення натиснувши кнопку «Мої замовлення», на даній формі показано, які позиції були замовлені, ціна кожної продукції та сума замовлення, зображено на рисунку 3.37.

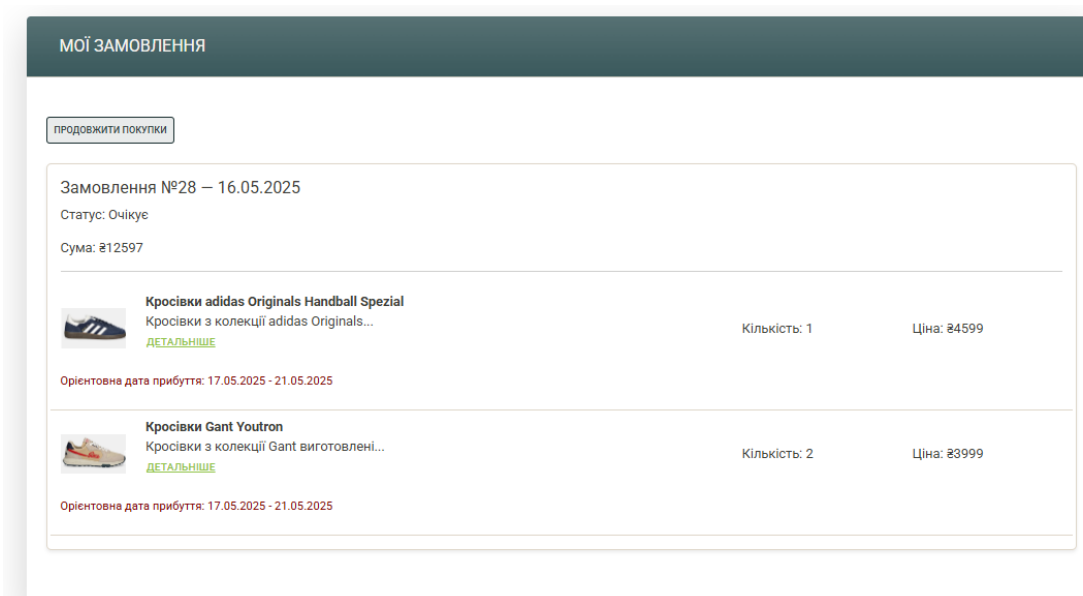


Рисунок 3.37 – Форма «Мої замовлення»

Також на даній формі присутня кнопка «Детальніше», вона реалізована для того щоб можна було переглянути, опис товару результат натискання на кнопку показано на рисунку 3.38.

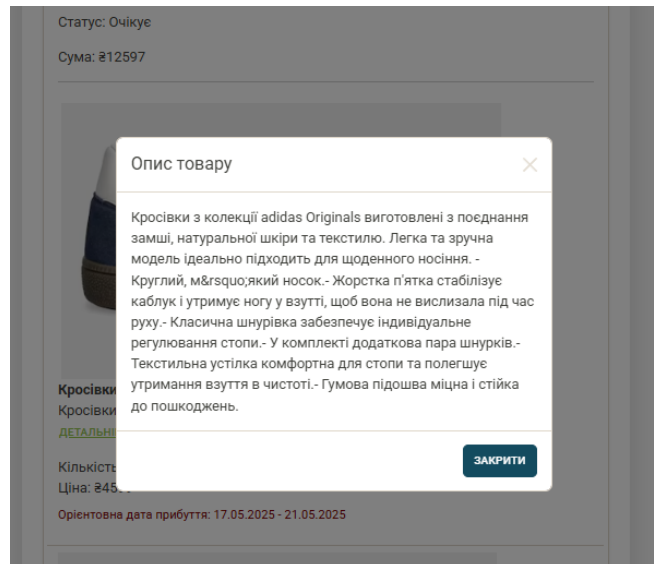


Рисунок 3.38 – Тестування кнопки «Детальніше»

Якщо спробувати увійти в адмін – панель без прав доступу за допомогою посилання URL для прикладу управління «Продукцією» URL(<https://localhost:7121/Admin/Product>), виведеться повідомлення яке зображено на рисунку 3.39.



Рисунок 3.39 – Тестування прав доступу

Також було проведено тест «Управління контентом», для прикладу вибрано форму «Продукція». На формі є «Пошук за назвою» та фільтр по «Категорія». Пошук та фільтрацію одночасно не можна використовувати, коли активне одне поле то інше очищується для прикладу в пошуку напишем «Adidas» повинно в таблиці відобразити всі позиції в яких присутнє це слово, результат пошуку, зображено на рисунку 3.40.

Список продукції

[+ СТВОРИТИ НОВУ ПРОДУКЦІЮ](#)

Пошук за назвою: Категорія:

[ПОШУК](#) [ОЧИСТИТИ](#)

Найменування товару	Ціна	Бренд	Опис	Категорія	Редагувати або видалити
Кросівки adidas Originals Handball Spezial	4599	Adidas	Кросівки з колекції adidas Originals... ДЕТАЛЬНІШЕ	Кросівки	ЗМІНИТИ ВИДАЛИТИ
Замшеві кросівки adidas Originals Forum Low CL	5099	Adidas	Замшеві кросівки adidas Originals Forum... ДЕТАЛЬНІШЕ	Кросівки	ЗМІНИТИ ВИДАЛИТИ
Замшеві кросівки adidas Originals Campus 00S	5099	Adidas	Замшеві кросівки adidas Originals Campus... ДЕТАЛЬНІШЕ	Кросівки	ЗМІНИТИ ВИДАЛИТИ
Кросівки adidas Originals SI 72 RS	4199	Adidas	Кросівки adidas Originals SI 72... ДЕТАЛЬНІШЕ	Кросівки	ЗМІНИТИ ВИДАЛИТИ

Рисунок 3.40 – Тестування «Пошук за назвою»

Для того щоб протестувати фільтр «Категорія» в ньому обираємо позицію «Шльопанці» результат зображено на рисунку 3.41.

Список продукції

[+ СТВОРИТИ НОВУ ПРОДУКЦІЮ](#)

Пошук за назвою: Категорія:

[ПОШУК](#) [ОЧИСТИТИ](#)

Найменування товару	Ціна	Бренд	Опис	Категорія	Редагувати або видалити
Шльопанці Birkenstock 1774 Arizona CosNy EXQ SYN	14599	Birkenstock	Шльопанці Birkenstock 1774 Arizona CosNy... ДЕТАЛЬНІШЕ	Шльопанці	ЗМІНИТИ ВИДАЛИТИ

Рисунок 3.41 – Тестування фільтру

Натиснувши кнопку очистити, в таблицю повертаються весь список продукції. Наступним етапом тестування буде «Створити нову продукцію», натиснувши цю кнопку відкривається форма «Створити продукцію», якщо на формі натиснути «створити», без заповнених полів результат буде зображено на рисунку 3.42

Рисунок 3.42 – Тестування форми «Створити продукцію»

Якщо в поле ціна ввести символи то виведеться таке повідомлення:

Рисунок 3.43 – Тестування поля «Ціна»

Для прикладу створим тестову продукцію (Рис. 3.44).

Створити продукцію

Назва товару: Ціна:

Бренд:

Доступні розміри: 38 39 40 41 42 43 44

Опис:
 File Edit View Insert Format
 Paragraph System Font 12pt B I U S ...
 TestDescription

id_категорії: Для кого:


Фото взуття:

Рисунок 3.44 – Створення тестової продукції


Відображення тестової продукції наведено на рисунку 3.45.

Пошук за назвою...

-- Всі категорії -- -- Для кого --



КРОСІВКИ ADIDAS ORIGINALS SL 72 RS
 Бренд: **Adidas**
 Ціна: ₴ 4199



TESTNAME
 Бренд: **Nike**
 Ціна: ₴ 1111

Рисунок 3.45 – Відображення тестової продукції на головній сторінці

TESTNAME
Бренд: Nike

Кросівки

Ціна: 1 111,00 €

Оберіть розмір:

39

TestDescription

Кількість

+

-

[ДОДАТИ ДО ЗАМОВЛЕННЯ](#)

Рисунок 3.46 – Відображення деталей тестової продукції

Після чого в управлінні продукцією замінимо дані тестової продукції зображено на рисунку 3.47.

Змінити продукцію

Назва товару TestName	Ціна 2222
Бренд Adidas	
Доступні розміри <input type="checkbox"/> 38 <input type="checkbox"/> 39 <input type="checkbox"/> 40 <input checked="" type="checkbox"/> 41 <input type="checkbox"/> 42 <input type="checkbox"/> 43 <input type="checkbox"/> 44	
Опис <div style="border: 1px solid #ccc; padding: 5px;"> <p style="font-size: 10px; margin: 0;">File Edit View Insert Format</p> <p style="font-size: 10px; margin: 0;">↶ ↷ Paragraph System Font 12pt B I U S ...</p> <p style="font-size: 12px; margin: 5px 0;">NewTestDescription</p> <p style="font-size: 10px; margin: 0;">p Build with tinyMCE</p> </div>	
id_категорії Мокасини	Для кого Чоловічі
Фото взуття <div style="border: 1px solid #ccc; padding: 2px;"> Вибрати файл Снимок екрана 2025-05-16 203529.png </div>	
<div style="display: flex; justify-content: space-around; margin: 0;"> <div style="background-color: #444; color: white; padding: 5px 15px; border-radius: 5px; font-weight: bold;">ЗМІНИТИ</div> <div style="border: 1px solid #ccc; padding: 5px 15px; border-radius: 5px; font-weight: bold;">ПОВЕРНУТИСЬ НАЗАД</div> </div>	

Рисунок 3.47 – Тестування зміни даних продукції

Після чого натиснемо «Змінити», результат зображено на рисунку 3.48.

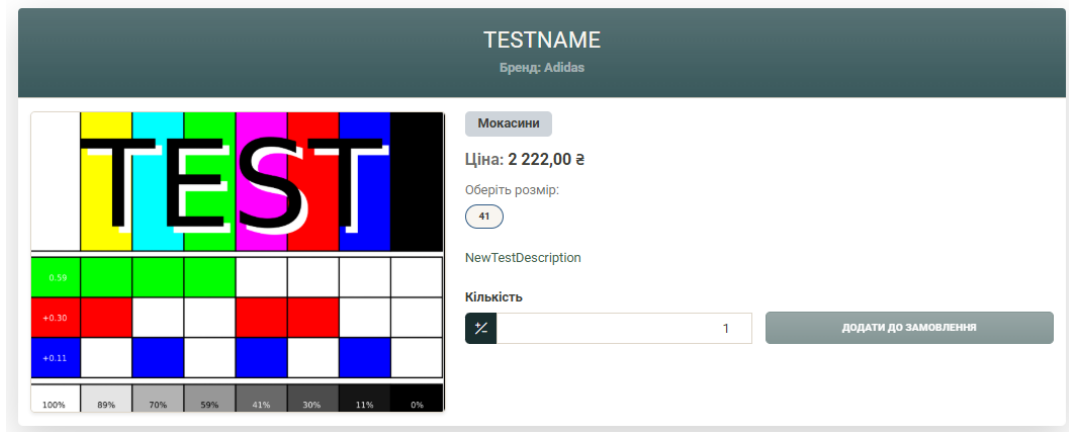


Рисунок 3.48 – Перегляд зміненої продукції

Після чого видалим тестову продукцію, зображено на рисунку 3.49.

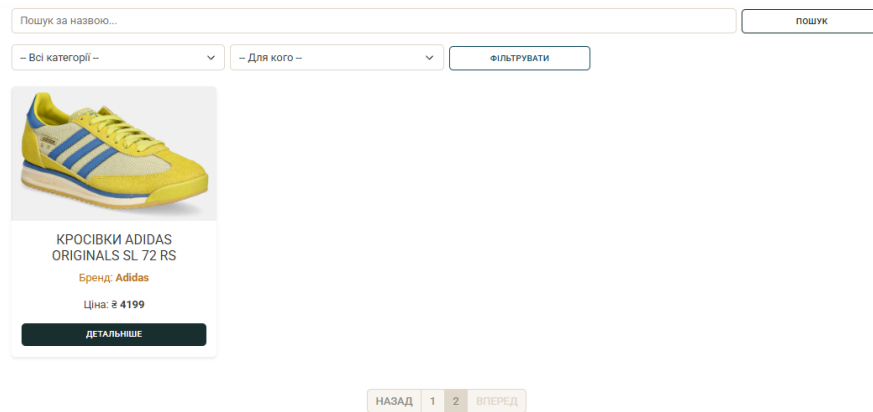


Рисунок 3.49 – Видалення продукції

Для інших форм управління даними, реалізація та логіка ідентична до форми продукція.

Тестування відображення на різних пристроях таких як: Iphone 12 pro, результат тестування зображено на рисунку 3.50.

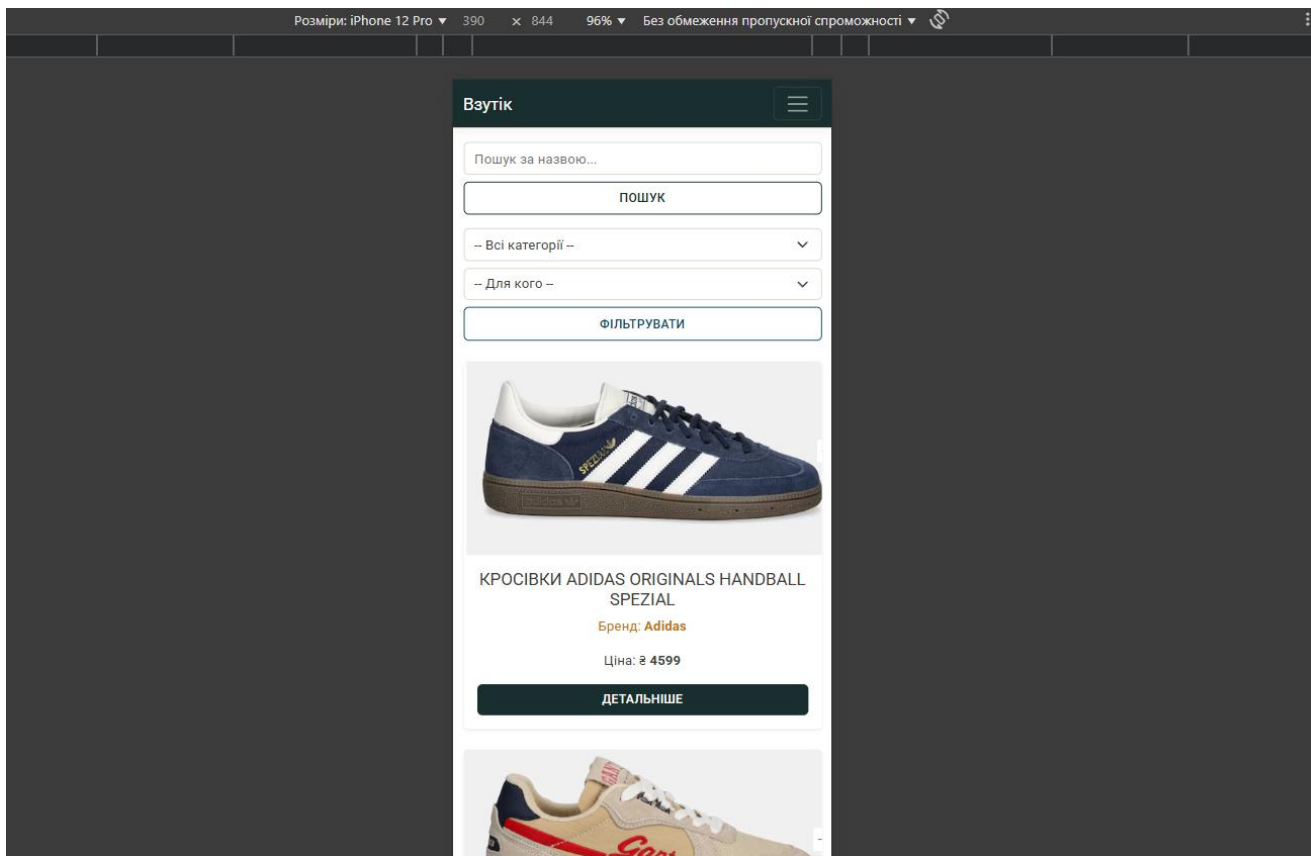


Рисунок 3.50 – Відображення на iPhone 12 pro
Для телефонів відображення буде відповідним до вищенаведеного.

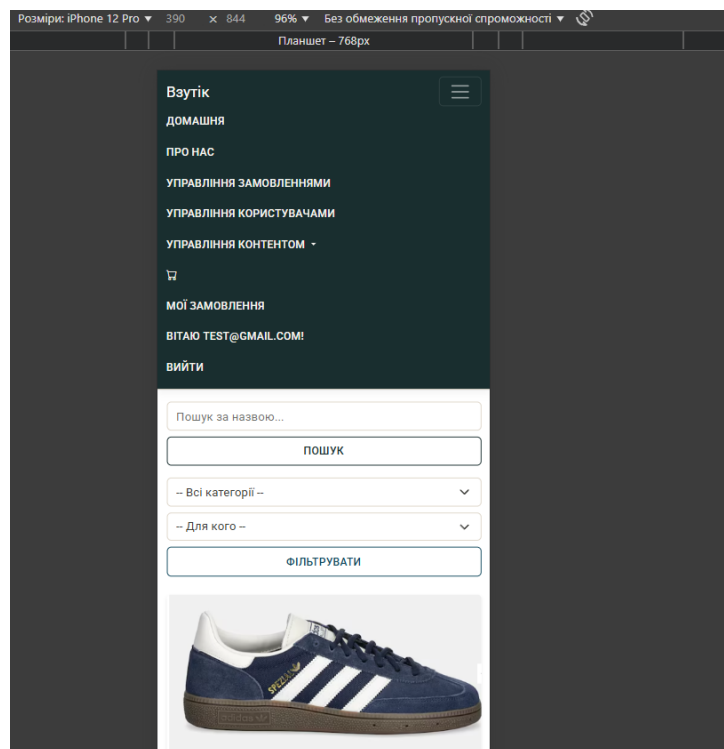


Рисунок 3.51 – Відображення навігаційного меню

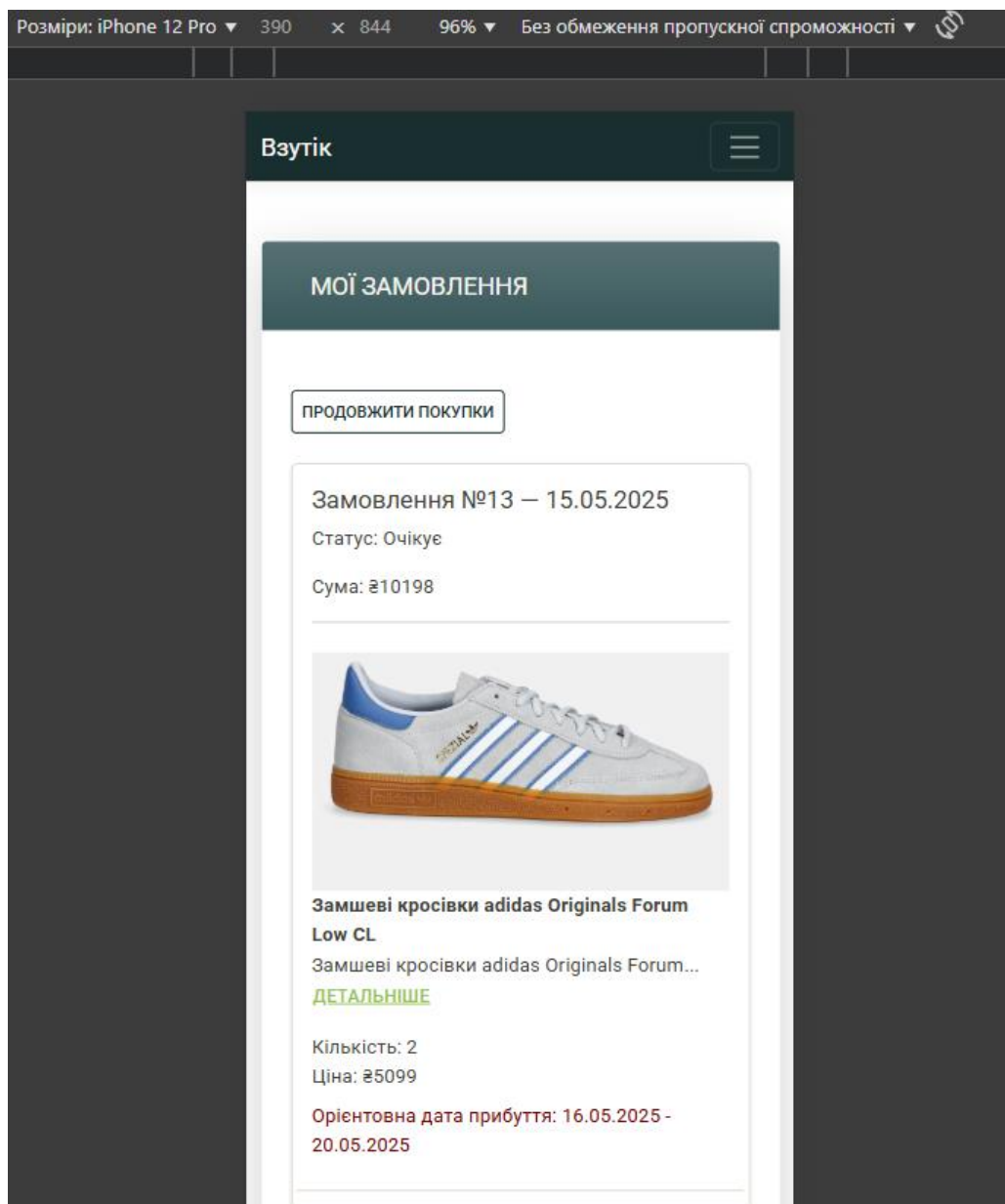


Рисунок 3.52 – Відображення форми «Мої замовлення»

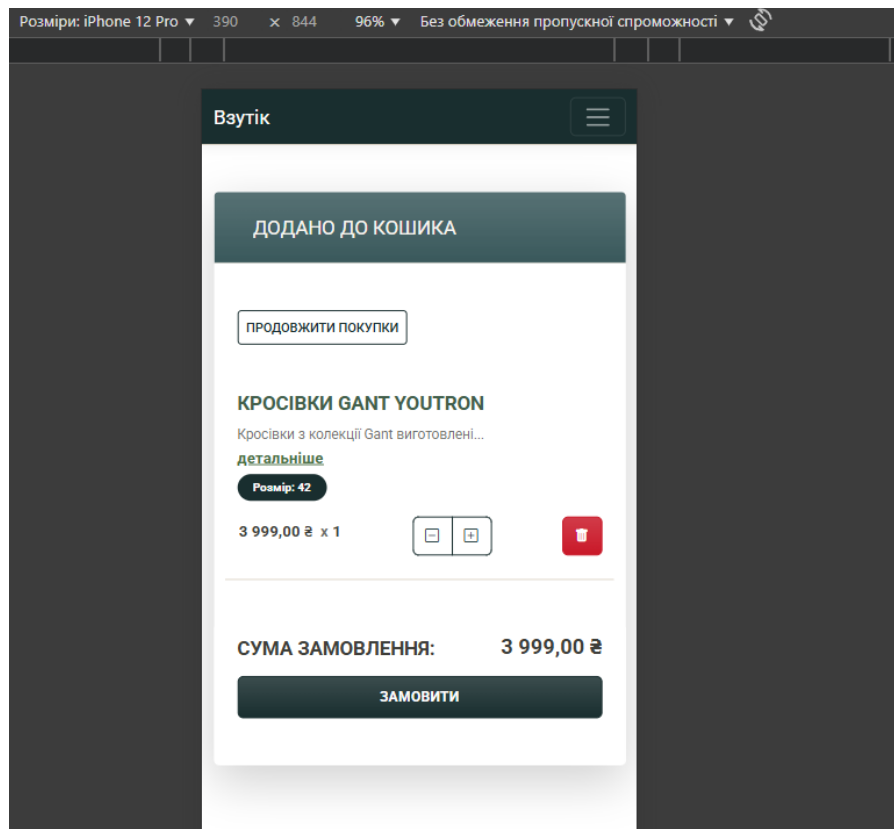


Рисунок 3.53 – Відображення «Додано до кошика»

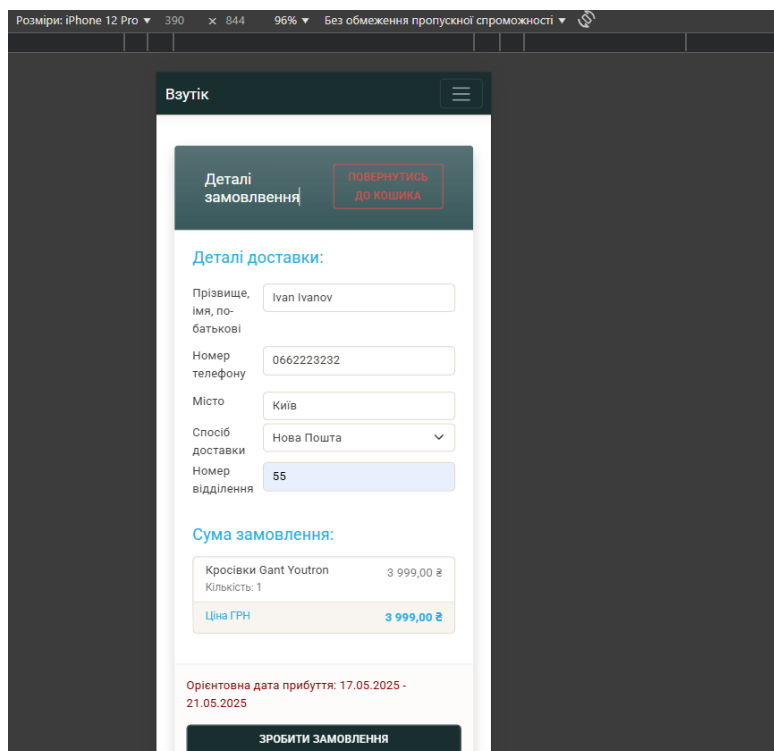


Рисунок 3.54 – Відображення «Деталі замовлення»

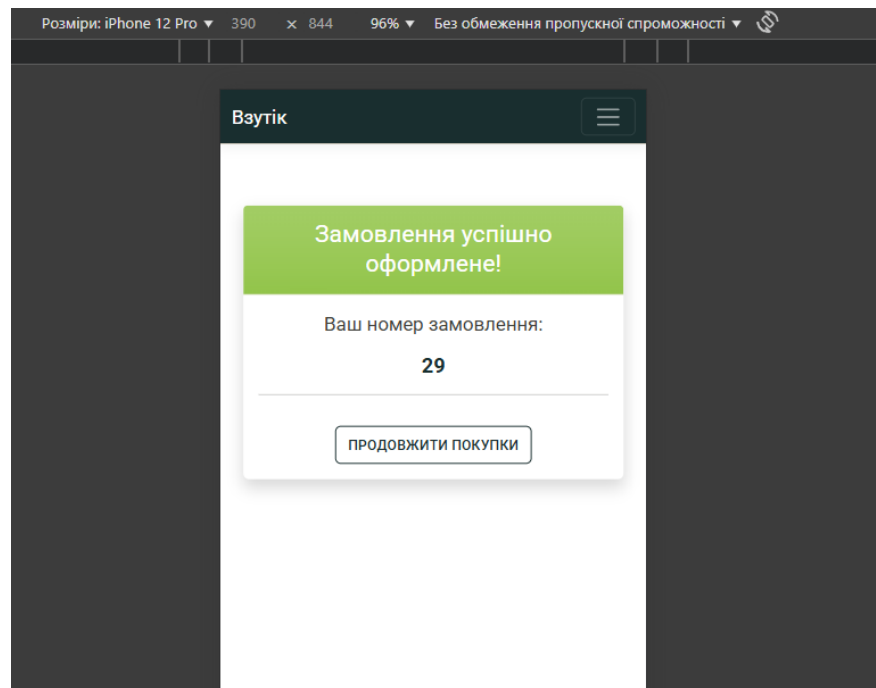


Рисунок 3.55 – Відображення створеного замовлення

Було проведено такі тести:

- Реєстрація нового користувача
- Авторизація з правильними даними
- Авторизація з неправильними даними
- Додавання товару в кошик
- Видалення товару з кошика
- Пошук за назвою
- Фільтрація за категорією та статтю
- Оформлення замовлення
- Вхід в адмін-панель неавторизованого
- Перевірка відображення елементів з різним розширенням екрану

Результатом тестування стало підтвердження того, що система працює відповідно до очікувань.

Усі перевірені функції виконувалися коректно: користувачі можуть успішно реєструватись й авторизуватись, додавати та видаляти товари з кошика, оформляти замовлення, фільтрувати та шукати продукцію. Також забезпечено обмеження доступу до адмін-панелі для неавторизованих осіб та коректне відображення елементів на різних розмірах екранів.

ВИСНОВКИ

У процесі розробки інтернет-магазину взуття для ФОП Синчук П.П. було реалізовано низку функціональних можливостей, що забезпечують ефективну роботу як для звичайних користувачів, так і для адміністратора системи. Основна мета — спростити процес продажу взуття, автоматизувати обробку замовлень та забезпечити зручне керування товарним асортиментом. Нижче наведено основні функції, реалізовані в системі.

Основні функції системи:

- реєстрація, авторизація та управління особистим кабінетом;
- перегляд каталогу, пошук і фільтрація продукції;
- додавання, редагування, оформлення замовлення;
- перегляд історії та деталей замовлень користувача;
- адмін-панель для керування продукцією, категоріями та замовленнями;
- управління замовленнями клієнтів замовлень.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. OpenCart Documentation. (n.d.). *Features Overview*. [Електронний ресурс]. – Режим доступу: <https://www.opencart.com>
2. WooCommerce. (n.d.). *Official Plugin Features*. [Електронний ресурс]. – Режим доступу: <https://woocommerce.com>
3. PrestaShop. (n.d.). *PrestaShop Features*. [Електронний ресурс]. – Режим доступу: <https://www.prestashop.com>
4. Adobe Commerce (Magento). (n.d.). *Platform Capabilities*. [Електронний ресурс]. – Режим доступу: <https://business.adobe.com/products/magento/magento-commerce.html>
5. Martin Fowler. *Microservices* [Електронний ресурс] / Martin Fowler. - 2014. Режим доступу до ресурсу: <https://martinfowler.com/articles/microservices.html>.
6. Model-View-Controller (MVC) [Електронний ресурс] - Режим доступу до ресурсу: <https://learn.microsoft.com/en-us/aspnet/core/mvc/overview/>.
7. Martin Fowler. *Presentation Domain Data Layering* [Електронний ресурс] / Martin Fowler. - 2015. - Режим доступу до ресурсу: <https://martinfowler.com/bliki/PresentationDomainDataLayering.html>.
8. Lokesh Gupta. *What is REST* [Електронний ресурс] / Lokesh Gupta. - 2022. - Режим доступу до ресурсу: <https://testfulapi.net/>.
9. *Unit testing C# in .NET Core using dotnet test and xUnit* [Електронний ресурс] - Режим доступу до ресурсу: <https://learn.microsoft.com/en-us/dotnet/core/testing/unit-testing-with-dotnet-test>.
10. Microsoft. *Документація ASP.NET Core* [електронний ресурс]. – URL: <https://learn.microsoft.com/aspnet/core> (дата звернення: 16.05.2025).
11. Microsoft. *Документація Visual Studio* [електронний ресурс]. – URL: <https://learn.microsoft.com/visualstudio> (дата звернення: 16.05.2025).
12. Microsoft. *Документація SQL Server* [електронний ресурс]. – URL: <https://learn.microsoft.com/sql> (дата звернення: 16.05.2025).
13. Закон України «Про захист прав споживачів» №1023-ХІІ (12.05.1991). – URL:

- <https://zakon.rada.gov.ua/laws/show/1023-12#Text> (дата звернення: 16.05.2025).
14. Закон України «Про захист персональних даних» №2297-VI (01.06.2010). – URL: <https://zakon.rada.gov.ua/laws/show/2297-17#Text> (дата звернення: 16.05.2025).
15. Дія.Бізнес [електронний ресурс] – офіційний вебпортал для підтримки підприємців. – URL: <https://business.diia.gov.ua> (дата звернення: 16.05.2025).
16. Кундал А.В. Цифрова трансформація та малий бізнес: виклики та перспективи. – Київ: Видавництво «Наука», 2022. – 208 с.
17. Чижмар Ю.Б. Електронна торгівля: сучасні технології. – Київ: «Перо», 2019. – 312 с.
18. Зозуленко В.І. Електронна комерція. – Київ: Ліра-К, 2018. – 256 с.
19. Розчинська О.І. Бази даних для веб: SQL Server // Навч. посібник. – Харків: ХНУ, 2022. – 180 с.
20. Вакарин І.І., Козак Л.П. Основи веб-програмування: ASP.NET. – Київ: Університет, 2021. – 240 с.
21. Медведєв Б.С. СУБД Microsoft SQL Server. – Київ: Знання, 2020. – 300 с.
22. Громов Б.О. ASP.NET MVC 5 та MS SQL Server: повний курс. – Львів: Електроніка, 2018. – 350 с.
23. Microsoft Docs: C# Guide [Електронний ресурс] - Режим доступу до ресурсу: <https://docs.microsoft.com/en-us/dotnet/csharp/>
24. ASP.NET Core Documentation [Електронний ресурс] - Режим доступу до ресурсу: <https://docs.microsoft.com/en-us/aspnet/core/>
25. Entity Framework Core [Електронний ресурс] - Режим доступу до ресурсу: <https://docs.microsoft.com/en-us/ef/core/>.
26. TypeScript Handbook [Електронний ресурс] - Режим доступу до ресурсу: <https://www.typescriptlang.org/docs/>.
27. ДСТУ ГОСТ 7.1:2006 Бібліографічний запис. Бібліографічний опис. Загальні вимоги і правила складання. – Київ: Держспоживстандарт України, 2006. – 36 с.
28. Закон України «Про рекламу» №270/96-ВР (03.07.1996). – URL: <https://zakon.rada.gov.ua/laws/show/270-14#Text> (дата звернення: 16.05.2025).

ДОДАТКИ

Додаток А. Код програми

Контролери для управління контентом:**CategoryController:**

```

using Microsoft.AspNetCore.Authorization;
using Microsoft.AspNetCore.Mvc;
using Vzutik.DataAccess.Data;
using Vzutik.DataAccess.Repository.IRepository;
using Vzutik.Models;
using Vzutik.Utility;

namespace Vzutik.Areas.Admin.Controllers
{
    [Area("Admin")]
    [Authorize(Roles = SD.Role_Admin)]
    public class CategoryController : Controller
    {
        private readonly IUnitOfWork _unitOfWork;
        public CategoryController(IUnitOfWork unitOfWork)
        {
            _unitOfWork = unitOfWork;
        }

        public IActionResult Index()
        {
            List<Category> objCategoryList = _unitOfWork.Category.GetAll().ToList();
            return View(objCategoryList);
        }

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

        [HttpPost]
        public IActionResult Create(Category obj)
        {
            if (ModelState.IsValid)
            {
                _unitOfWork.Category.Add(obj);
                _unitOfWork.Save();
                TempData["success"] = "Категорію додано";
                return RedirectToAction("Index");
            }
            return View(obj);
        }

        [HttpGet]
        public IActionResult Edit(int? id)
        {
            if (id == null || id == 0)
            {
                return NotFound();
            }

            Category? categoryFromDb = _unitOfWork.Category.Get(u => u.id_категорії == id);

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

```

```

        return View(categoryFromDb);
    }

    [HttpPost]
    public IActionResult Edit(Category obj)
    {
        if (ModelState.IsValid)
        {
            _unitOfWork.Category.Update(obj);
            _unitOfWork.Save();
            TempData["success"] = "Категорію змінено";
            return RedirectToAction("Index");
        }
        return View(obj);
    }

    [HttpGet]
    public IActionResult Delete(int? id)
    {
        if (id == null || id == 0)
        {
            return NotFound();
        }

        Category? categoryFromDb = _unitOfWork.Category.Get(u => u.id_катеропії == id);
        if (categoryFromDb == null)
        {
            return NotFound();
        }

        return View(categoryFromDb);
    }

    [HttpPost, ActionName("Delete")]
    public IActionResult DeletePOST(int? id)
    {
        Category? obj = _unitOfWork.Category.Get(u => u.id_катеропії == id);
        if (obj == null)
        {
            return NotFound();
        }
        _unitOfWork.Category.Remove(obj);
        _unitOfWork.Save();
        TempData["success"] = "Категорію видалено";
        return RedirectToAction("Index");
    }
}
}
}

```

CompanyController:

```

using Microsoft.AspNetCore.Mvc;
using Microsoft.AspNetCore.Mvc.Rendering;
using Vzutik.DataAccess.Data;
using Vzutik.DataAccess.Repository.IRepository;
using Vzutik.Models;
using Vzutik.Models.ViewModels;
using System.Text.RegularExpressions;
using Microsoft.AspNetCore.Authorization;
using Vzutik.Utility;

```

```

namespace Vzutik.Areas.Admin.Controllers
{
    [Area("Admin")]
    [Authorize(Roles = SD.Role_Admin)]

```

```

public class CompanyController : Controller
{
    private readonly IUnitOfWork _unitOfWork;
    public CompanyController(IUnitOfWork unitOfWork)
    {
        _unitOfWork = unitOfWork;
    }

    public IActionResult Index()
    {
        List<Company> objCompanyList = _unitOfWork.Company.GetAll().ToList();

        return View(objCompanyList);
    }

    public IActionResult Upsert(int? id)
    {
        if (id == null || id == 0)
        {
            return View(new Company());
        }
        else
        {
            Company company = _unitOfWork.Company.Get(u => u.id_компанії == id);
            return View(company);
        }
    }

    [HttpPost]
    public IActionResult Upsert(Company CompanyObj)
    {
        if (ModelState.IsValid)
        {
            if (CompanyObj.id_компанії == 0)
            {
                _unitOfWork.Company.Add(CompanyObj);
            }
            else
            {
                _unitOfWork.Company.Update(CompanyObj);
            }
            _unitOfWork.Save();
            TempData["success"] = "Компанію успішно додано";
            return RedirectToAction("Index");
        }
        else
        {
            return View(CompanyObj);
        }
    }

    [HttpGet]
    public IActionResult Delete(int? id)
    {
        if (id == null || id == 0)
        {
            return NotFound();
        }
    }
}

```

```

    Company? CompanyFromDb = _unitOfWork.Company.Get(u => u.id_компанії == id);
    if (CompanyFromDb == null)
    {
        return NotFound();
    }

    return View(CompanyFromDb);
}

[HttpPost, ActionName("Delete")]
public IActionResult DeletePOST(int? id)
{
    Company? obj = _unitOfWork.Company.Get(u => u.id_компанії == id);
    if (obj == null)
    {
        return NotFound();
    }

    _unitOfWork.Company.Remove(obj);
    _unitOfWork.Save();

    TempData["success"] = "Компанію успішно видалено";
    return RedirectToAction("Index");
}
}
}

```

OrderController:

```

using Microsoft.AspNetCore.Authorization;
using Microsoft.AspNetCore.Mvc;
using Vzutik.DataAccess.Repository.IRepository;
using Vzutik.Models;
using Vzutik.Models.ViewModels;
using Vzutik.Utility;

namespace Vzutik.Areas.Admin.Controllers
{
    [Area("Admin")]
    [Authorize(Roles = SD.Role_Admin)]
    public class OrderController : Controller
    {
        private readonly IUnitOfWork _unitOfWork;

        public OrderController(IUnitOfWork unitOfWork)
        {
            _unitOfWork = unitOfWork;
        }

        public IActionResult Index(string? orderId, string? name)
        {
            var orders = _unitOfWork.OrderHeader
                .GetAll(includeProperties: "Користувач")
                .AsQueryable();

            if (!string.IsNullOrEmpty(orderId) && int.TryParse(orderId, out int id))
            {
                orders = orders.Where(o => o.id_замовлення == id);
            }

            if (!string.IsNullOrEmpty(name))
            {
                orders = orders.Where(o => o.ПІБ.Contains(name, StringComparison.OrdinalIgnoreCase));
            }
        }
    }
}

```

```

    }

    return View(orders.ToList());
}

[HttpGet]
public IActionResult Details(int orderId)
{
    var orderHeader = _unitOfWork.OrderHeader.Get(u => u.id_замовлення == orderId, includeProperties: "Користувач");
    var orderDetails = _unitOfWork.OrderDetail.GetAll(
        u => u.id_замовлення == orderId,
        includeProperties: "Продукція.ProductSizes.SizeShoes");

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

    OrderVM viewModel = new()
    {
        OrderHeader = orderHeader,
        OrderDetails = orderDetails
    };

    return View(viewModel);
}

[HttpPost]
[ValidateAntiForgeryToken]
public IActionResult Details(OrderVM model)
{
    if (model?.OrderHeader == null)
    {
        TempData["error"] = "Некоректні дані замовлення.";
        return RedirectToAction("Index");
    }

    if (ModelState.IsValid)
    {
        var orderHeader = _unitOfWork.OrderHeader.Get(u => u.id_замовлення == model.OrderHeader.id_замовлення);
        if (orderHeader == null)
            return NotFound();

        orderHeader.ПІБ = model.OrderHeader.ПІБ;
        orderHeader.Номер_телефону = model.OrderHeader.Номер_телефону;
        orderHeader.Місто = model.OrderHeader.Місто;
        orderHeader.Статус_замовлення = model.OrderHeader.Статус_замовлення;

        _unitOfWork.OrderHeader.Update(orderHeader);
        _unitOfWork.Save();

        TempData["success"] = "Замовлення оновлено";
        return RedirectToAction("Details", new { orderId = model.OrderHeader.id_замовлення });
    }

    model.OrderDetails = _unitOfWork.OrderDetail.GetAll(
        u => u.id_замовлення == model.OrderHeader.id_замовлення,
        includeProperties: "Продукція.ProductSizes.SizeShoes");

    return View(model);
}

[HttpPost]
[ValidateAntiForgeryToken]
public IActionResult UpdateStatus(int id, string новийСтатус)

```

```

    {
        var замовлення = _unitOfWork.OrderHeader.Get(u => u.id_замовлення == id);
        if (замовлення == null) return NotFound();

        замовлення.Статус_замовлення = новийСтатус;
        _unitOfWork.OrderHeader.Update(замовлення);
        _unitOfWork.Save();

        TempData["success"] = "Статус замовлення оновлено";
        return RedirectToAction("Details", new { orderId = id });
    }
}
}

```

ProductController:

```

using Microsoft.AspNetCore.Mvc;
using Microsoft.AspNetCore.Mvc.Rendering;
using Vzutik.DataAccess.Repository.IRepository;
using Vzutik.Models;
using Vzutik.Models.ViewModels;
using Vzutik.Utility;
using Microsoft.AspNetCore.Authorization;
using System.Text.RegularExpressions;

namespace Vzutik.Areas.Admin.Controllers
{
    [Area("Admin")]
    [Authorize(Roles = SD.Role_Admin)]
    public class ProductController : Controller
    {
        private readonly IUnitOfWork _unitOfWork;
        private readonly IWebHostEnvironment _webHostEnvironment;

        public ProductController(IUnitOfWork unitOfWork, IWebHostEnvironment webHostEnvironment)
        {
            _unitOfWork = unitOfWork;
            _webHostEnvironment = webHostEnvironment;
        }

        public IActionResult Index(string? searchTerm, int? categoryId)
        {
            var products = _unitOfWork.Product.GetAll(includeProperties: "Category").AsQueryable();

            if (!string.IsNullOrEmpty(searchTerm))
            {
                products = products.Where(p => p.Найменування_товару.Contains(searchTerm.Trim(),
                StringComparison.OrdinalIgnoreCase));
                categoryId = null;
            }
            else if (categoryId.HasValue)
            {
                products = products.Where(p => p.id_категорії == categoryId);
            }

            ViewBag.SearchTerm = searchTerm;
            ViewBag.CategoryId = categoryId;
            ViewBag.Categories = _unitOfWork.Category.GetAll()
                .Select(c => new SelectListItem
                {
                    Text = c.Назва,
                    Value = c.id_категорії.ToString()
                });

            return View(products.ToList());
        }
    }
}

```

```

public IActionResult Upsert(int? id)
{
    ProductVM productVM = new()
    {
        Product = new Product(),
        CategoryList = _unitOfWork.Category.GetAll().Select(u => new SelectListItem
        {
            Text = u.Назва,
            Value = u.id_категорії.ToString()
        }),
        SizeShoesList = _unitOfWork.SizeShoes.GetAll().Select(u => new SelectListItem
        {
            Text = u.Розмір.ToString(),
            Value = u.id_розміру.ToString()
        }),
        GenderList = _unitOfWork.Gender.GetAll().Select(u => new SelectListItem
        {
            Text = u.Стать.ToString(),
            Value = u.id_статі.ToString()
        }),
        SelectedSizes = new List<int>()
    };

    if (id != null && id != 0)
    {
        productVM.Product = _unitOfWork.Product.Get(u => u.id_продукції == id, includeProperties: "ProductSizes");

        productVM.SelectedSizes = _unitOfWork.ProductSize
            .GetAll(u => u.id_продукції == id.Value)
            .Select(ps => ps.id_розміру)
            .ToList();
    }

    return View(productVM);
}

[HttpPost]
public IActionResult Upsert(ProductVM productVM, IFormFile? file)
{
    Console.WriteLine(" POST Upsert викликано");

    if (ModelState.IsValid)
    {
        Console.WriteLine(" ModelState.IsValid = TRUE");

        productVM.Product.Опис = Regex.Replace(productVM.Product.Опис ?? "", "<.*?>", string.Empty);
        string wwwRootPath = _webHostEnvironment.WebRootPath;

        if (file != null)
        {
            string fileName = Guid.NewGuid().ToString() + Path.GetExtension(file.FileName);
            string productPath = Path.Combine(wwwRootPath, @"images\product");

            if (!string.IsNullOrEmpty(productVM.Product.ImageUrl))
            {
                var oldImagePath = Path.Combine(wwwRootPath, productVM.Product.ImageUrl.TrimStart('\'));
                if (System.IO.File.Exists(oldImagePath))
                {
                    System.IO.File.Delete(oldImagePath);
                }
            }

            using (var fileStream = new FileStream(Path.Combine(productPath, fileName), FileMode.Create))
            {

```

```

        file.CopyTo(fileStream);
    }

    productVM.Product.ImageUrl = @"images\product\" + fileName;
}

try
{
    if (productVM.Product.id_продукції == 0)
    {
        _unitOfWork.Product.Add(productVM.Product);
        Console.WriteLine(" До Save, id_продукції = " + productVM.Product.id_продукції);
        _unitOfWork.Save();
        Console.WriteLine(" Після Save, id_продукції = " + productVM.Product.id_продукції);

        foreach (var sizeId in productVM.SelectedSizes)
        {
            Console.WriteLine($" 📏 Додається розмір: {sizeId} ");
            _unitOfWork.ProductSize.Add(new ProductSize
            {
                id_продукції = productVM.Product.id_продукції,
                id_розміру = sizeId
            });
        }
    }
    else
    {
        _unitOfWork.Product.Update(productVM.Product);

        var existingSizes = _unitOfWork.ProductSize
            .GetAll(u => u.id_продукції == productVM.Product.id_продукції)
            .ToList();

        _unitOfWork.ProductSize.RemoveRange(existingSizes);

        foreach (var sizeId in productVM.SelectedSizes)
        {
            _unitOfWork.ProductSize.Add(new ProductSize
            {
                id_продукції = productVM.Product.id_продукції,
                id_розміру = sizeId
            });
        }

        _unitOfWork.Save();
        TempData["success"] = "Продукцію збережено";
        return RedirectToAction("Index");
    }
    catch (Exception ex)
    {
        Console.WriteLine("❌ Виняток при збереженні: " + ex.Message);
        TempData["error"] = "Сталася помилка при збереженні продукту.";
    }
}
else
{
    Console.WriteLine("❌ ModelState.IsValid = FALSE");
    foreach (var state in ModelState)
    {
        foreach (var error in state.Value.Errors)
        {
            Console.WriteLine($"❌ Поле: {state.Key}, Помилка: {error.ErrorMessage}");
        }
    }
}
}

```

```

productVM.GenderList = _unitOfWork.Gender.GetAll().Select(u => new SelectListItem
{
    Text = u.Стать,
    Value = u.id_статі.ToString()
});
productVM.CategoryList = _unitOfWork.Category.GetAll().Select(u => new SelectListItem
{
    Text = u.Назва,
    Value = u.id_категорії.ToString()
});

productVM.SizeShoesList = _unitOfWork.SizeShoes.GetAll().Select(u => new SelectListItem
{
    Text = u.Розмір.ToString(),
    Value = u.id_розміру.ToString()
});

return View(productVM);
}

[HttpGet]
public IActionResult Delete(int? id)
{
    if (id == null || id == 0)
        return NotFound();

    Product? productFromDb = _unitOfWork.Product.Get(u => u.id_продукції == id);
    if (productFromDb == null)
        return NotFound();

    return View(productFromDb);
}

[HttpPost, ActionName("Delete")]
public IActionResult DeletePOST(int? id)
{
    Product? obj = _unitOfWork.Product.Get(u => u.id_продукції == id);
    if (obj == null)
        return NotFound();

    var productSizes = _unitOfWork.ProductSize
        .GetAll(u => u.id_продукції == obj.id_продукції).ToList();

    _unitOfWork.ProductSize.RemoveRange(productSizes);

    if (!string.IsNullOrEmpty(obj.ImageUrl))
    {
        string wwwRootPath = _webHostEnvironment.WebRootPath;
        string imagePath = Path.Combine(wwwRootPath, obj.ImageUrl.TrimStart('\'));

        if (System.IO.File.Exists(imagePath))
            System.IO.File.Delete(imagePath);
    }

    _unitOfWork.Product.Remove(obj);
    _unitOfWork.Save();

    TempData["success"] = "Продукцію та доступні розміри видалено";
    return RedirectToAction("Index");
}

```

```

    }
}
}

```

SizeShoesController:

```

using Microsoft.AspNetCore.Authorization;
using Microsoft.AspNetCore.Mvc;
using Vzutik.DataAccess.Data;
using Vzutik.DataAccess.Repository.IRepository;
using Vzutik.Models;
using Vzutik.Utility;

namespace Vzutik.Areas.Admin.Controllers
{
    [Area("Admin")]
    [Authorize(Roles = SD.Role_Admin)]
    public class SizeShoesController : Controller
    {
        private readonly IUnitOfWork _unitOfWork;
        public SizeShoesController(IUnitOfWork unitOfWork)
        {
            _unitOfWork = unitOfWork;
        }

        public IActionResult Index()
        {
            List<SizeShoes> objSizeShoesList = _unitOfWork.SizeShoes.GetAll().ToList();
            return View(objSizeShoesList);
        }

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

        [HttpPost]
        public IActionResult Create(SizeShoes obj)
        {
            if (ModelState.IsValid)
            {
                _unitOfWork.SizeShoes.Add(obj);
                _unitOfWork.Save();
                TempData["success"] = "Новий розмір додано";
                return RedirectToAction("Index");
            }
            return View(obj);
        }

        [HttpGet]
        public IActionResult Edit(int? id)
        {
            if (id == null || id == 0)
            {
                return NotFound();
            }

            SizeShoes? SizeShoesFromDb = _unitOfWork.SizeShoes.Get(u => u.id_розмір == id);

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

            return View(SizeShoesFromDb);
        }
    }
}

```

```

[HttpPost]
public IActionResult Edit(SizeShoes obj)
{
    if (ModelState.IsValid)
    {
        _unitOfWork.SizeShoes.Update(obj);
        _unitOfWork.Save();
        TempData["success"] = "Розмір змінено";
        return RedirectToAction("Index");
    }
    return View(obj);
}

[HttpGet]
public IActionResult Delete(int? id)
{
    if (id == null || id == 0)
    {
        return NotFound();
    }

    SizeShoes? SizeShoesFromDb = _unitOfWork.SizeShoes.Get(u => u.id_розмір == id);
    if (SizeShoesFromDb == null)
    {
        return NotFound();
    }

    return View(SizeShoesFromDb);
}

[HttpPost, ActionName("Delete")]
public IActionResult DeletePOST(int? id)
{
    SizeShoes? obj = _unitOfWork.SizeShoes.Get(u => u.id_розмір == id);
    if (obj == null)
    {
        return NotFound();
    }
    _unitOfWork.SizeShoes.Remove(obj);
    _unitOfWork.Save();
    TempData["success"] = "Розмір видалено";
    return RedirectToAction("Index");
}
}
}

```

UserController:

```

using Microsoft.AspNetCore.Identity;
using Microsoft.AspNetCore.Mvc;
using Vzutik.Models.ViewModels;

namespace Vzutik.Areas.Admin.Controllers
{
    [Area("Admin")]
    public class UserController : Controller
    {
        private readonly UserManager<IdentityUser> _userManager;
        private readonly RoleManager<IdentityRole> _roleManager;

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

        public IActionResult Index(string? searchEmail)

```

```

{
    var users = _userManager.Users.AsQueryable();

    if (!string.IsNullOrEmpty(searchEmail))
    {
        users = users.Where(u => u.Email.Contains(searchEmail));
    }

    return View(users.ToList());
}

public async Task<IActionResult> ManageRole(string userId)
{
    var user = await _userManager.FindByIdAsync(userId);
    if (user == null) return NotFound();

    var allRoles = _roleManager.Roles.Select(r => r.Name).ToList();
    var userRoles = await _userManager.GetRolesAsync(user);

    var model = new ManageRoleVM
    {
        UserId = user.Id,
        Email = user.Email,
        CurrentRole = userRoles.FirstOrDefault(),
        AllRoles = allRoles
    };

    return View(model);
}

[HttpPost]
public async Task<IActionResult> ManageRole(ManageRoleVM model)
{
    var user = await _userManager.FindByIdAsync(model.UserId);
    if (user == null) return NotFound();

    var roles = await _userManager.GetRolesAsync(user);
    await _userManager.RemoveFromRolesAsync(user, roles);
    await _userManager.AddToRoleAsync(user, model.SelectedRole);

    return RedirectToAction("Index");
}
}
}

```

ViewsCategory:

```

@model List<Category>
<partial name="_Notification"/>
<div class="card shadow border-0 mt-4">
    <div class="card-header bg-secondary bg-gradient ml-0 py-3">
        <div class="row">
            <div class="col-12 text-center">
                <h2 class="text-white py-2">Список категорій</h2>
            </div>
        </div>
    </div>
    <div class="row">
        <div class="col-6"></div>
        <div class="col-6 text-end py-4">
            <a asp-controller="Category" asp-action="Create" class="btn btn-primary">
                <i class="bi bi-plus-circle"></i> Створити нову категорію
            </a>
        </div>
    </div>
</div>

```

```

<table class="table table-bordered table-striped">
  <thead>
    <tr>
      <th>
        Назва категорії
      </th>
      <th>
        Код продукції
      </th>
      <th>
      </th>
    </tr>
  </thead>
  <tbody>
    @foreach (var obj in Model.OrderBy(u => u.Порядок_відображення))
    {
      <tr>
        <td>@obj.Назва</td>
        <td>@obj.Порядок_відображення</td>
        <td>
          <div class="w-75 btn-group" role="group">
            <a asp-controller="Category" asp-action="Edit" asp-route-id="@obj.id_категорії" class="btn btn-primary
mx-2">
              <i class="bi bi-pencil"></i> Змінити
            </a>
            <a asp-controller="Category" asp-action="Delete" asp-route-id="@obj.id_категорії" class="btn btn-danger
mx-2">
              <i class="bi bi-trash3"></i> Видалити
            </a>
          </div>
        </td>
      </tr>
    }
  </tbody>
</table>
</div>

```

DeleteCategory:

@model Category

```

<div class="card shadow border-0 mt-4">
  <div class="card-header bg-secondary bg-gradient py-3">
    <div class="row">
      <div class="col-12 text-center">
        <h2 class="text-white py-2">Видалити категорію</h2>
      </div>
    </div>
  </div>
  <div class="card-body p-4">
    <form method="post">
      <input asp-for="id_категорії" hidden />
      <div class="text-danger asp-validation-summary="ModelOnly"></div>

      <div class="mb-3 row p-1">
        <label asp-for="Назва" class="p-0 fw-bold"></label>
        <input asp-for="Назва" disabled class="form-control border border-primary bg-light" />
      </div>

      <div class="mb-3 row p-1">
        <label asp-for="Порядок_відображення" class="p-0 fw-bold"></label>
        <input asp-for="Порядок_відображення" disabled class="form-control border border-primary bg-light" />
      </div>
    </form>
  </div>
</div>

```

```

<div class="row mt-3">
  <div class="col-6 col-md-4">
    <button type="submit" class="btn btn-danger form-control">Видалити</button>
  </div>
  <div class="col-6 col-md-4">
    <a asp-controller="Category" asp-action="Index" class="btn btn-outline-secondary form-control">
      Повернутись назад
    </a>
  </div>
</div>
</form>
</div>
</div>

```

EditCategory:

@model Category

```

<div class="card shadow border-0 mt-4">
  <div class="card-header bg-secondary bg-gradient py-3">
    <div class="row">
      <div class="col-12 text-center">
        <h2 class="text-white py-2">Видалити категорію</h2>
      </div>
    </div>
  </div>
  <div class="card-body p-4">
    <form method="post">
      <div class="text-danger asp-validation-summary="ModelOnly"></div>

      <div class="mb-3 row p-1">
        <label asp-for="Назва" class="p-0 fw-bold"></label>
        <input asp-for="Назва" class="form-control border border-primary bg-light" />
        <span asp-validation-for="Назва" class="text-danger"></span>
      </div>

      <div class="mb-3 row p-1">
        <label asp-for="Порядок_відображення" class="p-0 fw-bold"></label>
        <input asp-for="Порядок_відображення" class="form-control border border-primary bg-light" />
        <span asp-validation-for="Порядок_відображення" class="text-danger"></span>
      </div>

      <div class="row mt-3">
        <div class="col-6 col-md-4">
          <button type="submit" class="btn btn-primary form-control">Змінити</button>
        </div>
        <div class="col-6 col-md-4">
          <a asp-controller="Category" asp-action="Index" class="btn btn-outline-secondary form-control">
            Повернутись назад
          </a>
        </div>
      </div>
    </form>
  </div>
</div>

```

```

@section Scripts {
  @{
    <partial name="_ValidationScriptsPartial" />
  }
}

```

CompanyIndex:

@model Category

```

<div class="card shadow border-0 mt-4">
  <div class="card-header bg-secondary bg-gradient py-3">
    <div class="row">
      <div class="col-12 text-center">
        <h2 class="text-white py-2">Видалити категорію</h2>
      </div>
    </div>
  </div>
  <div class="card-body p-4">
    <form method="post">
      <div class="text-danger asp-validation-summary="ModelOnly"></div>

      <div class="mb-3 row p-1">
        <label asp-for="Назва" class="p-0 fw-bold"></label>
        <input asp-for="Назва" class="form-control border border-primary bg-light" />
        <span asp-validation-for="Назва" class="text-danger"></span>
      </div>

      <div class="mb-3 row p-1">
        <label asp-for="Порядок_відображення" class="p-0 fw-bold"></label>
        <input asp-for="Порядок_відображення" class="form-control border border-primary bg-light" />
        <span asp-validation-for="Порядок_відображення" class="text-danger"></span>
      </div>

      <div class="row mt-3">
        <div class="col-6 col-md-4">
          <button type="submit" class="btn btn-primary form-control">Змінити</button>
        </div>
        <div class="col-6 col-md-4">
          <a asp-controller="Category" asp-action="Index" class="btn btn-outline-secondary form-control">
            Повернутись назад
          </a>
        </div>
      </div>
    </form>
  </div>
</div>

@section Scripts {
  @{
    <partial name="_ValidationScriptsPartial" />
  }
}

```

UpsertCompany:

@model Company

```

<div class="card shadow border-0 mt-4">
  <div class="card-header bg-secondary bg-gradient ml-0 py-3">
    <div class="row">
      <div class="col-12 text-center">
        <h2 class="text-white py-2">@(Model.id_компанії!=0?"Змінити":"Додати") компанію</h2>
      </div>
    </div>
  </div>
  <div class="card-body p-4">
    <form method="post" asp-action="Upsert" asp-controller="Company" class="row" enctype="multipart/form-data">

      <input asp-for="id_компанії" hidden />

      <div class="row">
        <div class="col-10">
          <div class="border p-3">
            <div class="mb-3 row p-1 col-12">
              <label asp-for="Назва" class="p-0 fw-bold"></label>

```

```

        <input asp-for="Назва" class="form-control border border-primary bg-light" />
        <span asp-validation-for="Назва" class="text-danger"></span>
    </div>
    <div class="mb-3 row p-1 col-12">
        <label asp-for="Номер_телефону" class="p-0 fw-bold"></label>
        <div class="input-group">
            <span class="input-group-text">+38</span>
            <input asp-for="Номер_телефону" class="form-control border border-primary bg-light"
                type="tel" inputmode="numeric" placeholder="066XXXXXXX"
                pattern="\d{10}" maxlength="10" />
        </div>
        <span asp-validation-for="Номер_телефону" class="text-danger"></span>
    </div>

    <div class="mb-3 row p-1 col-12">
        <label asp-for="Адреса" class="p-0 fw-bold"></label>
        <input asp-for="Адреса" class="form-control border border-primary bg-light" />
        <span asp-validation-for="Адреса" class="text-danger"></span>
    </div>
    <div class="mb-3 row p-1 col-12">
        <label asp-for="Місто" class="p-0 fw-bold"></label>
        <input asp-for="Місто" class="form-control border border-primary bg-light" />
        <span asp-validation-for="Місто" class="text-danger"></span>
    </div>
    <div class="row">
        <div class="col-6 col-md-4">
            @if (Model.id_компанії != 0)
            {
                <button type="submit" class="btn btn-primary form-control">Змінити</button>
            }
            else
            {
                <button type="submit" class="btn btn-primary form-control">Створити</button>
            }
        </div>
        <div class="col-6 col-md-4">
            <a asp-controller="Company" asp-action="Index" class="btn btn-outline-primary form-control">
                Повернутись назад
            </a>
        </div>
    </div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</form>
</div>
</div>

```

```

@section Scripts {
    <script>
        tinymce.init({
            selector: 'textarea',
            plugins: [

            ],
            toolbar: 'undo redo | blocks fontfamily fontsize | bold italic underline strikethrough | link image media table mergetags |
addcomment showcomments | spellcheckdialog a11ycheck typography | align lineheight | checklist numlist bullist indent outdent |
emoticons charmap | removeformat',
            tinycomments_mode: 'embedded',
            tinycomments_author: 'Author name',
            mergetags_list: [

```

```

        { value: 'First.Name', title: 'First Name' },
        { value: 'Email', title: 'Email' },
    ],
    ai_request: (request, respondWith) => respondWith.string() => Promise.reject('See docs to implement AI Assistant'),
  });
</script>
@{
  <partial name="_ValidationScriptsPartial" />
}
}

```

OrderIndex:

```
@model IEnumerable<Vzutik.Models.OrderHeader>
```

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

```
<h2 class="text-center mb-4">Список замовлень</h2>
```

```

<form method="get" class="row mb-4">
  <div class="col-md-4">
    <input type="text" name="orderId" class="form-control" placeholder="Пошук за номером замовлення"
      value="@Context.Request.Query["orderId"]" />
  </div>
  <div class="col-md-4">
    <input type="text" name="name" class="form-control" placeholder="Пошук за ПІБ"
      value="@Context.Request.Query["name"]" />
  </div>
  <div class="col-md-2">
    <button type="submit" class="btn btn-primary w-100">Пошук</button>
  </div>
  <div class="col-md-2">
    <a asp-action="Index" class="btn btn-outline-secondary w-100">Очистити</a>
  </div>
</form>

```

```

<table class="table table-striped table-bordered">
  <thead class="table-dark">
    <tr>
      <th>Номер</th>
      <th>ПІБ</th>
      <th>Місто</th>
      <th>Телефон</th>
      <th>Дата</th>
      <th>Статус</th>
      <th>Дії</th>
    </tr>
  </thead>
  <tbody>
    @foreach (var order in Model)
    {
      <tr>
        <td>@order.id_замовлення</td>
        <td>@order.ПІБ</td>
        <td>@order.Місто</td>
        <td>@order.Номер_телефону</td>
        <td>@order.Дата_замовлення.ToShortDateString()</td>
        <td>@order.Статус_замовлення</td>
        <td>
          <a asp-action="Details" asp-route-orderId="@order.id_замовлення" class="btn btn-sm btn-
            primary">Переглянути</a>
        </td>
      </tr>
    }
  </tbody>
</table>

```

```

    }
  </tbody>
</table>

```

OrderDetails:

```
@model Vzutik.Models.ViewModels.OrderVM
```

```

@{
    double загальнаСума = 0;
}

<div class="container my-4">
    <div class="card shadow-sm">
        <div class="card-body">
            <div class="d-flex flex-wrap justify-content-between align-items-center mb-3">
                <h4 class="mb-3 mb-md-0">Детали замовлення №@Model.OrderHeader.id_замовлення</h4>

                <form method="post" asp-action="UpdateStatus" class="d-flex align-items-center mt-2 mt-md-0">
                    @Html.AntiForgeryToken()
                    <input type="hidden" name="id" value="@Model.OrderHeader.id_замовлення" />
                    <label for="новийСтатус" class="form-label fw-semibold me-2 mb-0">Статус:</label>
                    <select id="новийСтатус" name="новийСтатус" class="form-select me-2" style="width:auto;">
                        <option selected disabled>Оберіть статус</option>
                        <option value="Очікує" selected="@((Model.OrderHeader.Статус_замовлення == "Очікує"))">Очікує</option>
                        <option value="Опрацьовується" selected="@((Model.OrderHeader.Статус_замовлення == "Опрацьовується"))">Опрацьовується</option>
                        <option value="Доставлено" selected="@((Model.OrderHeader.Статус_замовлення == "Доставлено"))">Доставлено</option>
                        <option value="Скасовано" selected="@((Model.OrderHeader.Статус_замовлення == "Скасовано"))">Скасовано</option>
                    </select>
                    <button type="submit" id="SaveBtn" class="btn btn-primary">Зберегти</button>
                </form>
            </div>

            <div class="mb-4">
                <p><strong>ПІБ:</strong> @Model.OrderHeader.ПІБ</p>
                <p><strong>Місто:</strong> @Model.OrderHeader.Місто</p>
                <p><strong>Телефон:</strong> @Model.OrderHeader.Номер_телефону</p>
                <p><strong>Номер відділення:</strong> @Model.OrderHeader.Номер_відділення</p>
            </div>
            <div class="table-responsive mb-3">
                <table class="table table-striped table-hover align-middle">
                    <thead class="table-light">
                        <tr>
                            <th>Товар</th>
                            <th class="text-center">Кількість</th>
                            <th class="text-end">Ціна</th>
                            <th class="text-end">Сума</th>
                        </tr>
                    </thead>
                    <tbody>
                        @foreach (var detail in Model.OrderDetails)
                        {
                            var сума = detail.Кількість * detail.Ціна;
                            загальнаСума += сума;

                            <tr>
                                <td>@detail.Продукція.Найменування_товару</td>
                                <td class="text-center">@detail.Кількість</td>
                                <td class="text-end">@detail.Ціна.ToString("0.00")</td>
                                <td class="text-end fw-semibold">@сума.ToString("0.00")</td>
                            </tr>
                        }
                    </tbody>
                </table>
            </div>
        </div>
    </div>

```

```

        </tr>
    }
</tbody>
<tfoot class="table-light">
    <tr>
        <td colspan="4" class="text-end fw-bold">Загальна сума:</td>
        <td class="text-end fw-bold text-success">@@загальнаСума.ToString("0.00")</td>
    </tr>
</tfoot>
</table>
</div>

<div class="text-start">
    <a asp-action="Index" class="btn btn-secondary">← Назад до списку замовлень</a>
</div>
</div>
</div>
</div>

```

ProductIndex:

```
@model List<Product>
```

```
@using Microsoft.AspNetCore.Mvc.Rendering
```

```

<div class="card shadow border-0 mt-4">
    <div class="card-header bg-secondary bg-gradient ml-0 py-3">
        <div class="row">
            <div class="col-12 text-center">
                <h2 class="text-white py-2">Список продукції</h2>
            </div>
        </div>
        <div class="row">
            <div class="col-6"></div>
            <div class="col-6 text-end py-4">
                <a asp-controller="Product" asp-action="Upsert" class="btn btn-primary">
                    <i class="bi bi-plus-circle"></i> Створити нову продукцію
                </a>
            </div>
        </div>
    </div>

    <form method="get" asp-action="Index" class="row g-2 mb-4 align-items-end">
        <div class="col-md-4">
            <label for="searchTerm" class="form-label">Пошук за назвою</label>
            <input type="text" name="searchTerm" id="searchTerm" class="form-control"
                placeholder="Введіть назву товару..."
                value="@ViewBag.SearchTerm"
                onchange="if(this.value) document.getElementById('categoryId').value=";" />
        </div>

        <div class="col-md-4">
            <label for="categoryId" class="form-label">Категорія</label>
            <select class="form-select" id="categoryId" name="categoryId"
                onchange="document.getElementById('searchTerm').value=";">
                <option value="">-- Виберіть категорію --</option>
                @foreach (var category in ViewBag.Categories as IEnumerable<SelectListItem>)
                {
                    <option value="@category.Value" selected="@((category.Value == ViewBag.id_категорії?.ToString()))">
                        @category.Text
                    </option>
                }
            </select>
        </div>

    </form>

```

```

        <button type="submit" class="btn btn-primary w-100">Пошук</button>
    </div>

    <div class="col-md-2">
        <a class="btn btn-outline-secondary w-100" href="@Url.Action("Index", "Product", new { area = "Admin" })">
            Очистити
        </a>
    </div>
</form>

<table class="table table-bordered table-striped">
    <thead>
        <tr>
            <th>Найменування товару</th>
            <th>Ціна</th>
            <th>Бренд</th>
            <th>Опис</th>
            <th>Категорія</th>
            <th>Редагувати або видалити</th>
        </tr>
    </thead>
    <tbody>
        @foreach (var obj in Model)
        {
            <tr>
                <td>@obj.Найменування_товару</td>
                <td>@obj.Ціна</td>
                <td>@obj.Бренд</td>
                @{
                    var words = obj.Опис?.Split(' ');
                    var preview = words != null ? string.Join(" ", words.Take(5)) + (words.Length > 5 ? "...": "") : "";
                }
                <td>
                    <span>@preview</span><br />
                    <button type="button"
                        class="btn btn-link p-0"
                        data-description="@obj.Опис"
                        onclick="openDescriptionFromAttr(this)">
                        детальніше
                    </button>
                </td>
                <td>@obj.Category.Назва</td>
                <td>
                    <div class="w-75 btn-group" role="group">
                        <a asp-controller="Product" asp-action="Upsert" asp-route-id="@obj.id_продукції" class="btn btn-primary
mx-2">
                            <i class="bi bi-pencil"></i> Змінити
                        </a>
                        <a asp-controller="Product" asp-action="Delete" asp-route-id="@obj.id_продукції" class="btn btn-danger mx-
2">
                            <i class="bi bi-trash3"></i> Видалити
                        </a>
                    </div>
                </td>
            </tr>
        }
    </tbody>
</table>
<div class="modal fade" id="imageModal" tabindex="-1" aria-labelledby="imageModalLabel" aria-hidden="true">
    <div class="modal-dialog modal-dialog-centered modal-lg">

```

```

<div class="modal-content">
  <div class="modal-body p-0 text-center">
    <img id="modalImage" src="" class="img-fluid rounded" alt="Зображення товару" />
  </div>
  <div class="modal-footer">
    <button type="button" class="btn btn-secondary" data-bs-dismiss="modal">Закрити</button>
  </div>
</div>
</div>
</div>
</div>

<div class="modal fade" id="descriptionModal" tabindex="-1" aria-labelledby="descriptionModalLabel" aria-hidden="true">
  <div class="modal-dialog modal-dialog-centered modal-md">
    <div class="modal-content">
      <div class="modal-header">
        <h5 class="modal-title">Опис товару</h5>
        <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Закрити"></button>
      </div>
      <div class="modal-body">
        <p id="modalDescription" class="mb-0"></p>
      </div>
      <div class="modal-footer">
        <button type="button" class="btn btn-secondary" data-bs-dismiss="modal">Закрити</button>
      </div>
    </div>
  </div>
</div>
</div>

</div>

@section Scripts {
  <script src="~/js/product.js"></script>
  <script>
    function toggleDescription(cell) {
      const span = cell.querySelector("span");
      const fullText = span.getAttribute("data-full");
      const isExpanded = span.getAttribute("data-expanded") === "true";

      if (isExpanded) {
        const words = fullText.split(' ');
        const preview = words.slice(0, 5).join(' ') + (words.length > 5 ? "... : """);
        span.innerText = preview;
        span.setAttribute("data-expanded", "false");
      } else {
        span.innerText = fullText;
        span.setAttribute("data-expanded", "true");
      }
    }

    function openDescriptionFromAttr(element) {
      const description = element.getAttribute('data-description');
      const modalDescription = document.getElementById("modalDescription");
      modalDescription.textContent = description;
      const modal = new bootstrap.Modal(document.getElementById("descriptionModal"));
      modal.show();
    }

    function openImage(url) {
      console.log(" Відкриваємо зображення: ", url);
      const modalImage = document.getElementById("modalImage");
      modalImage.src = url;
      const modal = new bootstrap.Modal(document.getElementById("imageModal"));
      modal.show();
    }
  </script>
}

```

ProductUpsert:

@model ProductVM

```

<div class="card shadow border-0 mt-4">
  <div class="card-header bg-secondary bg-gradient ml-0 py-3">
    <div class="row">
      <div class="col-12 text-center">
        <h2 class="text-white py-2">@(Model.Product.id_продукції!=0?"Змінити":"Створити") продукцію</h2>
      </div>
    </div>
  </div>
  <div class="card-body p-4">
    <form method="post" asp-action="Upsert" asp-controller="Product" enctype="multipart/form-data">
      <input asp-for="Product.id_продукції" hidden />
      <input asp-for="Product.ImageUrl" hidden />

      <div class="row">
        <div class="col-md-8">
          <div class="border p-3">
            <div class="row">
              <div class="col-md-6 mb-3">
                <label asp-for="Product.Найменування_товару" class="fw-bold"></label>
                <input asp-for="Product.Найменування_товару" class="form-control border border-primary bg-light" />
                <span asp-validation-for="Product.Найменування_товару" class="text-danger"></span>
              </div>
              <div class="col-md-6 mb-3">
                <label asp-for="Product.Ціна" class="fw-bold"></label>
                <input asp-for="Product.Ціна" class="form-control border border-primary bg-light" />
                <span asp-validation-for="Product.Ціна" class="text-danger"></span>
              </div>
              <div class="col-md-6 mb-3">
                <label asp-for="Product.Бренд" class="fw-bold"></label>
                <input asp-for="Product.Бренд" class="form-control border border-primary bg-light" />
                <span asp-validation-for="Product.Бренд" class="text-danger"></span>
              </div>
              <div class="col-md-12 mb-3">
                <label class="fw-bold">Доступні розміри</label>
                <div class="d-flex flex-wrap gap-3">
                  @foreach (var size in Model.SizeShoesList)
                  {
                    <div class="form-check">
                      <input class="form-check-input" type="checkbox" name="SelectedSizes" value="@size.Value"
                        @(Model.SelectedSizes.Contains(int.Parse(size.Value)) ? "checked" : "") />
                      <label class="form-check-label">@size.Text</label>
                    </div>
                  }
                </div>
              </div>
              <div class="col-md-12 mb-3">
                <label asp-for="Product.Опис" class="fw-bold"></label>
                <textarea asp-for="Product.Опис" class="form-control border border-primary bg-light"></textarea>
              </div>
              <div class="col-md-6 mb-3">
                <label asp-for="Product.id_категорії" class="fw-bold"></label>
                <select asp-for="Product.id_категорії" asp-items="Model.CategoryList" class="form-select border border-
primary bg-light">
                  <option disabled selected>Оберіть категорію</option>
                </select>
                <span asp-validation-for="Product.id_категорії" class="text-danger"></span>
              </div>
              <div class="col-md-6 mb-3">
                <label asp-for="Product.id_статі" class="fw-bold"></label>
                <select asp-for="Product.id_статі" asp-items="Model.GenderList" class="form-select border border-
primary bg-light">
                  <option disabled selected>Оберіть статть</option>
                </select>
              </div>
            </div>
          </div>
        </div>
      </div>
    </form>
  </div>
</div>

```

```

        </select>
        <span asp-validation-for="Product.id_срати" class="text-danger"></span>
    </div>
    <div class="col-md-12 mb-3">
        <label asp-for="Product.ImageUrl" class="fw-bold"></label>
        <input type="file" name="file" class="form-control border-0 shadow" />
    </div>
    <div class="col-6 col-md-4">
        <button type="submit" class="btn btn-primary form-control">
            @(Model.Product.id_продукції != 0 ? "Змінити" : "Створити")
        </button>
    </div>
    <div class="col-6 col-md-4">
        <a asp-controller="Product" asp-action="Index" class="btn btn-outline-primary form-control">
            Повернутись назад
        </a>
    </div>
</div>
</div>
</div>
<div class="col-md-4">
    @if (!string.IsNullOrEmpty(Model.Product.ImageUrl))
    {
        <div class="p-3 border bg-light rounded shadow-sm text-center">
            <label class="fw-bold mb-2">Поточне зображення</label>
            
        </div>
    }
</div>
</div>
</form>
<div>
    <label class="text-white">By By</label>
</div>
</div>
</div>
@section Scripts {
    <script>
        tinymce.init({
            selector: 'textarea',
            plugins: [

                ],
            toolbar: 'undo redo | blocks fontfamily fontsize | bold italic underline strikethrough | link image media table mergetags |
addcomment showcomments | spellcheckdialog a11ycheck typography | align lineheight | checklist numlist bullist indent outdent |
emoticons charmap | removeformat',
            tinymcecomments_mode: 'embedded',
            tinymcecomments_author: 'Author name',
            mergetags_list: [
                { value: 'First.Name', title: 'First Name' },
                { value: 'Email', title: 'Email' },
            ],
            ai_request: (request, respondWith) => respondWith.string(() => Promise.reject('See docs to implement AI Assistant')),
        });
    </script>
    @ {
        <partial name="_ValidationScriptsPartial" />
    }
}

```

SizeShoesIndex:

```

@model List<SizeShoes>
<partial name="_Notification"/>
<div class="card shadow border-0 mt-4">
  <div class="card-header bg-secondary bg-gradient ml-0 py-3">
    <div class="row">
      <div class="col-12 text-center">
        <h2 class="text-white py-2">Список доступних розмірів</h2>
      </div>
    </div>
  </div>
  <div class="row">
    <div class="col-6"></div>
    <div class="col-6 text-end py-4">
      <a asp-controller="SizeShoes" asp-action="Create" class="btn btn-primary">
        <i class="bi bi-plus-circle"></i> Додати новий розмір
      </a>
    </div>
  </div>

  <table class="table table-bordered table-striped">
    <thead>
      <tr>
        <th>
          Доступні розміри
        </th>
      </tr>
    </thead>
    <tbody>
      @foreach (var obj in Model.OrderBy(u => u.id_розміру))
      {
        <tr>
          <td>@obj.Розмір</td>
          <td>
            <div class="w-75 btn-group" role="group">
              <a asp-controller="SizeShoes" asp-action="Edit" asp-route-id="@obj.id_розміру" class="btn btn-primary
mx-2">
                <i class="bi bi-pencil"></i> Змінити
              </a>
              <a asp-controller="SizeShoes" asp-action="Delete" asp-route-id="@obj.id_розміру" class="btn btn-danger
mx-2">
                <i class="bi bi-trash3"></i> Видалити
              </a>
            </div>
          </td>
        </tr>
      }
    </tbody>
  </table>
</div>

```

SizeShoesEdit:

```

@model SizeShoes

<div class="card shadow border-0 mt-4">
  <div class="card-header bg-secondary bg-gradient py-3">
    <div class="row">
      <div class="col-12 text-center">
        <h2 class="text-white py-2">Видалити розмір</h2>
      </div>
    </div>
  </div>

```

```

    </div>
  </div>
</div>
<div class="card-body p-4">
  <form method="post">
    <div class="text-danger" asp-validation-summary="ModelOnly"></div>

    <div class="mb-3 row p-1">
      <label asp-for="Розмір" class="p-0 fw-bold"></label>
      <input asp-for="Розмір" class="form-control border border-primary bg-light" />
      <span asp-validation-for="Розмір" class="text-danger"></span>
    </div>

    <div class="row mt-3">
      <div class="col-6 col-md-4">
        <button type="submit" class="btn btn-primary form-control">Змінити</button>
      </div>
      <div class="col-6 col-md-4">
        <a asp-controller="SizeShoes" asp-action="Index" class="btn btn-outline-secondary form-control">
          Повернутись назад
        </a>
      </div>
    </div>
  </form>
</div>
</div>

@section Scripts {
  @{
    <partial name="_ValidationScriptsPartial" />
  }
}

```

SizeShoesCreate:

@model SizeShoes

```

<div class="card shadow border-0 mt-4">
  <div class="card-header bg-secondary bg-gradient ml-0 py-3">
    <div class="row">
      <div class="col-12 text-center">
        <h2 class="text-white py-2">Додати новий розмір</h2>
      </div>
    </div>
  </div>
  <div class="card-body p-4">
    <form method="post" class="row">
      <div class="border p-3">
        <div class="mb-3 row p-1">
          <label asp-for="Розмір" class="p-0 fw-bold"></label>
          <input asp-for="Розмір" class="form-control border border-primary bg-light" />
          <span asp-validation-for="Розмір" class="text-danger"></span>
        </div>
        <div class="row">
          <div class="col-6 col-md-4">
            <button type="submit" class="btn btn-primary form-control">Створити</button>
          </div>
          <div class="col-6 col-md-4">
            <a asp-controller="SizeShoes" asp-action="Index" class="btn btn-outline-primary form-control">
              Повернутись назад
            </a>
          </div>
        </div>
      </div>
    </form>
  </div>
</div>

```

```
</div>
```

```
@section Scripts{
  @{
    <partial name="_ValidationScriptsPartial" />
  }
}
```

UserIndex:

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

```
@{
  ViewData["Title"] = "Список користувачів";
}
```

```
<h2 class="mb-4">Список користувачів</h2>
<form method="get" class="row mb-4">
  <div class="col-md-6">
    <input type="text" name="searchEmail" class="form-control" placeholder="Введіть Email для пошуку"
      value="@Context.Request.Query["searchEmail"]" />
  </div>
  <div class="col-md-3">
    <button type="submit" class="btn btn-primary w-100">Пошук</button>
  </div>
  <div class="col-md-3">
    <a asp-action="Index" class="btn btn-outline-secondary w-100">Очистити</a>
  </div>
</form>
```

```
<table class="table table-bordered table-striped">
  <thead class="table-dark">
    <tr>
      <th>Email</th>
      <th>Ім'я користувача (UserName)</th>
      <th>Дія</th>
    </tr>
  </thead>
  <tbody>
    @foreach (var user in Model)
    {
      <tr>
        <td>@user.Email</td>
        <td>@user.UserName</td>
        <td>
          <a asp-action="ManageRole" asp-route-userId="@user.Id" class="btn btn-sm btn-primary">
            Змінити роль
          </a>
        </td>
      </tr>
    }
  </tbody>
</table>
```

ManageRoleViews:

```
@model ManageRoleVM
```

```
<h4>Зміна ролі для @Model.Email</h4>
```

```
<form asp-action="ManageRole" method="post">
  <input type="hidden" asp-for="UserId" />
  <div class="mb-3">
    <label class="form-label">Поточна роль</label>
    <input type="text" class="form-control" value="@Model.CurrentRole" readonly />
  </div>
```

```

<div class="mb-3">
  <label class="form-label">Нова роль</label>
  <select asp-for="SelectedRole" asp-items="@((new SelectList(Model.AllRoles))" class="form-select"></select>
</div>
<button type="submit" class="btn btn-success">Оновити</button>
</form>

```

CartController:

```

using Microsoft.AspNetCore.Authorization;
using Microsoft.AspNetCore.Mvc;
using System.Security.Claims;
using Vzutik.DataAccess.Repository.IRepository;
using Vzutik.Models;
using Vzutik.Models.ViewModels;
using Vzutik.Utility;

namespace Vzutik.Areas.Customer.Controllers
{
    [Authorize]
    [Area("Customer")]
    public class CartController : Controller
    {
        private readonly IUnitOfWork _unitOfWork;
        [BindProperty]
        public ShoppingCartVM ShoppingCartVM { get; set; }
        public CartController(IUnitOfWork unitOfWork)
        {
            _unitOfWork = unitOfWork;
        }
        public IActionResult Index()
        {
            var claimsIdentity = (ClaimsIdentity)User.Identity;
            var userId = claimsIdentity.FindFirst(ClaimTypes.NameIdentifier).Value;

            ShoppingCartVM = new()
            {
                ShoppingCartList = _unitOfWork.ShoppingCart.GetAll(
                    u => u.id_користувача == userId,
                    includeProperties: "Продукція"),
                OrderHeader = new OrderHeader()
            };

            foreach (var cart in ShoppingCartVM.ShoppingCartList)
            {
                double price = GetSum(cart);
                ShoppingCartVM.OrderHeader.Сума_замовлення += (price * cart.Кількість);
            }

            return View(ShoppingCartVM);
        }
        public IActionResult Summary()
        {
            var claimsIdentity = (ClaimsIdentity)User.Identity;
            var userId = claimsIdentity.FindFirst(ClaimTypes.NameIdentifier).Value;

            ShoppingCartVM = new()
            {
                ShoppingCartList = _unitOfWork.ShoppingCart.GetAll(
                    u => u.id_користувача == userId,
                    includeProperties: "Продукція"),
                OrderHeader = new OrderHeader()
            };

            var userFromDb = _unitOfWork.ApplicationUser.Get(u => u.Id == userId);

```

```

if (userFromDb == null)
{
    return RedirectToAction("Login", "Account");
}

ShoppingCartVM.OrderHeader.ПІБ = userFromDb.Назва;
ShoppingCartVM.OrderHeader.Номер_телефону = userFromDb.Номер_телефону;
ShoppingCartVM.OrderHeader.Місто = userFromDb.Місто;
ShoppingCartVM.OrderHeader.id_користувача = userId;

foreach (var cart in ShoppingCartVM.ShoppingCartList)
{
    double price = GetSum(cart);
    ShoppingCartVM.OrderHeader.Сума_замовлення += (price * cart.Кількість);
}

return View(ShoppingCartVM);
}

[HttpPost]
[ActionName("Summary")]
public IActionResult SummaryPOST()
{
    var claimsIdentity = (ClaimsIdentity)User.Identity;
    var userId = claimsIdentity.FindFirst(ClaimTypes.NameIdentifier).Value;

    ShoppingCartVM.ShoppingCartList = _unitOfWork.ShoppingCart.GetAll(u => u.id_користувача ==
userId,
    includeProperties: "Продукція");

    ShoppingCartVM.OrderHeader.Дата_замовлення = DateTime.Now;
    ShoppingCartVM.OrderHeader.id_користувача = userId;

    ApplicationUser applicationUser = _unitOfWork.ApplicationUser.Get(u => u.Id == userId);
    ShoppingCartVM.OrderHeader.Місто = applicationUser.Місто;

    foreach (var cart in ShoppingCartVM.ShoppingCartList)
    {
        double price = GetSum(cart);
        ShoppingCartVM.OrderHeader.Сума_замовлення += (price * cart.Кількість);
    }

    if(applicationUser.id_компанії.GetValueOrDefault() == 0)
    {
        ShoppingCartVM.OrderHeader.Статус_оплати = SD.PaymentStatusPending;
        ShoppingCartVM.OrderHeader.Статус_замовлення = SD.StatusPending;
    }
    else
    {
        ShoppingCartVM.OrderHeader.Статус_оплати = SD.PaymentStatusDelayedPayment;
        ShoppingCartVM.OrderHeader.Статус_замовлення = SD.StatusApproved;
    }
    _unitOfWork.OrderHeader.Add(ShoppingCartVM.OrderHeader);
    _unitOfWork.Save();
    foreach (var cart in ShoppingCartVM.ShoppingCartList)
    {
        OrderDetail orderDetail = new()

```

```

    {
        id_продукції = cart.id_продукції,
        id_замовлення = ShoppingCartVM.OrderHeader.id_замовлення,
        Ціна = cart.Продукція.Ціна,
        Кількість = cart.Кількість,
    };
    _unitOfWork.OrderDetail.Add(orderDetail);
    _unitOfWork.Save();
}

_unitOfWork.ShoppingCart.RemoveRange(ShoppingCartVM.ShoppingCartList);
_unitOfWork.Save();

if (applicationUser.id_компанії.GetValueOrDefault() == 0)
{
}

return RedirectToAction(nameof(OrderConfirmation), new
{id=ShoppingCartVM.OrderHeader.id_замовлення});
}

public IActionResult OrderConfirmation(int id)
{
    return View(id);
}

public IActionResult Plus(int cartId)
{
    var cartFromDb = _unitOfWork.ShoppingCart.Get(u => u.id_кошика == cartId);
    cartFromDb.Кількість += 1;
    _unitOfWork.ShoppingCart.Update(cartFromDb);
    _unitOfWork.Save();
    return RedirectToAction(nameof(Index));
}

public IActionResult Minus(int cartId)
{
    var cartFromDb = _unitOfWork.ShoppingCart.Get(u => u.id_кошика == cartId);
    if (cartFromDb.Кількість <= 1)
    {
        _unitOfWork.ShoppingCart.Remove(cartFromDb);
    }
    else
    {
        cartFromDb.Кількість -= 1;
        _unitOfWork.ShoppingCart.Update(cartFromDb);
    }

    _unitOfWork.Save();
    return RedirectToAction(nameof(Index));
}

public IActionResult Remove(int cartId)
{
    var cartFromDb = _unitOfWork.ShoppingCart.Get(u => u.id_кошика == cartId);
    _unitOfWork.ShoppingCart.Remove(cartFromDb);
    _unitOfWork.Save();
    return RedirectToAction(nameof(Index));
}

private double GetSum(ShoppingCart shoppingCart)

```



```

        {
            Text = g.Стать,
            Value = g.id_статі.ToString()
        })
    };

    return View(vm);
}

public IActionResult Details(ShopingCart shoppingCart, int productId)
{
    ShopingCart cart = new()
    {
        Продукція = _unitOfWork.Product.Get(u => u.id_продукції == productId,includeProperties:
"Category,ProductSizes.SizeShoes"),
        Кількість = 1,
        id_продукції = productId
    };

    return View(cart);
}
[HttpPost]
[Authorize]
public IActionResult Details(ShopingCart shoppingCart)
{
    var claimsIdentity = (ClaimsIdentity)User.Identity;
    var userId = claimsIdentity.FindFirst(ClaimTypes.NameIdentifier).Value;
    shoppingCart.id_користувача = userId;

    var cartFromDb = _unitOfWork.ShopingCart.Get(
        u => u.id_користувача == userId && u.id_продукції == shoppingCart.id_продукції
    );

    if (cartFromDb == null)
    {
        _unitOfWork.ShopingCart.Add(shopingCart);
    }
    else
    {
        cartFromDb.Кількість += shoppingCart.Кількість;

        cartFromDb.Обраний_розмір = shoppingCart.Обраний_розмір;

        _unitOfWork.ShopingCart.Update(cartFromDb);
    }

    _unitOfWork.Save();
    TempData["success"] = "Товар успішно додано до кошика!";
    return RedirectToAction(nameof(Index));
}

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

```

```

[ResponseCache(Duration = 0, Location = ResponseCacheLocation.None, NoStore = true)]
public IActionResult Error()
{
    return View(new ErrorViewModel { RequestId = Activity.Current?.Id ?? HttpContext.TraceIdentifier });
}
}
}

```

UserOrderController:

```

using Microsoft.AspNetCore.Authorization;
using Microsoft.AspNetCore.Mvc;
using System.Security.Claims;
using Vzutik.DataAccess.Repository.IRepository;
using Vzutik.Models;
using Vzutik.Models.ViewModels;

namespace Vzutik.Areas.Customer.Controllers
{
    [Area("Customer")]
    [Authorize]
    public class UserOrderController : Controller
    {
        private readonly IUnitOfWork _unitOfWork;

        public UserOrderController(IUnitOfWork unitOfWork)
        {
            _unitOfWork = unitOfWork;
        }

        public IActionResult Index()
        {
            var userId = User.FindFirstValue(ClaimTypes.NameIdentifier);

            var headers = _unitOfWork.OrderHeader
                .GetAll(o => o.id_користувача == userId, includeProperties: "Користувач")
                .OrderByDescending(o => o.Дата_замовлення)
                .ToList();

            var detailsByOrder = new Dictionary<int, IEnumerable<OrderDetail>>();

            foreach (var header in headers)
            {
                var details = _unitOfWork.OrderDetail
                    .GetAll(d => d.id_замовлення == header.id_замовлення, includeProperties: "Продукція");

                detailsByOrder[header.id_замовлення] = details;
            }

            var viewModel = new UserOrderListVM
            {
                OrderHeaders = headers,
                OrderDetailsByOrderId = detailsByOrder
            };

            return View(viewModel);
        }

        public IActionResult Details(int id)
        {
            var userId = User.FindFirstValue(ClaimTypes.NameIdentifier);

            var orderHeader = _unitOfWork.OrderHeader
                .Get(o => o.id_замовлення == id && o.id_користувача == userId, includeProperties: "Користувач");

```

```

var orderDetails = _unitOfWork.OrderDetail
    .GetAll(d => d.id_замовлення == id, includeProperties: "Продукція.ProductSizes.SizeShoes");

if (orderHeader == null)
    return NotFound();

OrderVM orderVM = new OrderVM
{
    OrderHeader = orderHeader,
    OrderDetails = orderDetails
};

return View(orderVM);
}
}
}

```

ViewsCartIndex:

@model ShopingCartVM

```

<form method="post">
    <div class="card shadow-lg border-0 mt-5 mb-5">

        <div class="card-header bg-secondary bg-gradient text-light py-4">
            <div class="px-4">
                <h5 class="mb-0 text-uppercase">Додано до кошика</h5>
            </div>
        </div>

        <div class="card-body px-4 py-5">
            <div class="mb-4">
                <a asp-area="Customer" asp-controller="Home" asp-action="Index"
                    class="btn btn-outline-primary text-uppercase btn-sm">
                    <small>Продовжити покупки</small>
                </a>
            </div>

            <div>
                @foreach (var item in Model.ShopingCartList)
                {
                    <div class="row border-bottom py-4 align-items-center">

                        <div class="d-none d-lg-block col-lg-1 text-center">
                            
                        </div>

                        <div class="col-12 col-lg-5">
                            <h5 class="text-uppercase text-secondary fw-bold mb-2">
                                @item.Продукція.Найменування_товару
                            </h5>

                            @{
                                var fullDescription = item.Продукція.Опис;
                                var words = fullDescription?.Split(' ');
                                var preview = words != null ? string.Join(" ", words.Take(5)) + (words.Length > 5 ? "...": "") : "";
                            }
                            <p class="text-muted mb-1">
                                <small>@preview</small>
                                <br />
                                <a href="javascript:void(0);"
                                    class="text-success fw-semibold"
                                    data-description="@item.Продукція.Опис"
                                    onclick="openDescriptionFromAttr(this)">

```

```

        детальніше
    </a>

</p>

@if (item.Обраний_розмір > 0)
{
    <span class="badge rounded-pill bg-primary text-white px-3 py-2">
        Розмір: @item.Обраний_розмір
    </span>
}
</div>

<div class="col-12 col-lg-6 mt-3 mt-lg-0">
    <div class="row align-items-center">
        <div class="col-4 col-md-3 text-center text-md-end">
            <h6 class="fw-semibold">
                @item.Продукція.Ціна.ToString("c")
            <span class="text-muted">&nbsp;x&nbsp;x&nbsp;x</span>@item.Кількість
            </h6>
        </div>

        <div class="col-6 col-md-5 text-center">
            <div class="btn-group" role="group">
                <a asp-action="Minus" asp-route-cartId="@item.id_кошика"
                    class="btn btn-outline-primary bg-gradient">
                    <i class="bi bi-dash-square"></i>
                </a>
                <a asp-action="Plus" asp-route-cartId="@item.id_кошика"
                    class="btn btn-outline-primary bg-gradient">
                    <i class="bi bi-plus-square"></i>
                </a>
            </div>
        </div>

        <div class="col-2 col-md-4 text-end">
            <a asp-action="Remove" asp-route-cartId="@item.id_кошика"
                class="btn btn-danger bg-gradient">
                <i class="bi bi-trash-fill"></i>
            </a>
        </div>
    </div>
</div>
</div>
</div>
</div>
}
</div>
</div>
<div class="card-footer bg-white border-0 px-4 pb-5">
    <div class="row mb-3">
        <div class="col-md-6 offset-md-3">
            <ul class="list-group">
                <li class="d-flex justify-content-between align-items-center">
                    <h5 class="text-dark fw-semibold text-uppercase mb-0">Сума замовлення:</h5>
                    <h4 class="text-dark fw-bolder mb-0">@Model.OrderHeader.Сума_замовлення.ToString("c")</h4>
                </li>
            </ul>
        </div>
    </div>
</div>
<div class="row">
    <div class="col-md-4 offset-md-4">
        @if (Model.ShopingCartList != null && Model.ShopingCartList.Any())
        {
            <a asp-action="Summary"
                class="btn btn-primary bg-gradient w-100 py-2 fw-semibold text-uppercase">
                Замовити
            </a>
        }
    </div>
</div>

```

```

    </a>
  }
  else
  {
    <button type="button"
      class="btn btn-secondary w-100 py-2 fw-semibold text-uppercase"
      disabled>
      Кошик порожній
    </button>
  }
</div>
</div>
</div>
</div>
</form>
<div class="modal fade" id="descriptionModal" tabindex="-1" aria-labelledby="descriptionModalLabel" aria-hidden="true">
  <div class="modal-dialog modal-dialog-centered modal-md">
    <div class="modal-content">
      <div class="modal-header">
        <h5 class="modal-title">Опис товару</h5>
        <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Закрити"></button>
      </div>
      <div class="modal-body">
        <p id="modalDescription" class="mb-0"></p>
      </div>
      <div class="modal-footer">
        <button type="button" class="btn btn-secondary" data-bs-dismiss="modal">Закрити</button>
      </div>
    </div>
  </div>
</div>
</div>
</div>
@section Scripts {
  <script>
    function openDescriptionFromAttr(element) {
      const description = element.getAttribute('data-description');
      const modalDescription = document.getElementById("modalDescription");
      modalDescription.textContent = description;
      const modal = new bootstrap.Modal(document.getElementById("descriptionModal"));
      modal.show();
    }
  </script>
}

```

OrderConfirmation:

```

@model int
<div class="container pt-5">
  <div class="row justify-content-center">
    <div class="col-md-8 text-center">
      <div class="card shadow border-0">
        <div class="card-header bg-success bg-gradient text-white">
          <h2 class="my-2">Замовлення успішно оформлене!</h2>
        </div>
        <div class="card-body">
          <p class="fs-5">Ваш номер замовлення:</p>
          <h4 class="text-primary fw-bold">@Model</h4>
          <hr />
          <a asp-area="Customer" asp-controller="Home" asp-action="Index" class="btn btn-outline-primary mt-3">
            Продовжити покупки
          </a>
        </div>
      </div>
    </div>
  </div>
</div>

```

```

</div>
</div>
</div>
</div>
</div>

```

Summary:

@model ShoppingCartVM

```

<form method="post">
  <br />
  <div class="container">
    <div class="card shadow border-0">
      <div class="card-header bg-secondary bg-gradient text-light ml-0 py-4">
        <div class="row px-4">
          <div class="col-6">
            <h5 class="pt-2 text-white">
              Деталі замовлення
            </h5>
          </div>
          <div class="col-6 text-end">
            <a asp-action="Index" class="btn btn-outline-danger btn-
sm">Повернутись до кошика</a>
          </div>
        </div>
      </div>
      <div class="card-body">
        <div class="container rounded p-2">
          <div class="row">
            <div class="col-12 col-lg-6 pb-4">
              <div class="row">
                <h4 class="d-flex justify-content-between align-items-
center mb-3">
                  <span class="text-info">Деталі
доставки:</span>
                </h4>
              </div>
              <div class="row my-1">
                <div class="col-3">
                  <label asp-for="OrderHeader.ПІБ"
class="form-label"></label>
                </div>
                <div class="col-9">
                  <input asp-for="OrderHeader.ПІБ"
class="form-control" />
                  <span asp-validation-
for="OrderHeader.ПІБ" class="text-danger"></span>
                </div>
              </div>
              <div class="row my-1">
                <div class="col-3">
                  <label asp-
for="OrderHeader.Номер_телефону" class="form-label"></label>
                </div>
                <div class="col-9">
                  <input asp-
for="OrderHeader.Номер_телефону" class="form-control" />
                  <span asp-validation-
for="OrderHeader.Номер_телефону" class="text-danger"></span>
                </div>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
</form>

```

```

class="form-control" />
for="OrderHeader.Micro" class="text-danger"></span>
</div>
</div>
<div class="row my-1">
  <div class="col-3">
    доставки</label>
  </div>
  <div class="col-9">
    <input asp-for="OrderHeader.Micro"
    <span asp-validation-
    </div>
  </div>
  <div class="row my-1">
    <div class="col-3">
      доставки</label>
    </div>
    <div class="col-9">
      <select class="form-select"
      id="Спосіб_доставки" name="Спосіб_доставки" onchange="toggleWarehouseField()"
      <option disabled selected>Оберіть
      <option value="Нова
      пошта">Нова Пошта</option>
      <option value="Укр
      пошта">Укрпошта</option>
      <option
      value="Самовивіз">Самовивіз</option>
      </select>
    </div>
  </div>
  <div class="row my-1" id="WarehouseRow" style="display:
  none;">
    <div class="col-3">
      <label asp-
      for="OrderHeader.Номер_відділення" class="form-label"></label>
    </div>
    <div class="col-9">
      <input asp-
      for="OrderHeader.Номер_відділення" class="form-control" />
      <span asp-validation-
      for="OrderHeader.Номер_відділення" class="text-danger"></span>
    </div>
  </div>
</div>
<div class="col-12 col-lg-5 offset-lg-1">
  <h4 class="d-flex justify-content-between align-items-center
  mb-3">
    <span class="text-info">Сума замовлення:</span>
  </h4>
  <ul class="list-group mb-3">
    @foreach(var details in Model.ShoppingCartList){
    <li class="list-group-item d-flex justify-content-
    between">
      <div>
        <h6 class="my-
        0">@details.Продукція.Найменування_товару</h6>
        <small class="text-
        muted">Кількість: @details.Кількість</small>
        <div>
          <span class="text-
          muted">@(((details.Продукція.Ціна*details.Кількість).ToString("c")) </span>
        </li>
      }

```

```

between bg-light">
<li class="list-group-item d-flex justify-content-
<small class="text-info">Ціна ГРН</small>
<strong class="text-
info">@Model.OrderHeader.Сума_замовлення.ToString("c")</strong>
</li>
</ul>
</div>
</div>
</div>
<div class="card-footer">
<div class="row">
<div class="col-12 col-md-8 pt-2">
<p style="color:maroon; font-size:14px;">
Орієнтовна дата прибуття:
@DateTime.Now.AddDays(1).ToShortDateString() -
@DateTime.Now.AddDays(5).ToShortDateString()
</p>
</div>
<div class="col-12 col-md-4">
<button type="submit" value="Place Order" class="btn btn-primary form-
control">Зробити замовлення</button>
</div>
</div>
</div>
</div>
</div>
</form>
@section Scripts {
<script>
function toggleWarehouseField() {
const delivery = document.getElementById("Список_доставки").value;
const warehouseRow = document.getElementById("WarehouseRow");

if (delivery === "Нова пошта" || delivery === "Укр пошта") {
warehouseRow.style.display = "flex";
} else {
warehouseRow.style.display = "none";
}
}
</script>
}

```

HomeViewsIndex:

```

@model Vzutik.Models.ViewModels.ProductListVM

@{
var selectedNoneCategory = Model.id_kategorii == null ? "selected" : "";
var selectedNoneGender = Model.id_статі == null ? "selected" : "";
}

<form method="get" class="row g-2 mb-3 align-items-end">
<div class="col-md-10">
<input type="text" name="searchTerm" class="form-control"
placeholder="Пошук за назвою..."
value="@Model.SearchTerm" />
</div>
<div class="col-md-2 d-grid">
<button type="submit" class="btn btn-outline-primary">Пошук</button>
</div>
</form>

```

```

<form method="get" class="row g-2 mb-4 align-items-end">
  <input type="hidden" name="searchTerm" value="" />

  <div class="col-md-3">
    <select class="form-select" name="id_категорії">
      <option value="">-- Всі категорії --</option>
      @foreach (var category in Model.CategoryList ?? Enumerable.Empty<SelectListItem>())
      {
        <text>
          @if (Model.id_категорії?.ToString() == category.Value)
          {
            <option value="@category.Value" selected>@category.Text</option>
          }
          else
          {
            <option value="@category.Value">@category.Text</option>
          }
        </text>
      }
    </select>
  </div>

  <div class="col-md-3">
    <select class="form-select" name="id_статі">
      <option value="">-- Для кого --</option>
      @foreach (var gender in Model.GenderList ?? Enumerable.Empty<SelectListItem>())
      {
        <text>
          @if (Model.id_статі?.ToString() == gender.Value)
          {
            <option value="@gender.Value" selected>@gender.Text</option>
          }
          else
          {
            <option value="@gender.Value">@gender.Text</option>
          }
        </text>
      }
    </select>
  </div>

  <div class="col-md-2 d-grid">
    <button type="submit" class="btn btn-outline-secondary">Фільтрувати</button>
  </div>
</form>

<div class="row g-4">
  @foreach (var product in Model.Products)
  {
    <div class="col-lg-3 col-md-4 col-sm-6">
      <div class="card h-100 border-0 shadow-sm">
        

        <div class="card-body d-flex flex-column">
          <h5 class="card-title text-center text-uppercase text-dark">
            @product.Найменування_товару
          </h5>
          <p class="card-text text-center text-warning">
            Бренд: <strong>@product.Бренд</strong>
          </p>
          <p class="card-text text-center text-dark">
            Ціна: ₪ <strong>@product.Ціна</strong>
          </p>
        </div>
      </div>
    </div>
  }

```

```

        <a asp-action="Details" asp-route-productId="@product.id_продукції" class="mt-auto btn btn-primary btn-
gradient border-0">
            Детальніше
        </a>
    </div>
</div>
</div>
}
</div>

@if (Model.TotalPages > 1)
{
    <nav aria-label="Page navigation" class="mt-5">
        <ul class="pagination justify-content-center">
            <li class="page-item @(Model.CurrentPage == 1 ? "disabled" : "")">
                <a class="page-link"
                    asp-action="Index"
                    asp-route-page="@ (Model.CurrentPage - 1)"
                    asp-route-searchTerm="@Model.SearchTerm"
                    asp-route-id_категорії="@Model.id_категорії"
                    asp-route-id_статі="@Model.id_статі">
                    Назад
                </a>
            </li>

            @for (int i = 1; i <= Model.TotalPages; i++)
            {
                <li class="page-item @(Model.CurrentPage == i ? "active" : "")">
                    <a class="page-link"
                        asp-action="Index"
                        asp-route-page="@i"
                        asp-route-searchTerm="@Model.SearchTerm"
                        asp-route-id_категорії="@Model.id_категорії"
                        asp-route-id_статі="@Model.id_статі">
                        @i
                    </a>
                </li>
            }

            <li class="page-item @(Model.CurrentPage == Model.TotalPages ? "disabled" : "")">
                <a class="page-link"
                    asp-action="Index"
                    asp-route-page="@ (Model.CurrentPage + 1)"
                    asp-route-searchTerm="@Model.SearchTerm"
                    asp-route-id_категорії="@Model.id_категорії"
                    asp-route-id_статі="@Model.id_статі">
                    Вперед
                </a>
            </li>
        </ul>
    </nav>
}

```

HomeDetails:

@model ShoppingCart

```

<form method="post">
    <input hidden asp-for="id_продукції" />

    <div class="card shadow-lg border-0 mt-5 mb-5">
        <div class="card-header bg-secondary bg-gradient text-light py-4">
            <div class="text-center">
                <h3 class="text-uppercase mb-1">@Model.Продукція.Найменування_товару</h3>
                <p class="text-white-50 fw-semibold mb-0">Бренд: @Model.Продукція.Бренд</p>
            </div>

```

```

</div>

<div class="card-body">
  <div class="row align-items-start">
    <div class="col-lg-5 text-center mb-4 mb-lg-0">
      
    </div>
    <div class="col-lg-7">
      <div class="mb-3">
        <span class="badge bg-dark-subtle text-dark px-3 py-2 fs-6">
          @Model.Продукція.Category.Назва
        </span>
      </div>
      <div class="mb-3">
        <h5 class="fw-semibold text-muted">
          Ціна:
          <span class="text-dark">@Model.Продукція.Ціна.ToString("c")</span>
        </h5>
      </div>
      <div class="mb-4">
        <h6 class="text-dark text-opacity-75 mb-2">Оберіть розмір:</h6>
        <div id="size-options" class="d-flex flex-wrap gap-2">
          @foreach (var size in Model.Продукція.ProductSizes)
          {
            <span class="badge size-option rounded-pill bg-light text-dark border border-primary px-3 py-2"
              data-size="@size.SizeShoes.Розмір"
              style="cursor:pointer;">
              @size.SizeShoes.Розмір
            </span>
          }
        </div>
        <input type="hidden" name="Обраний_розмір" id="selectedSizeInput" />
      </div>
      <div class="mb-4">
        <p class="text-secondary lh-base">@Html.Raw(Model.Продукція.Опис)</p>
      </div>
      <div class="row g-3 align-items-end">
        <div class="col-sm-6">
          <label class="form-label fw-semibold">Кількість</label>
          <div class="input-group">
            <span class="input-group-text bg-primary text-white border-0 fw-semibold">
              <i class="bi bi-plus-slash-minus"></i>
            </span>
            <input asp-for="Кількість" type="number" value="1" min="1"
              class="form-control text-end" />
          </div>
        </div>
        <div class="col-sm-6">
          <button id="addToCartBtn" type="submit"
            class="btn btn-primary bg-gradient w-100 py-2 text-uppercase fw-semibold"
            disabled>
            Додати до замовлення
          </button>
        </div>
      </div>
    </div>
  </div>
</form>

@section Scripts {
  <script>
    document.addEventListener("DOMContentLoaded", function () {
      const options = document.querySelectorAll(".size-option");
  
```

```

const hiddenInput = document.getElementById("selectedSizeInput");
const submitBtn = document.getElementById("addToCartBtn");

options.forEach(option => {
  option.addEventListener("click", function () {
    options.forEach(o => {
      o.classList.remove("bg-primary", "text-white");
      o.classList.add("bg-light", "text-dark");
    });

    this.classList.remove("bg-light", "text-dark");
    this.classList.add("bg-primary", "text-white");

    hiddenInput.value = this.dataset.size;

    submitBtn.disabled = false;
  });
});

if (!hiddenInput.value) {
  submitBtn.disabled = true;
}
});
</script>
}

```

HomePrivacy:

```

@{
  ViewData["Title"] = "Про компанію Взутік";
}
<h1>@ViewData["Title"]</h1>

```

<p>Ласкаво просимо до "Взутік" – вашого надійного партнера у світі якісного взуття!</p>

<p>Компанія "Взутік" займається продажем якісного та стильного взуття для всієї родини. Ми пропонуємо широкий асортимент моделей – від класичних та ділових варіантів до спортивного та повсякденного взуття.</p>

<h2>Наша місія</h2>

<p>Ми прагнемо зробити якісне та зручне взуття доступним для кожного. У нашому магазині ви знайдете товари від перевірених виробників, які поєднують у собі сучасний дизайн, надійність та комфорт.</p>

<h2>Чому обирають "Взутік"?</h2>

- Широкий вибір взуття для жінок, чоловіків та дітей;
- Тільки якісні матеріали та перевірені бренди;
- Зручний онлайн-магазин з швидкою доставкою;
- Регулярні знижки та акції для наших клієнтів;
- Індивідуальний підхід до кожного покупця.

<h2>Контакти</h2>

<p>Якщо у вас є запитання або пропозиції, зв'яжіться з нами:</p>

<p>Телефон: +380 (95) 222-11-11</p>

<p>Email: info@vzutyk.com</p>

<p>Адреса: м. Київ, вул. Прикладна, 12</p>

<p>Дякуємо, що обираєте "Взутік"! Ми працюємо для вашого комфорту.</p>

UserOrder:

```
@model Vzutik.Models.ViewModels.UserOrderListVM
```

```

<form method="post">
  <div class="card shadow-lg border-0 mt-5 mb-5">
    <div class="card-header bg-secondary bg-gradient text-light py-4">
      <div class="px-4">
        <h5 class="mb-0 text-uppercase">Мої замовлення</h5>
      </div>
    </div>

    <div class="card-body px-4 py-5">
      <div class="mb-4">
        <a asp-area="Customer" asp-controller="Home" asp-action="Index"
          class="btn btn-outline-primary text-uppercase btn-sm">
          <small>Продовжити покупки</small>
        </a>
      </div>

      <div>
        @foreach (var header in Model.OrderHeaders)
        {
          <div class="mb-4 p-3 border rounded shadow-sm">
            <h5>Замовлення №@header.id_замовлення — @header.Дата_замовлення.ToShortDateString()</h5>
            <p>Статус: @header.Статус_замовлення</p>
            <p>Сума: @header.Сума_замовлення</p>

            <hr />

            @if (Model.OrderDetailsByOrderId.ContainsKey(header.id_замовлення))
            {
              var details = Model.OrderDetailsByOrderId[header.id_замовлення];

              foreach (var item in details)
              {
                <div class="row border-bottom py-2 align-items-center">
                  <div class="col-lg-1">
                    
                  </div>
                  <div class="col-lg-7">
                    <strong>@item.Продукція.Найменування_товару</strong>
                    @ {
                      var words = item.Продукція.Опис?.Split(' ');
                      var preview = words != null ? string.Join(" ", words.Take(5)) + (words.Length > 5 ? "...": "") : "";
                    }
                    <p>
                      <span>@preview</span><br />
                      <button type="button"
                        class="btn btn-link p-0"
                        data-description="@item.Продукція.Опис"
                        onclick="openDescriptionFromAttr(this)">
                        детальніше
                      </button>
                    </p>
                  </div>
                  <div class="col-lg-2">
                    <span>Кількість: @item.Кількість</span>
                  </div>
                  <div class="col-lg-2">
                    <span>Ціна: @item.Ціна</span>
                  </div>
                  <div class="col-12 col-md-8 pt-2">
                    <p style="color:maroon; font-size:14px;">
                      Орієнтовна дата прибуття:
                      @header.Дата_замовлення.AddDays(1).ToShortDateString() -
                      @header.Дата_замовлення.AddDays(5).ToShortDateString()
                    </p>
                </div>
              }
            }
          </div>
        }
      </div>
    </div>
  </div>

```

```

    }
  }
}
else
{
  <p class="text-muted">Немає товарів у замовленні</p>
}
</div>
}

</div>
</div>
</div>
</form>
<div class="modal fade" id="descriptionModal" tabindex="-1" aria-labelledby="descriptionModalLabel" aria-hidden="true">
  <div class="modal-dialog modal-dialog-centered modal-md">
    <div class="modal-content">
      <div class="modal-header">
        <h5 class="modal-title">Опис товару</h5>
        <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Закрити"></button>
      </div>
      <div class="modal-body">
        <p id="modalDescription" class="mb-0"></p>
      </div>
      <div class="modal-footer">
        <button type="button" class="btn btn-secondary" data-bs-dismiss="modal">Закрити</button>
      </div>
    </div>
  </div>
</div>
</div>
</div>

@section Scripts {
  <script>
    function openDescriptionFromAttr(element) {
      const description = element.getAttribute('data-description');
      const modalDescription = document.getElementById("modalDescription");
      modalDescription.textContent = description;
      const modal = new bootstrap.Modal(document.getElementById("descriptionModal"));
      modal.show();
    }
  </script>
}
}

```