

Хмельницький національний університет
Факультет інформаційних технологій
Кафедра інженерії програмного забезпечення

КВАЛІФІКАЦІЙНА РОБОТА

Придолоби Сергія Віталійовича

Прізвище, ім'я, по батькові студента

На здобуття ступеня вищої освіти Бакалавра

«Вебзастосунок для продажу туристичного спорядження»

Назва теми

Галузь знань 12 «Інформаційні технології»

Спеціальність 121 «Інженерія програмного забезпечення»

Освітня програма Освітньо-професійна програма «Інженерія програмного забезпечення»

Шифр КвРІПЗ.2101085.01.14.ПЗ

Виконав студент IV курсу групи ПЗ-21-1



Підпис

Сергій ПРИДОЛОБА

Ініціали, прізвище

Керівник канд. техн. наук, доцент

Науковий ступінь, звання



Підпис

Оксана ЯШИНА

Ініціали, прізвище

Нормоконтролер канд. техн. наук, доцент



Підпис

Наталія ПРАВОРСЬКА

Ініціали, прізвище

До захисту допускаю:

Завідувач кафедри інженерії
програмного забезпечення



Підпис

Леонід БЕДРАТЮК

Ініціали, прізвище

10 червня 2025 р.

Хмельницький 2025

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет Інформаційних технологій

Кафедра Інженерії програмного забезпечення

Рівень вищої освіти Перший (бакалаврський)

Галузь знань 12 «Інформаційні технології»

Спеціальність 121 «Інженерія програмного забезпечення»

Освітня програма Освітньо-професійна програма «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри інз

Л. П. Бедратюк

02 01 2025 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Придолобі Сергію Віталійовичу

Прізвище, ім'я, по батькові студента

1. Тема роботи Вебзастосунок для продажу туристичного спорядження

Керівник роботи Яшина Оксана Миколаївна, канд. техн. наук, доцент

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 07.02.2025 р. №23

2. Строк подання студентом роботи на кафедру 01.06.25 р.

3. Вихідні дані до роботи Методичні матеріали до кваліфікаційної роботи

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

Дослідження предметної області та постановка задачі, проектування програмного забезпечення, програмна реалізація, тестування вебзастосунку

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень)

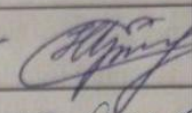
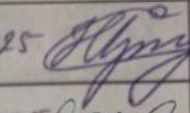
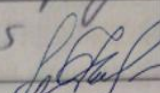
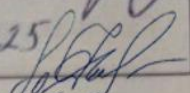
Три креслення:

1. Діаграма варіантів використання

2. Схеми бази даних

3. Архітектурне рішення

6. Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Праворська Н. І., доцент кафедри ІПЗ	5.06.25 	8.06.25 
Антиплагиат	Форкун Ю. В., доцент кафедри ІПЗ	8.06.25 	8.06.25 

7. Дата видачі завдання «02» січня 2024 р.

КАЛЕНДАРНИЙ ПЛАН

Назва етапів (розділів) кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1 Ознайомлення з тематикою кваліфікаційної роботи (КвР), визначення та узгодження індивідуальних тем КвР	01.12 – 31.12.2024	
2 Збір матеріалу за темою КвР; дослідження предметної області, в якій планується використання програмного забезпечення (ПЗ), визначення задач та вимог, розробка технічного завдання	02.01 – 31.01.2025	
3 Проектування програмного забезпечення	01.02 – 28.02.2025	
4 Програмна реалізація з використанням відповідних засобів розробки	01.03 – 10.04.2025	
5 Тестування програмного забезпечення	11.04 – 30.04.2025	
6 Написання вступу, загальних висновків, оформлення переліку джерел посилання та додатків. Оформлення пояснювальної записки КвР згідно вимог	01.05 – 25.05.2025	
7 Попередній захист КвР	травень 2025 (згідно графіка)	
8 Перевірка КвР на плагіат, нормоконтроль, отримання відгуків, рецензій та інших супровідних документів. Брошування (зшиття) пояснювальної записки.	26.05 – 30.05.2025	
9 Здача КвР на кафедрі; підготовка КвР для розміщення у репозиторії ХНУ; підготовка до захисту та захист КвР	з 01.06.2025	

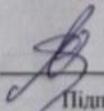
Студент


Підпис

Сергій ПРИДОЛОБА

Ініціали, прізвище

Керівник роботи


Підпис

Оксана ЯШИНА

Ініціали, прізвище

АНОТАЦІЯ

Тема кваліфікаційної роботи: «Вебзастосунок для продажу туристичного спорядження».

Автор роботи: Придолоба Сергій Віталійович.

Керівник роботи: Яшина Оксана Миколаївна.

Пояснювальна записка: 97 с., 17 рис., 5 табл., 2 дод., 32 джерела.

Графічна частина: 3 креслення у форматі А3.

ОНЛАЙН-РЕСУРС, ФРЕЙМВОРК, МОВА ПРОГРАМУВАННЯ, ОНЛАЙН-ПЛАТФОРМА, ЕЛЕКТРОННА КОМЕРЦІЯ, ТУРИЗМ, ТУРИСТИЧНЕ СПОРЯДЖЕННЯ, ПОКУПКА ОНЛАЙН, МЕРЕЖІ ІНТЕРНЕТ, ASP.NET MVC, MYSQL, HTML, CSS, C#.

Метою кваліфікаційної роботи є розробка вебзастосунку для продажу туристичного спорядження, що має можливість реалізувати функціонал різних категорії користувачів.

Під час проведення роботи було здійснено аналіз предметної області, а саме інтернет-торгівлі туристичним спорядженням та супутніми товарами, використовуючи відповідні джерела у всесвітній павутині. Також було визначено вимоги до розроблюваного вебзастосунку, визначено завдання, які необхідно вирішити для досягнення поставленої мети; спроектовано базу даних та обрано архітектуру.

У результаті отримано вебзастосунок для торгівлі туристичним спорядженням, на якому є можливість реєстрації, авторизації, вибору різноманітного роду товарів, їх покупка, замовлення, оплата та доставка.

1.06.25

Дата



Підпис

ВІДОМІСТЬ ДОКУМЕНТІВ

№ рядка	Формат	Позначення документа	Найменування документа	К-сть аркушів	№ екз.	Примітка
			<u>Текстові документи</u>			
1	A4	КвРІПЗ.2101085.01.14.П	Пояснювальна записка	85		
2	A4		Завдання на кваліфікаційну роботу	2		
3	A4		Анотація	1		
			<u>Графічні документи</u>			
4	A3	КвРІПЗ.2101085.01.14.Е8	Діаграма варіантів використання	1		
5	A3	КвРІПЗ.2101085.01.14.Е8	Схема Баз даних	1		
6	A3	КвРІПЗ.2101085.01.14.Е8	Архітектура	1		

КвРІПЗ.2101085.01.14.ПЗ				
Змн.	Арк.	№ докум.	Підпис	Дата
Виконав		Придолоба С.В.	<i>[Підпис]</i>	1.06
Керівник		Яшина О. М.	<i>[Підпис]</i>	1.06
Н. контр.		Праворська Н. І.	<i>[Підпис]</i>	1.06
Зав. каф.		Бедратюк Л.П.	<i>[Підпис]</i>	1.06
Вебзастосунок для торгівлі туристичним спорядженням				
Відомість документів				
		Літ.	Арк.	Аркуші
			1	1
ХНУ, ІПЗ-21-1				

ЗМІСТ

Вступ.....	7
1 Дослідження предметної області та постановка задачі.....	9
1.1 Змістовний аналіз предметної області, її структурних та функціональних особливостей.....	9
1.2 Аналіз наявного програмно-технічного забезпечення предметної області.....	11
1.3 Визначення вимог до програмного забезпечення та постановка задачі....	18
2 Проектування вебзастосунку.....	26
2.1 Проектування архітектури та структури вебзастосунку.....	25
2.2 Проектування логічної моделі бази даних вебзастосунку.....	27
2.3 Проектування інтерфейсу користувача вебзастосунку.....	34
2.4 Аналіз та вибір технологій і методів реалізації вебзастосунку.....	36
3 Програмна реалізація та тестування вебзастосунку.....	41
3.1 Реалізація бази даних.....	41
3.2 Реалізація модулів вебзастосунку.....	47
3.3 Інструкція користувача.....	54
3.4 Тестування вебзастосунку.....	56
Висновки.....	59
Перелік джерел посилання.....	61
Додаток А Презентаційні матеріали.....	64
Додаток Б Технічне завдання.....	82
Додаток В Лістинг коду.....	86

КвРІПЗ.2101085.01.14.ПЗ				
Змн.	Арк.	№ докум.	Підпис	Дата
Виконав		Придолоба С.В.		1.06
Керівник		Яшина О. М.		1.06
Н. контр.		Праворська Н. І.		1.06
Зав. каф.		Бедратюк Л.П.		1.06
Вебзастосунок для торгівлі туристичним спорядженням				
Відомість документів				
		Літ.	Арк.	Аркушів
			6	97
ХНУ, ІПЗ-21-1				

ВСТУП

Сучасне суспільство пройшло шлях від економіки товарів і послуг до економіки вражень. Через це цінність туристичного продукту сьогодні виходить далеко за межі товарів і послуг. Досвід є рушійною силою туристичної індустрії і робить туристичний продукт все більш і більш різноманітним. Більше того, для туристів стає надзвичайно важливим не лише сам досвід подорожі, але й досвід купівлі туристичного спорядження. У таких нових економічних умовах постачальникам туристичних товарів та послуг необхідно розуміти бажання, мотивацію та вимоги мандрівників, а також унікальні особливості туристичного продукту.

Крім того, важливо також зазначити, що в умовах воєнних дій на території України не всі мають можливість поїхати на відпочинок туристичним пакетом, а обирають самостійний так званий зелений туризм, де вдало підібране туристичне спорядження відіграє ключову роль. Також товари даного сегменту стають у пригоді нашим захисникам у ЗСУ.

Метою кваліфікаційної роботи є розробка вебзастосунку для продажу туристичного спорядження, що має можливість реалізувати функціонал різних категорії користувачів.

Процес розробки передбачає вирішення таких завдань:

- здійснити детальний аналіз предметної області, а саме онлайн-сервісів, платформ та загалом електронної комерції з продажу туристичного спорядження;
- проаналізувати існуючі на ринку програмно-технічні рішення для торгівлі товарів для туризму та супутніх товарів, туристичного спорядження, визначити ключові характеристики, недоліки та переваги цих ресурсів;
- проаналізувати та визначити вимоги до розроблюваного програмного продукту із виокремленням функціональних та нефункціональних вимог;

					КВРІПЗ.2101085.01.14.ПЗ	Арк.
						7
Зм.	Арк.	№ докум.	Підпис	Дата		

– на основі проведеного аналізу та визначення вимог до програмного забезпечення оформити необхідну документацію для розробки, а саме технічне завдання;

– здійснити аналіз типів архітектури та обрати відповідне архітектурне рішення для розроблюваного вебзастосунок;

– здійснити реалізацію з деталізованим описом програмного забезпечення, а саме вебзастосунок для торгівлі туристичним спорядженням;

– протестувати розроблений вебзастосунок для продажу туристичного спорядження.

					КвРІПЗ.2101085.01.14.ПЗ	Арк.
						8
Зм.	Арк	№ докум.	Підпис	Дата		

1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Змістовний аналіз предметної області, її структурних та функціональних особливостей

Застосунки, що використовуються та розробляються в сучасних реаліях стали важливими бізнес-інструментами. Незалежно від того, чи це невеликий інтернет-магазин, вебсайт, соціальна мережа, чи щось більш складне, наприклад, корпоративне програмне забезпечення, вебзастосунки дозволяють користувачам взаємодіяти з іншими користувачами та системами по всьому світу через Інтернет.

Попит на розробку вебзастосунків значно зріс за останні роки. Згідно з галузевими звітами Dataintel, світовий ринок вебдизайну та розробки у 2024 році оцінювався у 64,7 мільярда доларів США і, як очікується, зростатиме на 8%, щоб досягти 123,2 мільярда доларів США до 2032 року.

Крім того, можливості для розробки динамічних і зручних онлайн-додатків значно зросли з появою прогресивних онлайн-застосунків (PWA) та розвитком таких фреймворків, як React, Angular і Vue.js. Масштабовані рішення користуються великим попитом для підтримки бізнесу на шляху цифрової трансформації.

Вебзастосунки – це програмне забезпечення, доступ до якого здійснюється через веббраузер через HTTP та яке працює на віддаленому сервері. На відміну від традиційних настільних застосунків, вебзастосунки не потребують завантаження чи встановлення, що робить їх неймовірно зручними для користувачів. Ця доступність означає, що вебзастосунки можна використовувати на будь-якому пристрої з підключенням до Інтернету, що забезпечує гнучкість та простоту використання.

					КвРІПЗ.2101085.01.14.ПЗ	Арк.
						9
Зм.	Арк	№ докум.	Підпис	Дата		

Розробка вебзастосунків вимагає розуміння їхньої загальної клієнт-серверної архітектури. Зазвичай розміщені на хмарних платформах, розробка вебзастосунків пропонує покращену масштабованість і може легко адаптуватися до зростання, включаючи процеси сервера вебзастосунків. Масштабованість є особливо вигідною для підприємств, які прагнуть розширити свій охоплення та можливості без значних початкових інвестицій.

Такі відомі назви, як Google Docs, Gmail та Facebook, є популярними прикладами вебзастосунків. Ці застосунки демонструють різноманітні функції, які може запропонувати більшість вебзастосунків, від редагування документів до соціальних мереж. Незалежно від того, чи йдеться про оптимізацію операцій, підтримку цифрової трансформації чи створення нових продуктів, вебзастосунки підвищують залученість користувачів та сприяють успіху бізнесу.

Застосунок для веброзробки - це інтерактивна програма, яка працює на сервері і доступна через веббраузер. Ці застосунки необхідні для надання послуг користувачам і забезпечують цінний зворотній зв'язок для розробників. Цей зворотний зв'язок часто включає дані про використання, що дає уявлення про взаємодію користувачів, їхні вподобання та часто використовувані функції.

Вебзастосунки мають важливе значення, оскільки вони покращують користувацький досвід і надають компаніям важливі дані для формування їхніх продуктових стратегій. Компанії можуть налаштовувати свої пропозиції, аналізуючи поведінку користувачів, покращуючи продуктивність та узгоджуючи свої послуги з очікуваннями клієнтів.

Безперебійне функціонування вебзастосунків залежить від кількох основних технічних компонентів. Розуміння цих елементів має вирішальне значення для створення привабливих вебзастосунків і забезпечення безперебійної роботи користувачів під час взаємодії з програмним забезпеченням.

					КвРІПЗ.2101085.01.14.ПЗ	Арк.
						10
Зм.	Арк	№ докум.	Підпис	Дата		

1.2 Аналіз наявного програмно-технічного забезпечення предметної області

Для виявлення наявного програмного забезпечення було проаналізовано існуючі розробки та зроблено висновок про їх переваги та недоліки.

Як приклад можна навести OLX.ua (рисунок 1.1) - українська платформа оголошень, де можна купити та продати товари, включаючи туристичне спорядження та має простий інтерфейс і велику кількість користувачів.

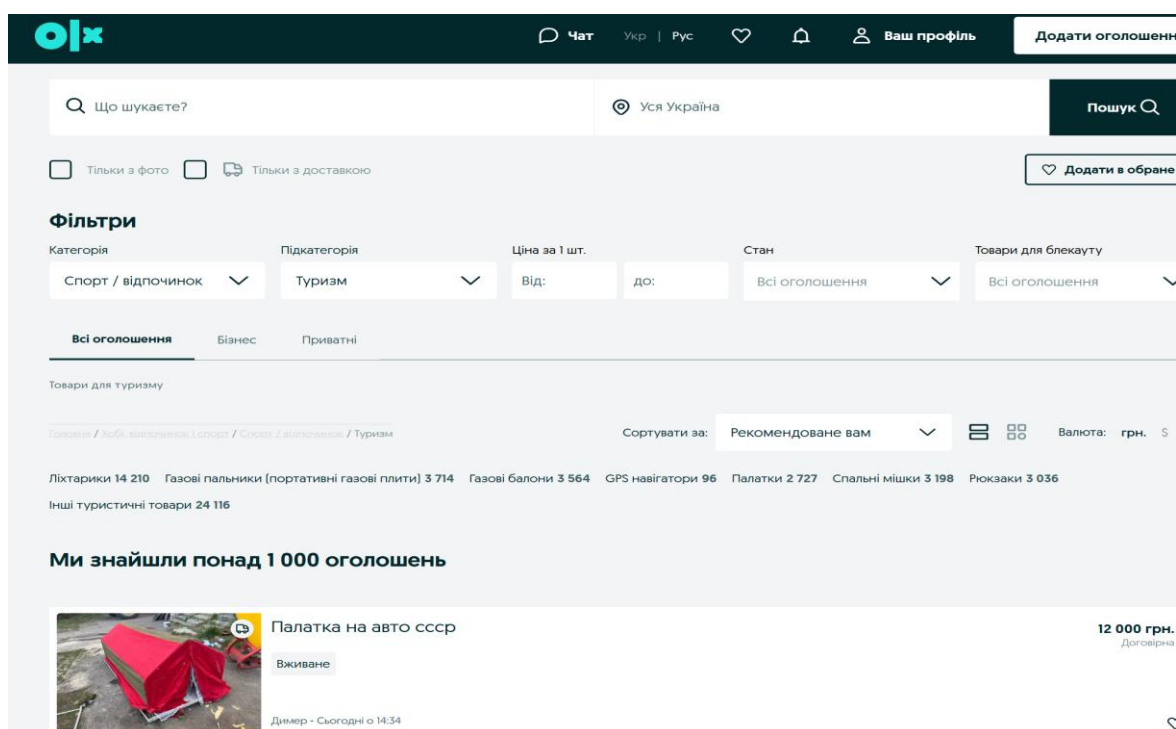


Рисунок 1.1 – Головна сторінка ресурсу OLX.ua

Переваги:

- велика аудиторія, що потенційно збільшує продажі;
- простота використання для продавців;
- можливість продажу вживаного спорядження.

Недоліки:

- відсутність спеціалізованих інструментів для продажу туристичного спорядження;

					КвРІПЗ.2101085.01.14.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		11

- необхідність самостійно керувати продажами та доставкою;
- досить високий ризик зіткнутися з шахрайством;
- мінімум можливостей для інтеграції, відсутність API для повноцінної роботи.

Prom.ua (рисунок 1.2) - український маркетплейс, що об'єднує безліч продавців, у тому числі туристичного спорядження та має інструменти для управління продажами та спілкування з клієнтами.

Переваги:

- велика аудиторія потенційних покупців;
- інструменти для управління продажами та замовленнями;
- можливість створення власного міні-сайту на платформі.

Недоліки:

- конкуренція з іншими продавцями;
- необхідність сплачувати комісію за продажі;
- обмежені можливості кастомізації, складнощі з інтеграцією сторонніх сервісів.

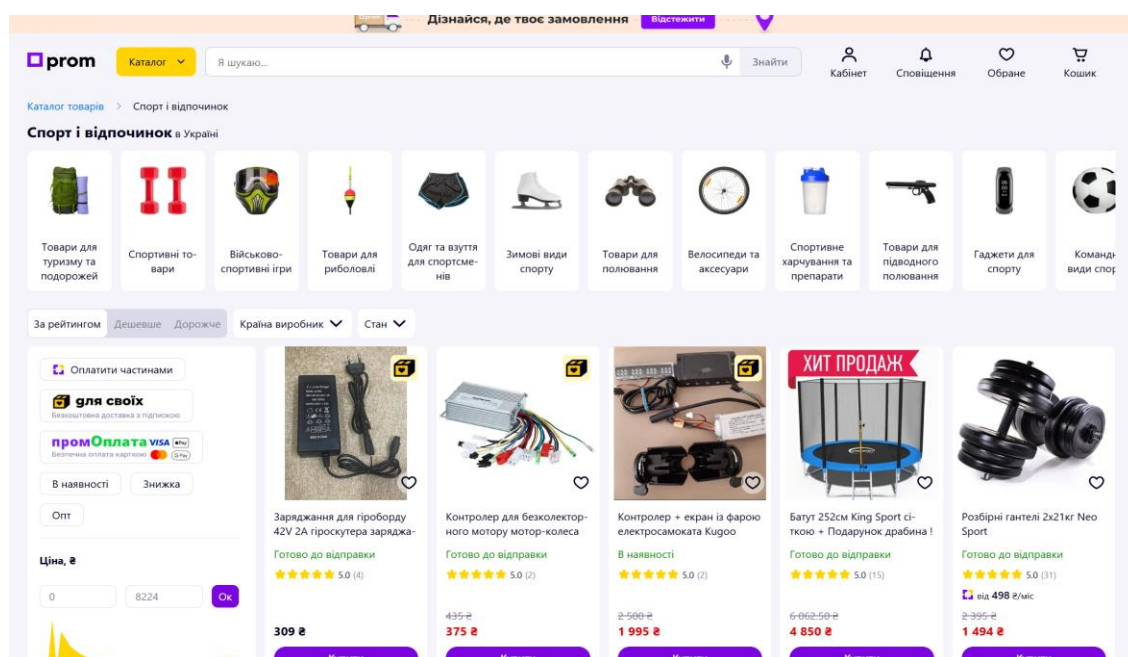


Рисунок 1.2 – Головна сторінка ресурсу Prom.ua

					КвРІПЗ.2101085.01.14.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		12

Еpicentrk.ua (рисунок 1.3) - великий гіпермаркет, де також є відділ туристичного спорядження, широкий асортимент товарів, можливість доставки.

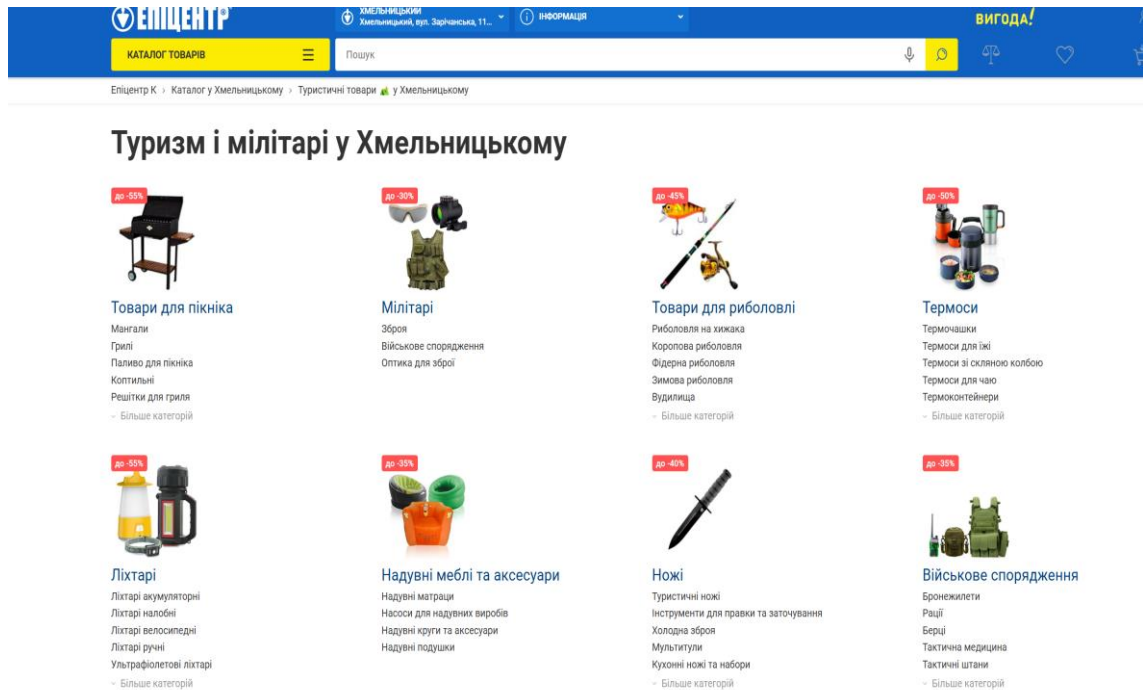


Рисунок 1.3 – Головна сторінка ресурсу EpiCentre.ua

Переваги:

- велика мережа гіпермаркетів, що викликає довіру;
- можливість оглянути товар у фізичному магазині;
- налагоджена система доставки.

Недоліки:

- інтернет-магазин не спеціалізується виключно на туристичному спорядженні;
- велика конкуренція серед інших товарів на сайті;
- складна структура сайту, обмежені можливості для інтеграції.

При розробці інтернет-магазину туристичного спорядження важливо врахувати:

- спеціальні фільтри для пошуку товарів;
- інтеграцію з платіжними системами та службами доставки;

					КвРІПЗ.2101085.01.14.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		13

- систему відгуків та рейтингів товарів;
- адаптивний дизайн для зручного використання на мобільних пристроях;
- можливість інтеграції з сервісами аналітики для відстеження поведінки користувачів.

Використання готових платформ, таких як Prom.ua, може спростити запуск магазину, але обмежує можливості кастомізації. Розробка власного сайту вимагає більше часу та ресурсів, але надає повний контроль над функціоналом та дизайном.

Rozetka.ua (рисунок 1.4) - великий український інтернет-магазин, що пропонує широкий асортимент товарів, включаючи туристичне спорядження, має зручний інтерфейс, систему фільтрів та відгуків.

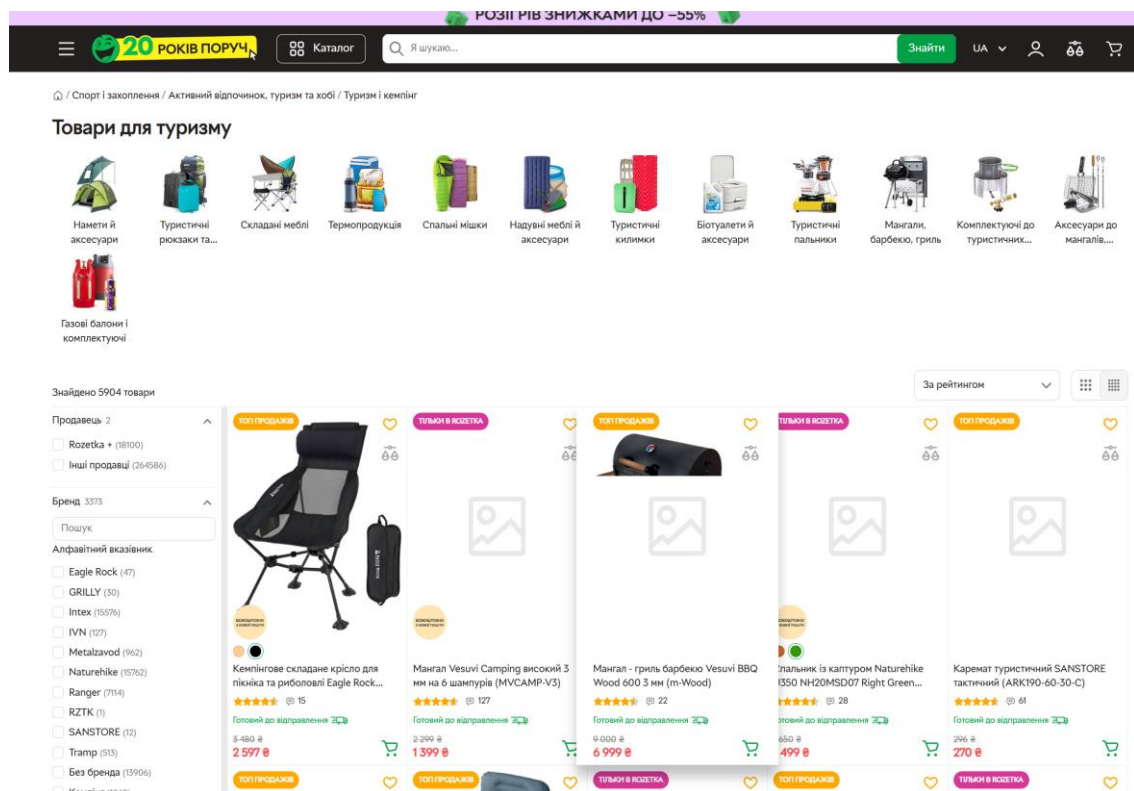


Рисунок 1.4 – Головна сторінка ресурсу Rozetka.ua

Переваги:

- широкий вибір товарів;
- конкурентні ціни;

									Арк.
Зм.	Арк	№ докум.	Підпис	Дата					14

КвРІПЗ.2101085.01.14.ПЗ

- зручна доставка;
- система відгуків покупців.

Недоліки:

- не завжди спеціалізується на туристичному спорядженні, тому може не мати вузькоспеціалізованих товарів;
- великий об'єм товарів може ускладнити пошук.

Gorgany.com (рисунок 1.5) - український інтернет-магазин туристичного спорядження, спеціалізується на товарах для активного відпочинку, походів, альпінізму та має фізичні магазини у великих містах.

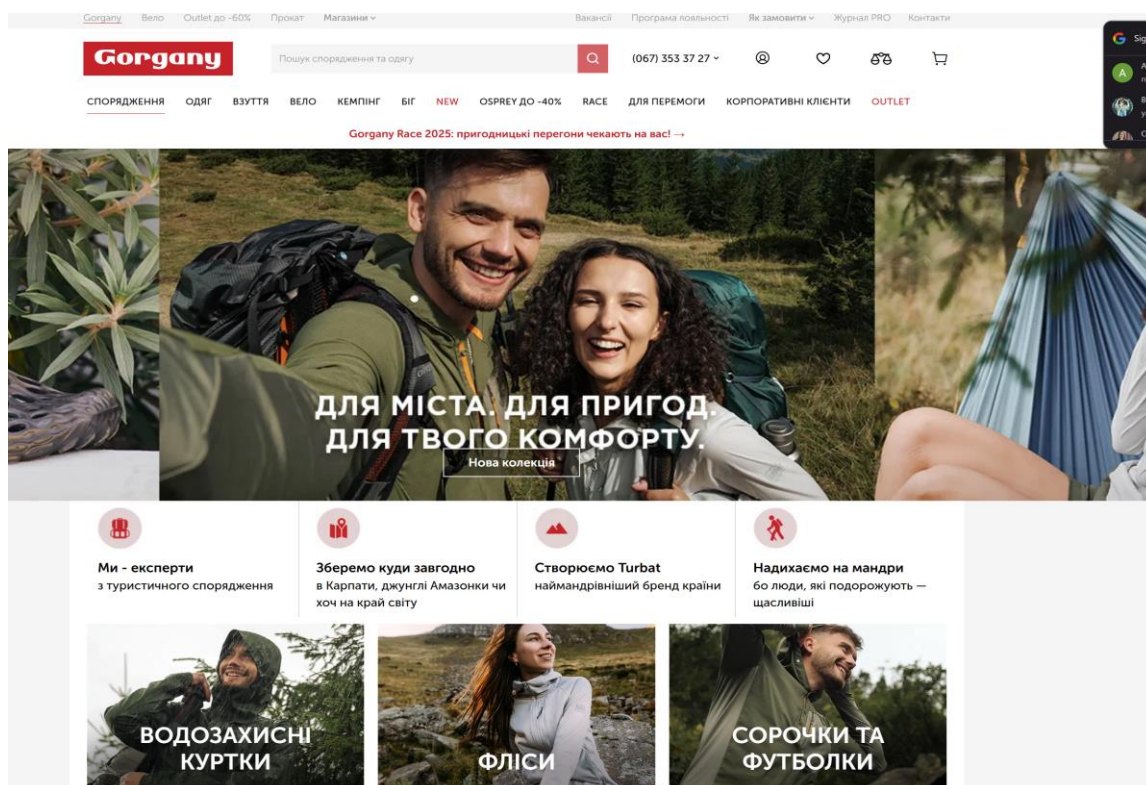


Рисунок 1.5 – Головна сторінка ресурсу Gorgany.com

Переваги:

- вузька спеціалізація на туристичному спорядженні;
- консультації фахівців;
- широкий вибір якісного спорядження;
- можливість відвідати фізичні магазини.

					КвРІПЗ.2101085.01.14.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		15

Недоліки:

- ціни можуть бути вищими, ніж у великих універсальних магазинах;
- менший вибір товарів, порівняно з великими маркетплейсами.

Alp.com.ua (рисунок 1.6) - інтернет-магазин альпіністського та туристичного спорядження, що пропонує широкий вибір товарів для екстремального туризму та має досвідчених консультантів.

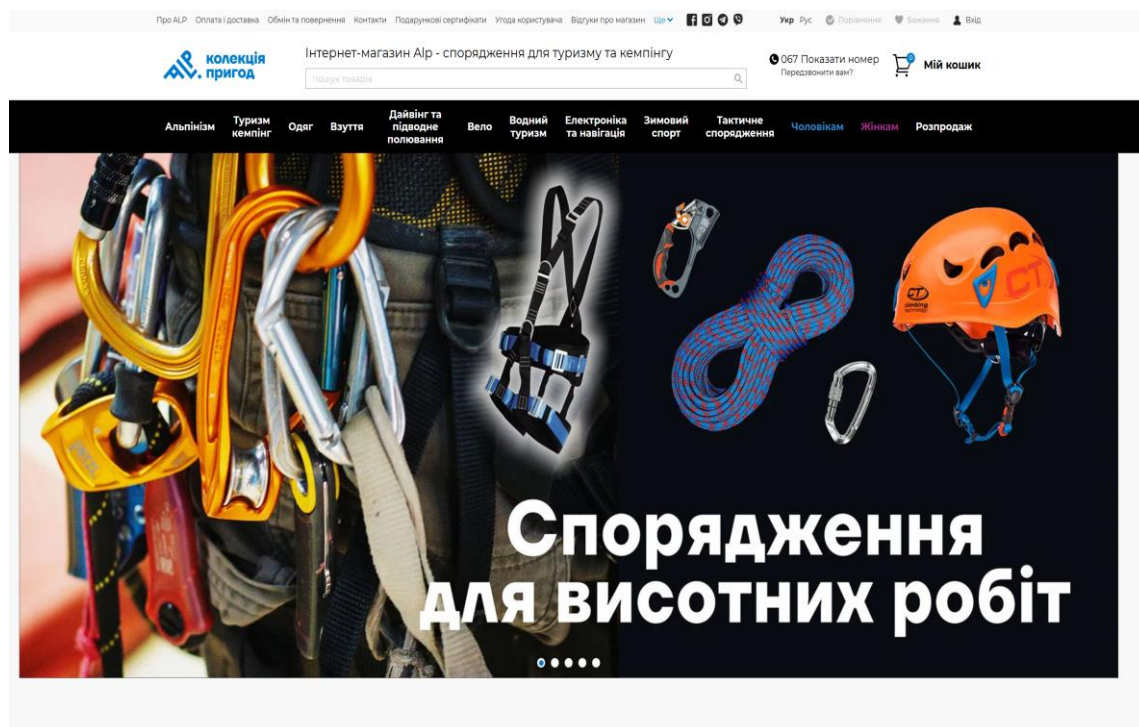


Рисунок 1.6 – Головна сторінка ресурсу Alp.com.ua

Переваги:

- спеціалізований асортимент для альпінізму та екстремального туризму;
- висока якість товарів;
- професійні консультації.

Недоліки:

- ціни можуть бути високими;
- менший вибір товарів для загального туризму.

					КвРІПЗ.2101085.01.14.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		16

Adventure.com.ua (рисунок 1.7) - інтернет магазин, що спеціалізується на продажі товарів для активного відпочинку, туризму та спорту, має широкий асортимент товарів від відомих брендів.

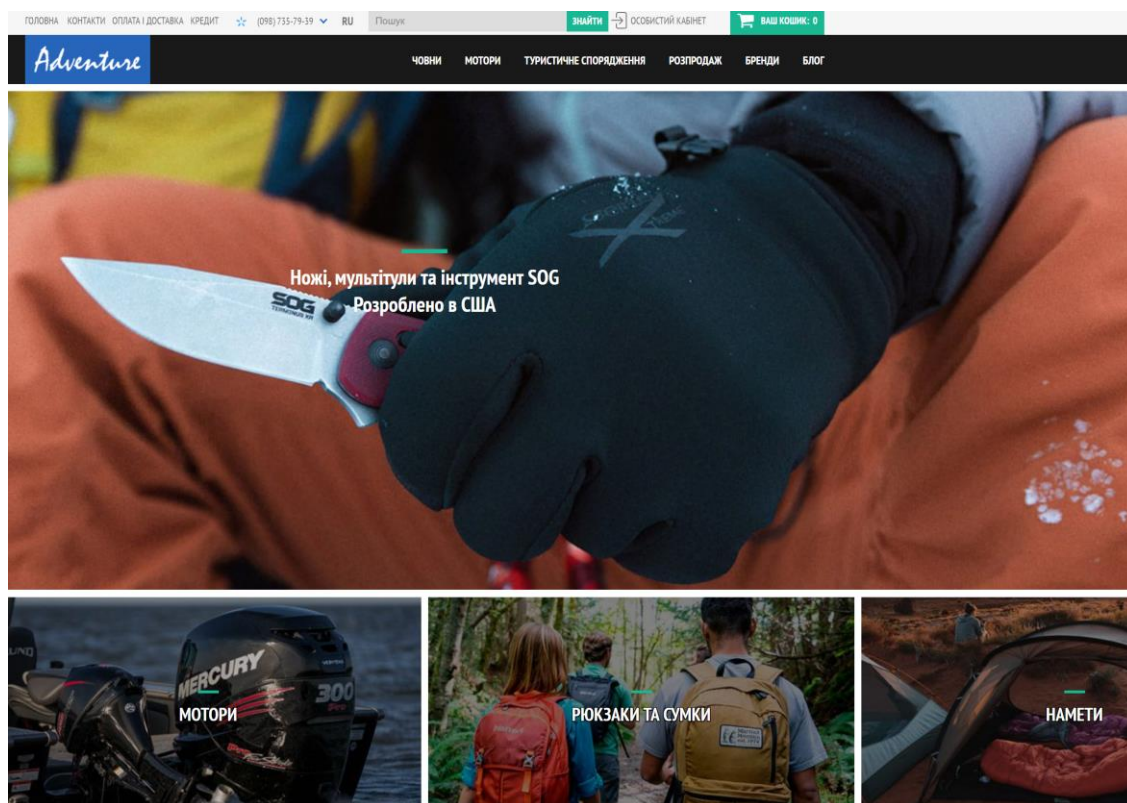


Рисунок 1.7 – Головна сторінка ресурсу Adventure.com.ua

Переваги:

- великий вибір туристичного та спортивного спорядження;
- якісні товари від відомих брендів;
- зручна навігація по сайту.

Недоліки:

- ціни можуть бути вищими за середні;
- не завжди є в наявності весь асортимент.

Отже, проаналізувавши існуючі рішення можна зробити висновки про те, що більшість вебсайтів мають зручну навігацію, фільтри для пошуку та систему відгуків, а також систему оплати та доставки, зручний та адаптивний дизайн.

					КвРІПЗ.2101085.01.14.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		17

1.3 Визначення вимог до програмного забезпечення та розробка технічного завдання

Під час аналітики вимог до розроблюваного програмного продукту і закладається вирішальна основа для успішного проекту. Тому тут потрібно буде заглибитися як у функціональні, так і в нефункціональні аспекти.

Функціональні вимоги – це такі вимоги, які описують, що саме робитиме застосунок. Тому необхідно визначити ті основні функції, які знадобляться користувачеві та адміністраторам. До таких функцій належать: управління каталогом продукції; перегляд продуктів, тобто користувачі повинні мати можливість переглядати каталог туристичного спорядження. Це включає:

- відображення назв продуктів, описів, зображень, цін та наявності;
- фільтрацію продуктів за категоріями (наприклад, намети, рюкзаки, взуття, одяг, навігація), підкатегоріями, брендами, ціновими діапазонами, характеристиками (наприклад, водонепроникність, легкість) та розмірами;
- сортування продуктів за ціною (від низької до високої, від високої до низької), назвою та популярністю;
- реалізацію функції пошуку для пошуку продуктів за ключовими словами.

Відомості про продукт повинні містити такий функціонал, щоб користувачі мали можливість переглядати детальну інформацію про кожен продукт, включаючи:

- кілька зображень;
- детальні описи, характеристики та матеріали;
- відгуки та оцінки клієнтів;
- інформація про політику доставки та повернення товару (можливо);
- супутні товари або пропозиції на основі історії переглядів або категорії.

Проектування вебзастосунку передбачає Адміністраторське управління товарами, тобто адміністратори можуть:

					КвРІПЗ.2101085.01.14.ПЗ	Арк.
						18
Зм.	Арк	№ докум.	Підпис	Дата		

- додавати нові товари з усіма необхідними деталями (назва, опис, зображення, ціна, артикул, категорія, підкатегорія, характеристики, наявність);
- редагувати наявну інформацію про товар;
- керувати запасами товарів (рівень запасів);
- класифікувати товари за категоріями та підкатегоріями;
- керувати зображеннями товарів;
- потенційно керувати знижками та акціями на товари.

Облікові записи користувачів та автентифікація передбачають:

- реєстрація;
- вхід/вихід;
- керування профілем;
- відновлення пароля.

Нові користувачі повинні мати можливість створювати облікові записи з необхідною інформацією (ім'я, електронна пошта, пароль, адреса, номер телефону). Має бути безпечна автентифікація для зареєстрованих користувачів. Користувачі повинні мати можливість переглядати та редагувати інформацію свого профілю, включаючи адреси та способи оплати (зберігаються безпечно, можливо, токенизовані). Функціональність для користувачів, що дозволяють скидати забуті паролі.

Кошик для покупок та оформлення замовлення має мати такий функціонал:

- додати до кошика, тобто користувачі повинні мати можливість додавати товари до свого кошика;
- перегляд кошика, тобто користувачі повинні мати можливість бачити товари у своєму кошику, включаючи кількість та ціни;
- оновлення кошика, тобто користувачі повинні мати можливість змінювати кількість або видаляти товари з кошика;

					КвРІПЗ.2101085.01.14.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		19

– розрахувати загальну суму, тобто застосунок повинен автоматично розраховувати проміжний підсумок, вартість доставки, податки (якщо застосовується) та остаточну суму.

Процес оформлення замовлення повинен бути чіткий та безпечний багатоетапний процес оформлення замовлення:

- вибір або введення адреси доставки;
- вибір способу доставки;
- вибір способу оплати та безпечна обробка платежів (інтеграція з платіжними шлюзами).
- підтвердження замовлення та зведення.

Керування замовленнями:

- розміщення замовлення, тобто користувачі повинні отримувати підтвердження свого замовлення з деталями та номером відстеження (якщо застосовується);
- історія замовлень, тобто зареєстровані користувачі повинні мати можливість переглядати історію своїх попередніх замовлень та деталі замовлення.

Керування замовленнями адміністратора, тобто адміністратори повинні мати можливість:

- переглядати та керувати всіма замовленнями;
- оновлювати статуси замовлень (наприклад, очікує на розгляд, обробка, відправлено, доставлено, скасовано);
- генерувати рахунки-фактури та транспортні етикетки;
- потенційно обробляти повернення та відшкодування.

Підтримка клієнтів:

- форма зворотного зв'язку, тобто спосіб для користувачів зв'язатися зі службою підтримки клієнтів із будь-якими запитами;
- розділ поширених запитань, тобто інформативний розділ, що відповідає на поширені запитання.

					КвРІПЗ.2101085.01.14.ПЗ	Арк.
						20
Зм.	Арк	№ докум.	Підпис	Дата		

Нефункціональні вимоги описують якості програми та те, наскільки добре вона виконує свої функції. До нефункціональних вимог відносять:

- продуктивність;
- чутливість;
- масштабованість;
- ефективність;
- безпека;
- захист даних;
- безпечні платежі.

Програма повинна швидко завантажувати сторінки та реагувати на дії користувача (наприклад, протягом кількох секунд), а також повинна мати можливість обробляти зростаючу кількість користувачів та продуктів без значного зниження продуктивності, ефективно використовувати ресурси (сервер, базу даних).

Має бути передбачено безпечне зберігання та обробка конфіденційних даних користувачів (персональна інформація, платіжні реквізити). Відповідність відповідним правилам конфіденційності даних (наприклад, GDPR, CCPA); інтеграція із безпечними платіжними шлюзами та дотримання стандартів PCI DSS; захист від поширених веб-вразливостей: (наприклад, SQL-ін'єкції, міжсайтовий скриптинг (XSS), CSRF); безпечні механізми автентифікації та авторизації.

Повинна бути передбачена зручність використання, а саме:

- інтуїтивно зрозуміла навігація, тобто проста та зрозуміла навігація по вебсайту;
- зручний інтерфейс, а саме чистий, послідовний та візуально привабливий дизайн;
- дотримання правил доступності (наприклад, WCAG) для забезпечення зручності використання людьми з інвалідністю;

					КвРІПЗ.2101085.01.14.ПЗ	Арк.
						21
Зм.	Арк	№ докум.	Підпис	Дата		

– мобільна адаптація передбачає, що застосунок має бути повністю функціональним та добре відображатися.

Діаграму варіантів використання згідно описаних вимог для користувача (клієнта) подано на рисунку 1.8.



Рисунок 1.8 – Діаграма варіантів використання для користувача (клієнта)

Окрім діаграми варіантів використання для користувача у ролі потенційного клієнта ще доцільно продемонструвати діаграму варіантів використання для адміністратора, що зображено на рисунку 1.9.

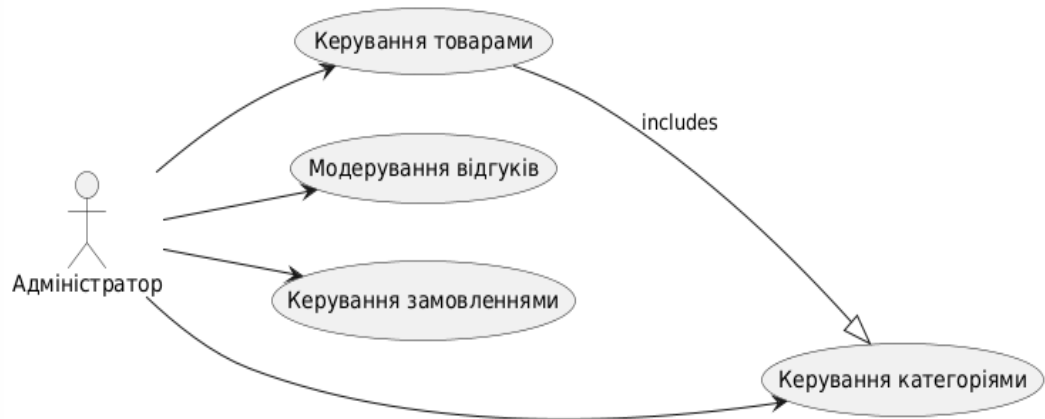


Рисунок 1.9 - Діаграма варіантів використання для адміністратора

Діаграму станів продемонстровано на рисунку 1.10.

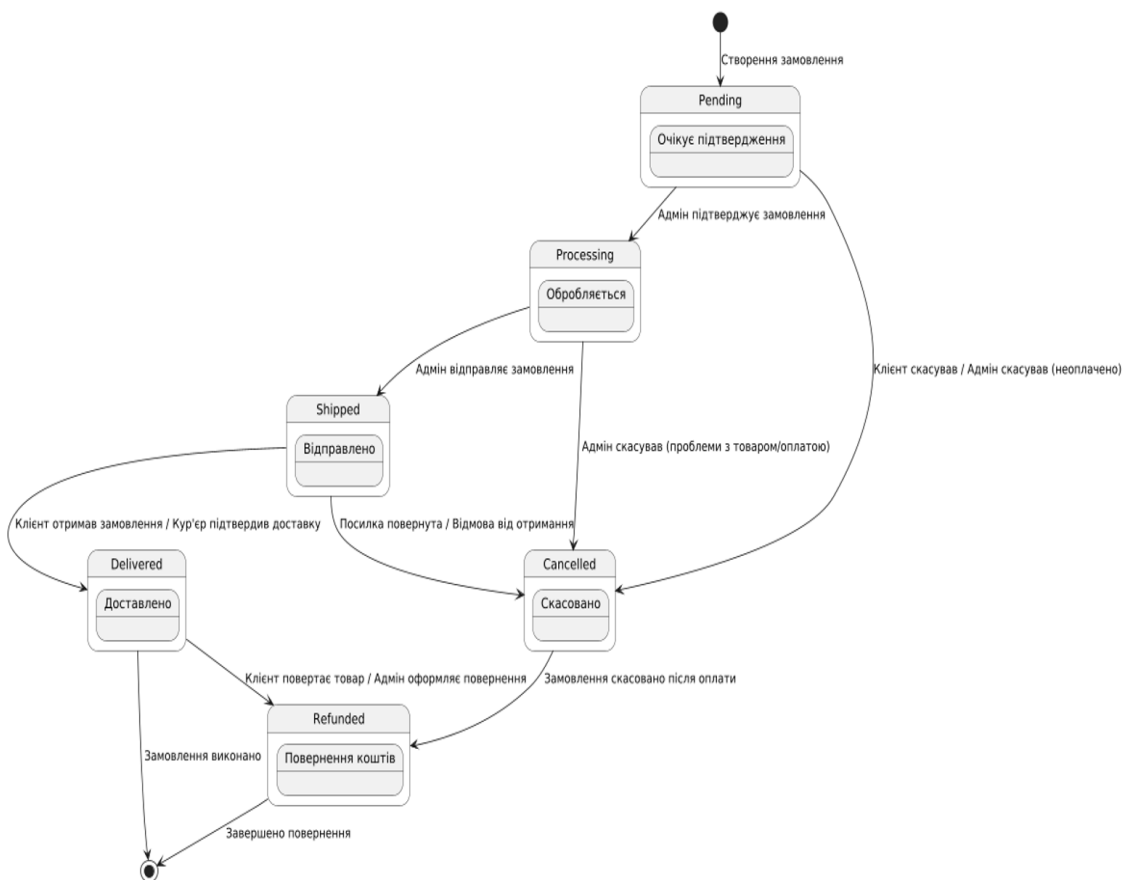


Рисунок 1.10 – Діаграма станів

Висновки до 1-го розділу

Отже, в першому розділі подано означення вебзастосунків та розглянуто їх актуальність та доцільність розробки, з детальним аналізом предметної області, а саме електронної комерції з продажу туристичного спорядження.

Також показано, які є програмно-технічні рішення на ринку сьогодні. Визначено функціональні та нефункціональні вимоги до розроблюваного вебзастосунку та продемонстровано діаграми варіантів використання для користувача в ролі покупця та адміністратора, а також діаграму станів.

					КвРІПЗ.2101085.01.14.ПЗ	Арк.
						24
Зм.	Арк	№ докум.	Підпис	Дата		

2 ПРОЕКТУВАННЯ ДОВІДКОВО-ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1 Проектування архітектури та структури системи

Архітектура вебзастосунків є основою будь-якого сучасного веб-продукту, визначаючи, як його компоненти взаємодіють між собою для забезпечення бездоганного користувацького досвіду. Вибір правильної архітектури має вирішальне значення, оскільки безпосередньо впливає на продуктивність, масштабованість, безпеку та зручність обслуговування вашого додатка. З огляду на різноманітність архітектурних моделей і технологій, знайти найкращий варіант для вашого конкретного продукту може бути складно.

При виборі архітектури необхідно враховувати такі аспекти, як складність застосунку, очікуваний трафік користувачів, майбутня масштабованість та вимоги до безпеки. Незалежно від того, чи створюється простий односторінковий додаток (SPA) чи складне багаторівневе корпоративне рішення, правильна архітектура забезпечує оптимальну функціональність та задоволення користувачів.

Архітектура вебзастосунків – це фундаментальна структура, яка формує дизайн та розробку вебзастосунку. Вона складається з процедур та найкращих практик, які визначають, як різні елементи додатку, такі як фронтенд, бекенд та база даних, працюють разом, щоб забезпечити необхідну функціональність.

Простіше кажучи, це як макет, який забезпечує безперебійну взаємодію між видимими елементами, з якими взаємодіють користувачі, логікою на стороні сервера та сховищем даних. Мета полягає в тому, щоб створити безперебійну, масштабовану систему, яка відповідає функціональним вимогам додатку, а також зосереджується на продуктивності, безпеці та простоті обслуговування.

Вебзастосунки відрізняються складністю та функціональністю, що безпосередньо впливає на кількість шарів та компонентів, що беруть участь в

					КвРІПЗ.2101085.01.14.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		25

їхній архітектурі. Деякі застосунки достатньо прості, щоб працювати як монолітна структура, де вся архітектура централізована в одному місці. Однак більшість вебзастосунків складаються з кількох компонентів, які працюють разом.

Як правило, архітектура вебзастосунків поділяється на дві основні групи: компоненти інтерфейсу користувача та структурні вебкомпоненти. Структурні компоненти включають як клієнтські, так і серверні елементи, які взаємодіють для забезпечення необхідної функціональності.

Архітектура системи також може включати як існуючі сторонні сервіси, так і спеціально розроблені рішення. Важливо підкреслити, що кожна архітектура є багаторівневою.

Для даного вебзастосунку було обрано архітектуру клієнт-сервер, що показано на рисунку 2.1 [32].

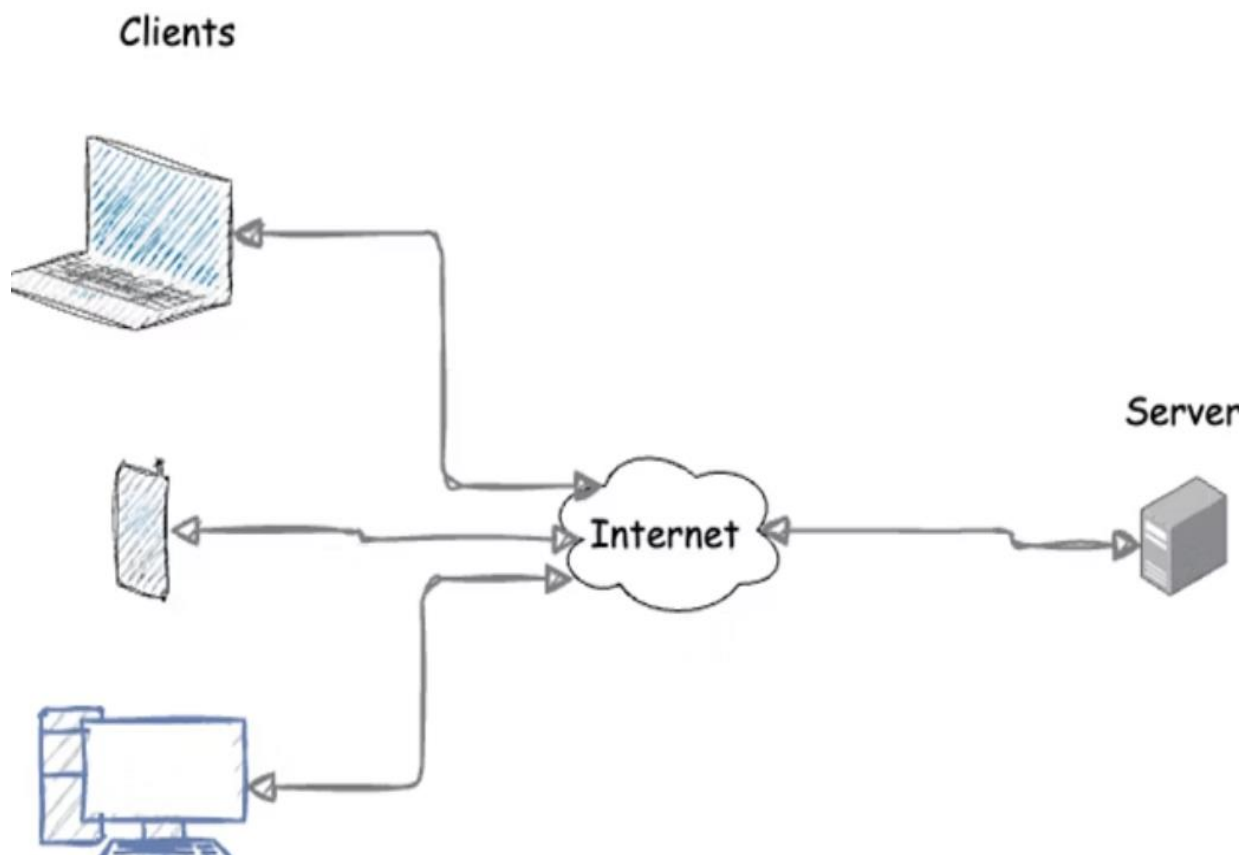


Рисунок 2.1 – Вигляд клієнт-серверної архітектури

2.2 Проектування логічної моделі бази даних

Схема бази даних є основою будь-якого вебзастосунку. Для магазину туристичного спорядження вона повинна включати такі ключові сутності (таблиці). Детальний опис знаходиться у таблицях 1-8.

Таблиця 1 - Users (Користувачі)

Зберігає інформацію про всіх користувачів системи, включаючи покупців та адміністраторів.

Назва Поля	Тип Даних	Обмеження / Примітки
id	BIGINT UNSIGNED	PRIMARY KEY, AUTO_INCREMENT
name	VARCHAR(255)	Ім'я користувача (для покупців) або Ім'я адміністратора
email	VARCHAR(255)	UNIQUE, Email користувача
password	VARCHAR(255)	Захешований пароль
role	ENUM('customer', 'admin')	Роль користувача (за замовчуванням: customer)
phone	VARCHAR(20)	Телефонний номер користувача, NULLABLE
shipping_address	TEXT	Адреса доставки за замовчуванням, NULLABLE
email_verified_at	TIMESTAMP	Час підтвердження email, NULLABLE
remember_token	VARCHAR(100)	Токен для «запам'ятати мене», NULLABLE
created_at	TIMESTAMP	Час створення запису
updated_at	TIMESTAMP	Час останнього оновлення запису

Таблиця 2 – Таблиця Categories (Категорії)

Зберігає ієрархічну структуру категорій товарів (наприклад, «Намети», «Спальники», «Рюкзаки»).

Назва Поля	Тип Даних	Обмеження / Примітки
id	BIGINT UNSIGNED	PRIMARY KEY, AUTO_INCREMENT
name	VARCHAR(255)	UNIQUE, Назва категорії
slug	VARCHAR(255)	UNIQUE, ЧПУ-рядок для URL (наприклад, palatki), для SEO
description	TEXT	Опис категорії, NULLABLE
parent_id	BIGINT UNSIGNED	FOREIGN KEY до categories.id (для підкатегорій), NULLABLE
created_at	TIMESTAMP	Час створення запису
updated_at	TIMESTAMP	Час останнього оновлення запису

Таблиця 3 – Таблиця Products (Товари)

Зберігає деталі про кожен туристичний товар, що продається.

Назва Поля	Тип Даних	Обмеження / Примітки
id	BIGINT UNSIGNED	PRIMARY KEY, AUTO_INCREMENT
category_id	BIGINT UNSIGNED	FOREIGN KEY до categories.id
name	VARCHAR(255)	Назва товару
slug	VARCHAR(255)	UNIQUE, ЧПУ-рядок для URL
description	TEXT	Детальний опис товару, NULLABLE

short_description	VARCHAR(500)	Короткий опис, NULLABLE
price	DECIMAL(10,2)	Ціна товару (наприклад, 12345678.99)
old_price	DECIMAL(10,2)	Стара ціна для акційних товарів, NULLABLE
stock	INT	Кількість товару на складі
sku	VARCHAR(50)	Артикул товару, UNIQUE, NULLABLE
image	VARCHAR(255)	Шлях до основного зображення товару, NULLABLE
attributes	JSON	Додаткові атрибути товару (вага, розмір, колір тощо), NULLABLE
is_active	BOOLEAN	Чи активний товар (відображається на сайті), за замовчуванням: TRUE
created_at	TIMESTAMP	Час створення запису
updated_at	TIMESTAMP	Час останнього оновлення запису

Таблиця 4 – Таблиця product_images (Зображення товарів)
Зберігає шляхи до додаткових зображень кожного товару.

Назва Поля	Тип Даних	Обмеження / Примітки
id	BIGINT UNSIGNED	PRIMARY KEY, AUTO_INCREMENT
product_id	BIGINT UNSIGNED	FOREIGN KEY до products.id
path	VARCHAR(255)	Шлях до файлу зображення
sort_order	INT	Порядок сортування зображень, NULLABLE
created_at	TIMESTAMP	Час створення запису

updated_at	TIMESTAMP	Час останнього оновлення запису
------------	-----------	---------------------------------

Таблиця 5 – Таблиця Orders (Замовлення)

Зберігає інформацію про кожне розміщене замовлення.

Назва Поля	Тип Даних	Обмеження / Примітки
id	BIGINT UNSIGNED	PRIMARY KEY, AUTO_INCREMENT
user_id	BIGINT UNSIGNED	FOREIGN KEY до users.id (якщо зареєстрований користувач), NULLABLE
order_number	VARCHAR(50)	UNIQUE, Унікальний номер замовлення (можна генерувати)
customer_name	VARCHAR(255)	Ім'я замовника
customer_email	VARCHAR(255)	Email замовника
customer_phone	VARCHAR(20)	Телефон замовника, NULLABLE
shipping_address	TEXT	Адреса доставки
shipping_method	VARCHAR(50)	Метод доставки (наприклад, 'nova_poshta', 'courier', 'pickup')
payment_method	VARCHAR(50)	Метод оплати (наприклад, 'card', 'cash_on_delivery', 'bank_transfer')
total_amount	DECIMAL(10,2)	Загальна сума замовлення
status	ENUM('pending', 'processing', 'shipped', 'delivered', 'cancelled', 'refunded')	Поточний статус замовлення, за замовчуванням: pending
comment	TEXT	Коментар клієнта до замовлення, NULLABLE

admin_notes	TEXT	Нотатки адміністратора, NULLABLE
created_at	TIMESTAMP	Час створення замовлення
updated_at	TIMESTAMP	Час останнього оновлення замовлення

Таблиця 6 – Таблиця Order_items (Позиції замовлення)

Зберігає деталі про кожен товар у замовленні. Це необхідно, оскільки ціна товару може змінитися після розміщення замовлення.

Назва Поля	Тип Даних	Обмеження / Примітки
id	BIGINT UNSIGNED	PRIMARY KEY, AUTO_INCREMENT
order_id	BIGINT UNSIGNED	FOREIGN KEY до orders.id
product_id	BIGINT UNSIGNED	FOREIGN KEY до products.id
product_name	VARCHAR(255)	Назва товару на момент замовлення (для історії)
price	DECIMAL(10,2)	Ціна товару на момент замовлення
quantity	INT	Кількість цього товару в замовленні
created_at	TIMESTAMP	Час створення запису
updated_at	TIMESTAMP	Час останнього оновлення запису

Таблиця 7 – Таблиця reviews (Відгуки)

Зберігає відгуки користувачів про товари.

Назва Поля	Тип Даних	Обмеження / Примітки
------------	-----------	----------------------

id	BIGINT UNSIGNED	PRIMARY KEY, AUTO_INCREMENT
user_id	BIGINT UNSIGNED	FOREIGN KEY до users.id
product_id	BIGINT UNSIGNED	FOREIGN KEY до products.id
rating	TINYINT UNSIGNED	Оцінка від 1 до 5
comment	TEXT	Текст відгуку, NULLABLE
is_approved	BOOLEAN	Чи схвалено відгук адміністратором, за замовчуванням: FALSE
created_at	TIMESTAMP	Час створення запису
updated_at	TIMESTAMP	Час останнього оновлення запису

Таблиця 8 – Таблиця Promotions (Акції/Знижки)

Зберігає інформацію про промокоди або загальні акції.

Назва Поля	Тип Даних	Обмеження / Примітки
id	BIGINT UNSIGNED	PRIMARY KEY, AUTO_INCREMENT
code	VARCHAR(50)	Промокод (наприклад, 'SUMMER20'), UNIQUE, NULLABLE
type	ENUM('percentage', 'fixed_amount', 'free_shipping')	Тип знижки
value	DECIMAL(10,2)	Значення знижки (наприклад, 10.00 для 10% або 50.00 грн)
start_date	TIMESTAMP	Дата початку акції
end_date	TIMESTAMP	Дата закінчення акції, NULLABLE

min_order_amount	DECIMAL(10,2)	Мінімальна сума замовлення для застосування, NULLABLE
usage_limit	INT	Кількість разів, яку можна використовувати промокод, NULLABLE
used_count	INT	Скільки разів промокод вже використано, за замовчуванням: 0
is_active	BOOLEAN	Чи активна акція, за замовчуванням: TRUE
created_at	TIMESTAMP	Час створення запису
updated_at	TIMESTAMP	Час останнього оновлення запису

Типи зв'язків, що використовують у базі даних: Один-до-багатьох (One-to-Many), Багато-до-одного (Many-to-One),

Один-до-багатьох (One-to-Many):

- users до orders, тобто один користувач може мати багато замовлень. (user_id в orders);
- categories до products, тобто одна категорія може мати багато товарів. (category_id в products);
- products до product_images, тобто один товар може мати багато зображень. (product_id в product_images);
- orders до order_items, тобто одне замовлення може мати багато позицій (товарів). (order_id в order_items);
- products до reviews, тобто один товар може мати багато відгуків. (product_id в reviews)
- users до reviews, тобто один користувач може залишити багато відгуків. (user_id в reviews).

Багато-до-одного (Many-to-One):

– products до categories, тобто багато товарів належать одній категорії. (category_id в products);

– order_items до orders, тобто багато позицій належать одному замовленню. (order_id в order_items);

– reviews до users та products, тобто кожен відгук належить одному користувачеві та одному товару. (user_id та product_id в reviews).

Самореферентний зв'язок (Self-referencing):

– categories до categories, де категорія може мати батьківську категорію (parent_id в categories), що дозволяє створювати ієрархії.

2.3 Проектування інтерфейсу користувача

UX-дизайн допомагає забезпечити безперешкодний і приємний досвід подорожей. Дізнайтеся про його важливість у туристичній індустрії.

Не можна недооцінювати важливість забезпечення оптимального досвіду для користувачів, які користуються вебсайтами або застосунками в туристичній галузі. Зосередившись на створенні ефективною та зручною платформи, туристичні компанії можуть запропонувати своїм клієнтам приємний і безперешкодний досвід подорожей і, водночас, завоювати довіру та авторитет.

Ось чому важливо враховувати користувацький досвід під час проектування та розробки цифрових продуктів, що впроваджуються туристичною галуззю. Це можуть бути онлайн-продажі, такі як електронна комерція або додатки для продажу авіаквитків чи автобусних квитків, бронювання житла чи укладання договорів на туристичні пакети.

UX, що розшифровується як «користувацький досвід», означає спосіб взаємодії та сприйняття людиною зручності, доступності та задоволеності при використанні продукту або послуги, таких як вебсайт, мобільний застосунок, програмне забезпечення або будь-яка інша цифрова система.

					КвРІПЗ.2101085.01.14.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		34

UX-дизайн, або дизайн користувацького досвіду, зосереджується на створенні продуктів, які є інтуїтивно зрозумілими, ефективними та приємними у використанні для кінцевих користувачів, враховуючи їхні потреби, бажання та здібності. Це включає в себе дослідження, проектування, оцінку та безперервну ітерацію для покращення користувацького досвіду з плином часу.

Основна мета UX полягає в тому, щоб будь-який користувач отримувал максимальну цінність і насолоджувався позитивним досвідом при взаємодії з цифровим продуктом або послугою.

UX має вирішальне значення в туристичній індустрії, оскільки він визначає задоволеність користувачів при взаємодії з додатками і платформами. На такому висококонкурентному ринку, як туристичний, надання безперебійного, інтуїтивно зрозумілого та персоналізованого досвіду має вирішальне значення для того, щоб виділитися та утримати клієнтів. Крім того, добре продуманий UX може впливати на сприйняття бренду і рішення про покупку, що безпосередньо впливає на успіх бізнесу, збільшуючи продажі.

UX в туристичній індустрії полегшує бронювання та планування подорожей. Хороший UX-дизайн на вебсайтах або в додатках полегшує навігацію, пошук і бронювання квитків. Це зменшує тертя і робить процес більш впорядкованим і приємним для користувачів.

Підвищує задоволеність клієнтів: Позитивний користувацький досвід підвищує задоволеність клієнтів, що, в свою чергу, може генерувати лояльність до бренду та рекомендації іншим мандрівникам. Це має вирішальне значення в такій конкурентній галузі, як подорожі.

Користувачі схильні довіряти вебсайтам і застосункам, які пропонують плавний та інтуїтивно зрозумілий користувацький досвід. Зрозуміла навігація, детальна інформація та безпечні способи оплати допомагають зміцнити цю довіру.

Адаптивність до мобільних пристроїв: З ростом використання мобільних пристроїв для пошуку і бронювання подорожей, UX-дизайн повинен бути

					КвРІПЗ.2101085.01.14.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		35

адаптивним, тобто правильно підлаштовуватися під різні розміри екранів і забезпечувати послідовний і функціональний досвід на мобільних пристроях.

Персоналізація та рекомендації: UX може допомогти надати персоналізовані рекомендації на основі вподобань користувача та історії пошуку, що підвищує релевантність представлених варіантів і полегшує прийняття рішень.

UX є важливим при розробці таких продуктів, як платформа електронної комерції, це частина завдання - зрозуміти користувачів і запропонувати рішення, які їм дійсно подобаються і приносять задоволення. Наприклад, коли команда отримує запит від партнера на продукт, першим кроком є спілкування з командою дизайнерів.

Повинна працювати досвідчена команда професіоналів розробників, дизайнерів, щоб розглянути юзабіліті та користувацький досвід в цілому. Вони оцінюють, чи доречна типографіка, чи дозволить робочий процес користувачам дотримуватися звичної для них послідовності, чи вдосконалять його, а також інші функціональні можливості.

Важливість дизайну користувацького досвіду (UX) у туристичній індустрії не викликає сумнівів. Зосередившись на створенні ефективної та зручної платформи, компанії можуть забезпечити своїм клієнтам приємний досвід подорожей, що, в свою чергу, формує довіру та авторитет.

Для проектування користувацького інтерфейсу досить вдало використовуються прототипи.

2.4 Аналіз та вибір технологій і методів реалізації системи

Для розробки необхідно проаналізувати існуючі методи, засоби, середовища та інструментарій програмування. В сучасних умовах це діапазон надзвичайно великий і вибір таких технологій буде за розробником

					КвРІПЗ.2101085.01.14.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		36

програмного забезпечення, однак із врахування вимог до розроблюваного програмного продукту.

Для реалізації даного програмного продукту було обрано платформу ASP.NET з усіма її інструментами та можливостями.

ASP.NET - це наступна версія Active Server Pages (ASP), яка є уніфікованою платформою веброзробки, що надає послуги, необхідні розробникам для створення вебзастосунків корпоративного класу. Хоча ASP.NET значною мірою сумісний з синтаксисом, він також надає нову модель програмування та інфраструктуру для більш безпечних, масштабованих та стабільних застосунків.

ASP.NET є компільованим середовищем на основі .NET, де можна створювати застосунки будь-якою мовою, сумісною з .NET, включаючи Visual Basic .NET, C# та JScript .NET. Крім того, весь .NET Framework доступний для будь-якої програми ASP.NET. Розробники можуть легко отримати доступ до переваг цих технологій, включаючи кероване середовище виконання загальномовних даних (CLR), безпеку типів, успадкування тощо. ASP.NET розроблено для безперебійної роботи з WYSIWYG HTML-редакторами та іншими інструментами програмування, включаючи Microsoft Visual Studio .NET. Це не лише спрощує веброзробку, але й надає всі переваги, які пропонують ці інструменти, включаючи графічний інтерфейс, який розробники можуть використовувати для розміщення серверних елементів керування на вебсторінці, та повністю інтегровану підтримку налагодження.

Розробники можуть вибрати одну з двох наступних функцій під час створення ASP.NET застосунку вебформи та вебсервіси або комбінувати їх будь-яким зручним для них способом. Кожна з них підтримується однією і тією ж інфраструктурою, яка дозволяє використовувати схеми автентифікації, кешувати часто використовувані дані або налаштувати конфігурацію вашого додатку, і це лише деякі з можливостей.

.NET Framework – середовище загальномовного виконання. Доступ до баз даних із програм ASP.NET – це часто використовуваний метод відображення

					КвРІПЗ.2101085.01.14.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		37

даних відвідувачам вебсайту. ASP.NET спрощує доступ до баз даних для цієї мети, ніж будь-коли. Він також дозволяє нам керувати базою даних з вашого коду.

ASP.NET надає просту модель, яка дозволяє веброзробникам писати логіку, що працює на рівні програми. Розробники можуть писати цей код у текстовому файлі `global.aspx` або в скомпільованому класі, розгорнутому як збірка. Ця логіка може включати події рівня програми, але розробники можуть легко розширити цю модель відповідно до потреб своєї вебпрограми. ASP.NET надає прості у використанні засоби обробки стану програм і сеансів, які знайомі розробникам ASP і легкосумісні з усіма іншими API .NET Framework. ASP.NET пропонує інтерфейси `HttpHandler` та `HttpModule`. Реалізація інтерфейсу `HttpHandler` надає вам засоби взаємодії з низькорівневими службами запитів та відповідей веб-сервера IIS та забезпечує функціональність подібну до розширень ISAPI, але з простішою моделлю програмування. Реалізація інтерфейсу `HttpModule` дозволяє включати користувацькі події, які беруть участь у кожному запиті, зробленому до програми.

ASP.NET використовує переваги покращень продуктивності, що містяться в .NET Framework та середовищі загальномовного виконання. Крім того, він був розроблений для забезпечення значного покращення продуктивності порівняно з ASP та іншими платформами веб-розробки. Весь код ASP.NET компілюється, а не інтерпретується, що дозволяє виконувати раннє зв'язування, сувору типізацію та компіляцію «точно в часі» (JIT) у власний код, і це лише деякі з його переваг. ASP.NET також легко факторизується, що означає, що розробники можуть видаляти модулі (наприклад, модуль сеансу), які не є релевантними для програми, що вони розробляють. ASP.NET надає розширені служби кешування (як вбудовані служби, так і API кешування). ASP.NET також постачається з лічильниками продуктивності, які розробники та системні адміністратори можуть відстежувати для тестування нових програм та збору показників для існуючих програм. Написання власних операторів налагодження на вебсторінці може

					КвРІПЗ.2101085.01.14.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		38

дуже допомогти у вирішенні проблем із кодом програми. Однак, це може спричинити незручності, якщо їх не видалити.

Проблема полягає в тому, що видалення операторів налагодження зі сторінок, коли програма готова до перенесення на робочий сервер, може вимагати значних зусиль. ASP.NET пропонує клас Trace Context, який дозволяє нам писати власні оператори налагодження на наших сторінках під час їх розробки.

Вони з'являються лише тоді, коли увімкнено трасування для сторінки або всієї програми. Увімкнення трасування також додає деталі про запит до сторінки або, якщо розробник вкаже спеціального переглядача трасування, який зберігається в кореневому каталозі вашої програми. .NET Framework та ASP.NET надають схеми авторизації та автентифікації за замовчуванням для вебпрограм. Можна легко видалити, застосунки або замінити ці схеми, залежно від потреб програми. Параметри конфігурації ASP.NET зберігаються у файлах на основі XML, які є читабельними та доступними для запису людиною. Кожна з програм може мати окремий файл конфігурації, і можна розширити схему конфігурації відповідно до вимог.

Загалом для розроблюваного вебзастосування передбачено використання таких технологій та інструментів описаних нижче для користувацької та серверної частини.

Фронтенд:

- HTML5, CSS3, JavaScript для реалізації основи вебінтерфейсу;
- Bootstrap / Tailwind CSS: Для адаптивного дизайну та швидкої розробки інтерфейсу;
- jQuery / React / Angular / Vue.js: Для інтерактивності та покращення користувацького досвіду (особливо для SPA-архітектури).

Бекенд:

- ASP.NET Core MVC (для традиційного вебзастосування) / ASP.NET Core Web API (для SPA/мобільного застосування): Основний фреймворк для розробки серверної логіки;

					КвРІПЗ.2101085.01.14.ПЗ	Арк.
						39
Зм.	Арк	№ докум.	Підпис	Дата		

- Entity Framework Core: ORM для взаємодії з базою даних;
- DI (Dependency Injection): Вбудований механізм ASP.NET Core для управління залежностями.

Інструменти розробки:

- Visual Studio / Visual Studio Code, тобто IDE для .NET розробки;
- SQL Server Management Studio (SSMS) / DBeaver / pgAdmin для управління базою даних;
- Git в якості системи контролю версій.

					КвРІПЗ.2101085.01.14.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		40

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ

3.1 Реалізація бази даних

Розробка бази даних для вебзастосунку з торгівлі туристичним спорядженням за допомогою ASP.NET вимагає ретельного планування та реалізації. Нижче наведено опис ключових аспектів такої реалізації. І характерну роль тут виконує вибір систем управління базами даних (СУБД). Для ASP.NET застосунків найчастіше використовуються такі СУБД:

- Microsoft SQL Server, що є нативним та найбільш поширеним вибором для .NET-розробок. Забезпечує високу продуктивність, масштабованість та інтеграцію з інструментами Microsoft;

- PostgreSQL – це потужна, відкрита та дуже надійна СУБД. Гарний вибір, якщо потрібна кросплатформенність або економія на ліцензіях;

- MySQL є популярною відкритою СУБД, широко використовується в веб-розробці. Може бути доречною для менших проектів або тих, що вже мають досвід роботи з MySQL;

Azure SQL Database - це хмарна технологія Microsoft для SQL Server, що забезпечує високу доступність, автоматичне масштабування та управління. Ідеально підходить для хмарних ASP.NET застосунків.

Вибір СУБД залежить від масштабу проекту, бюджету, вимог до продуктивності та наявності експертизи.

Для взаємодії ASP.NET застосунку з базою даних рекомендується використовувати ORM. Найпопулярнішим ORM для .NET є Entity Framework Core.

Переваги Entity Framework Core:

- зручність, бо дозволяє працювати з базою даних за допомогою об'єктів C#, що значно спрощує розробку та підтримку;
- productivity, що автоматично генерує SQL-запити, звільняючи розробника від ручного написання;

					КвРІПЗ.2101085.01.14.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		41

- migration забезпечує механізм міграцій, що дозволяє легко управляти змінами в схемі бази даних з часом;
- LINQ-інтеграція (Language Integrated Query), що дозволяє писати запити до бази даних прямо в коді С#;
- Data Seeding з можливістю додавати початкові дані в базу даних під час міграцій.

Для початку відбувається створення моделей (Entity Classes), де кожна таблиця в базі даних відображається на клас С#. Лістинг коду показано нижче.

```

public class Product
{
    public int Id { get; set; }
    public string Name { get; set; }
    public string Description { get; set; }
    public decimal Price { get; set; }
    public int StockQuantity { get; set; }
    public int CategoryId { get; set; }
    public Category Category { get; set; } // Navigation property
    // ... інші властивості
}

public class Category
{
    public int Id { get; set; }
    public string Name { get; set; }
    // ... інші властивості
    public ICollection<Product> Products { get; set; } // Navigation property
}

```

					КвРІПЗ.2101085.01.14.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		42

Далі створюється Створення DbContext: Клас DbContext є точкою входу для взаємодії з базою даних.

```
public class ApplicationDbContext : DbContext
{
    public ApplicationDbContext(DbContextOptions<ApplicationDbContext>
options)
        : base(options)
    {
    }
    public DbSet<Product> Products { get; set; }
    public DbSet<Category> Categories { get; set; }
    public DbSet<User> Users { get; set; }
    public DbSet<Order> Orders { get; set; }
    public DbSet<OrderItem> OrderItems { get; set; }
    // ... інші DbSet'и
    protected override void OnModelCreating(ModelBuilder modelBuilder)
    {
        // Налаштування зв'язків та обмежень, якщо потрібно
        modelBuilder.Entity<OrderItem>()
            .HasOne(oi => oi.Product)
            .WithMany()
            .HasForeignKey(oi => oi.ProductId);
        // Налаштування для ASP.NET Core Identity
        base.OnModelCreating(modelBuilder);
    }
}
```

Конфігурація в Program.cs

```
builder.Services.AddDbContext<ApplicationDbContext>(options =>
```

					КвРІПЗ.2101085.01.14.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		43

```
options.UseSqlServer(builder.Configuration.GetConnectionString("DefaultConnecti  
on")));
```

Використання в контролерах/сервісах:

```
public class ProductsController : ControllerBase  
{  
    private readonly ApplicationDbContext _context;  
  
    public ProductsController(ApplicationDbContext context)  
    {  
        _context = context;  
    }  
    [HttpGet]  
    public async Task<ActionResult<IEnumerable<Product>>>  
GetProducts()  
    {  
        return await _context.Products.Include(p =>  
p.Category).ToListAsync();  
    }  
    [HttpPost]  
    public async Task<ActionResult<Product>> PostProduct(Product  
product)  
    {  
        _context.Products.Add(product);  
        await _context.SaveChangesAsync();  
        return CreatedAtAction("GetProduct", new { id = product.Id },  
product);  
    }  
}
```

					КвРІПЗ.2101085.01.14.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		44

Також для реалізації важливо враховувати безпеку бази даних, оптимізацію продуктивності, масштабованість, принципи CI/CD та розгортання, а також додаткові сутності (для майбутнього розширення). Найважливішим аспектом, звісно є безпека бази даних, що має передбачати:

- хешування паролів;
- параметризовані запити;
- розмежування доступу;
- резервне копіювання;
- принципи найменших привілеїв;
- резервне копіювання;
- шифрування даних.

Не можна зберігати паролі у відкритому вигляді. Для цього використовуються надійні алгоритми хешування (наприклад, BCrypt, Argon2) з сіллю. ASP.NET Core Identity вже включає в себе такі механізми. Entity Framework Core автоматично використовує параметризовані запити, що захищає від SQL-ін'єкцій. Доцільним є розмежування доступу на основі ролей (Role-Based Access Control, RBAC) для адміністраторів, звичайних користувачів тощо. Користувач бази даних, під яким працює застосунок, повинен мати лише ті дозволи, які необхідні для його функціонування. Необхідною є регулярне створення резервних копій бази даних. Для конфіденційних даних (наприклад, платіжних даних, хоча зазвичай вони не зберігаються в базі даних вебзастосунку, а обробляються платіжними шлюзами) можна розглядати шифрування на рівні бази даних або на рівні застосунку.

Для оптимізація продуктивності використовуються:

- індекси;
- кешування;
- денормалізація;
- оптимізація SQL-запитів;

					КвРІПЗ.2101085.01.14.ПЗ	Арк.
						45
Зм.	Арк	№ докум.	Підпис	Дата		

- розбиття на сторінки (Pagination).

Створюйте індекси на стовпцях, які часто використовуються в умовах WHERE, JOIN та ORDER BY. Entity Framework Core дозволяє додавати індекси через Fluent API або атрибути.

Потрібно враховувати так звані «ліниві» завантаження (Lazy Loading) в Entity Framework Core, оскільки вони можуть призвести до проблеми N+1 запитів. Використовуйте Include() або ThenInclude() для так званого «жадібного завантаження (Eager Loading) пов'язаних даних. Потрібно кешувати часто запитувані, але рідко змінювані дані (наприклад, категорії товарів, список виробників) за допомогою In-Memory Cache або розподіленого кешу (Redis, Memcached). Для звітів або часто запитуваних комбінованих даних, денормалізація може покращити продуктивність за рахунок надлишковості. Однак, це збільшує складність підтримки цілісності даних.

Іноді може знадобитися ручна оптимізація складних SQL-запитів або використання «сирих» SQL-запитів через Entity Framework Core, якщо LINQ-запити не дають бажаної продуктивності. Завжди використовуйте розбиття на сторінки для великих наборів даних (наприклад, списку товарів, замовлень), щоб уникнути завантаження всіх даних одразу.

Велике значення має також масштабованість, яка включає вертикальне масштабування, тобто збільшення ресурсів (CPU, RAM) сервера бази даних; горизонтальне масштабування, тобто розділення бази даних на кілька серверів (шардинг, реплікація), що є складнішим підходом, який зазвичай розглядається для великих проектів; використання хмарних баз даних, наприклад Azure SQL Database, AWS RDS тощо, що забезпечують вбудовані можливості масштабування та високої доступності.

До принципів CI/CD та розгортання відносяться:

- автоматизовані міграції;
- версійний контроль.

Щоб автоматично оновлювати схему бази даних при розгортанні нових версій застосунку застосовується міграцій бази даних у конвеєр CI/CD

					КвРІПЗ.2101085.01.14.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		46

Збереження міграції бази даних у системі версійного контролю (Git) разом з кодом застосунку відбувається за допомогою версійного контролю.

Для майбутнього розширення можуть використовуватись додаткові сутності, наприклад:

- атрибути товарів (ProductAttributes), що використовуються для специфічних характеристик товарів (наприклад, «колір», «розмір», «матеріал»), які можуть змінюватися для різних категорій;
- варіанти товарів (ProductVariants), якщо товар має кілька варіантів (наприклад, куртка в різних розмірах і кольорах, кожен з яких має свій SKU та залишок);
- списки бажань (Wishlists) для збереження товарів, які користувач хоче придбати пізніше;
- пошукові індекси для повнотекстового пошуку за товарами (може бути реалізовано за допомогою спеціалізованих пошукових систем, таких як Elasticsearch, або вбудованих функцій СУБД);
- журнали (Logs) для відстеження подій у застосунку (дії користувачів, помилки).

3.2 Реалізація модулів системи

Коли є чітке розуміння реалізації бази даних, можна поглибитись в опис інших ключових модулів вебзастосунку для продажу туристичного спорядження, розробленого за допомогою ASP.NET. Ці модулі взаємодіятимуть з описаною базою даних для забезпечення повної функціональності електронної комерції.

За управління доступом користувачів до системи, забезпечуючи безпеку та персоналізацію Модуль аутентифікації та авторизації. Реалізується ASP.NET Core Identity, що є найпотужнішим та найрекомендованішим фреймворком для

					КвРІПЗ.2101085.01.14.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		47

управління користувачами, ролями, паролями та іншими даними аутентифікації в ASP.NET Core. Він вже інтегрований з Entity Framework Core і дозволяє легко налаштувати таблиці Users (що розглянуто раніше) та Roles у базі даних.

Реєстрація Користувачів містить форму реєстрації, що збирає дані, такі як ім'я користувача, електронна пошта, пароль. Валідація даних на стороні клієнта (JavaScript) та сервера (ASP.NET Core Model Binding та Data Annotations) для забезпечення коректності та безпеки. Хешування паролів автоматично виконується ASP.NET Core Identity. Підтвердження електронної пошти для запобігання фальшивим реєстраціям.

Вхід (Login) відбувається за формою входу з полями для імені користувача/електронної пошти та пароля. ASP.NET Core Identity перевіряє облікові дані та видає аутентифікаційні куки (для традиційних веб-застосунків) або JWT-токен (для API та SPA) із використанням функції «Запам'ятати мене».

Управління паролями працює завдяки функціоналу відновлення пароля (зазвичай через надсилання токена на електронну пошту) із можливістю зміни пароля для авторизованих користувачів.

Управління профілем користувача:

- сторінка для перегляду та редагування особистих даних (адреса, телефон, ім'я тощо);
- інтеграція з таблицею Users для оновлення цих даних.

Ролі та дозволи (Authorization) передбачають використання вбудованих функцій ролей ASP.NET Core Identity (IsAdmin у таблиці Users може бути реалізовано через систему ролей). Декоратори [Authorize(Roles = «Admin»)] над контролерами або діями для обмеження доступу. Наприклад, тільки адміністратори можуть додавати, редагувати або видаляти товари. Політики авторизації для складніших сценаріїв доступу.

Модуль управління каталогом товарів є основою застосунку, що дозволяє користувачам переглядати, шукати та фільтрувати товари.

					КвРІПЗ.2101085.01.14.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		48

Перегляд Категорій створюється за рахунок динамічного формування меню категорій на основі даних з таблиці Categories. Також передбачено можливість відображення ієрархічних категорій (батьківська/дочірня).

Список товарів передбачає:

- відображення товарів з таблиці Products;
- пагінацію для великих обсягів даних, щоб уникнути завантаження всіх товарів одразу;
- фільтрацію за категоріями, виробниками, ціновим діапазоном, наявністю;
- сортування за ціною (зростання/спадання), назвою, датою додавання, популярністю (кількістю замовлень);
- повнотекстовий пошук за назвою та описом товару. Можна використовувати вбудовані можливості СУБД або інтегрувати з ElasticSearch/Solr для більш потужного пошуку.

Детальна сторінка опису товару містить відображення повної інформації про товар (назва, опис, ціна, залишок, зображення, характеристики), інтеграцію з таблицями Products, Manufacturers, Categories; відгуки та рейтинги, що відображаються з таблиці Reviews та середнього рейтингу із можливістю додавання нового відгуку (для авторизованих користувачів); кнопка «Додати в кошик».

Модуль Адмін-панелі для управління каталогом:

- CRUD-операції (Create, Read, Update, Delete) для товарів, категорій, виробників;
- форми для додавання/редагування товарів з полями для всіх властивостей (ціна, кількість на складі, зображення, опис тощо);
- можливість завантаження кількох зображень для товару;
- обробка файлів із використанням ASP.NET Core для завантаження зображень на сервер (зазвичай в окрему папку) та збереження шляхів до них у базі даних.

					КвРІПЗ.2101085.01.14.ПЗ	Арк.
						49
Зм.	Арк	№ докум.	Підпис	Дата		

Модуль Кошика та Оформлення замовлення, що дозволяє користувачам збирати бажані товари та завершувати покупку.

Модуль Додавання в Кошик:

- коли користувач натискає «Додати в кошик», товар додається до сесії користувача або до таблиці ShoppingCarts у базі даних;
- обробка випадків, коли товар вже є в кошику (збільшення кількості);
- перевірка наявності товару на складі (StockQuantity).

Модуль Перегляд Кошика:

- сторінка, що відображає всі товари в кошику, їхню кількість, ціну та загальну суму;
- можливість змінити кількість товару або видалити його з кошика;
- Застосування промокодів/знижок (інтеграція з таблицею Discounts).

```
namespace YourProjectName.Models
{
    public class ShoppingCartItem
    {
        public int ProductId { get; set; }
        public string ProductName { get; set; }
        public decimal Price { get; set; }
        public int Quantity { get; set; }
        public string ImageUrl { get; set; } // Для відображення в кошику
    }
}
```

Модуль Оформлення замовлення (Checkout), де оформлення зазвичай включає:

- інформацію про доставку, тобто адреса, контактні дані (автоматично заповнюються з профілю користувача, якщо він авторизований);
- спосіб доставки передбачає вибір доступних методів доставки (наприклад, Нова Пошта, Укрпошта, кур'єр). Можна зберігати тарифи доставки в окремій таблиці або налаштувати в коді;

					КвРІПЗ.2101085.01.14.ПЗ	Арк.
						50
Зм.	Арк	№ докум.	Підпис	Дата		

- спосіб оплати, тобто вибір методів оплати (наприклад, онлайн-оплата карткою, післяплата);
- валідація, тобто перевірку всіх даних перед підтвердженням замовлення;
- створення замовлення, після підтвердження замовлення дані записуються в таблиці Orders та OrderItems;
- зменшення кількості товару на складі, після якого відбувається автоматичне оновлення StockQuantity у таблиці Products.

Модуль Підтвердження замовлення:

- сторінка подяки з номером замовлення та деталями;
- надсилання електронного листа з підтвердженням замовлення користувачеві.

Модуль Інтеграції з платіжними системи забезпечує можливість безпечних онлайн-платежів. Вибір платіжного шлюзу відбувається на основі наявних. В Україні популярні платіжні шлюзи: Portmone, LiqPay, WayForPay, Stripe (міжнародний). Вибір залежить від комісій, функціоналу, зручності інтеграції. Сюди також відноситься й інтеграція API. Більшість платіжних систем надають SDK або API для інтеграції. Перенаправлення користувача на сторінку платіжного шлюзу або відображення віджету оплати на сайті. Обробка Callback/Webhooks: Платіжний шлюз надсилає зворотні виклики на ваш сервер, повідомляючи про статус платежу (успішно, невдало, в очікуванні). Оновлення OrderStatus в таблиці Orders та збереження TransactionId. Безпека передбачає заборону збереження конфіденційних платіжних даних (номери карток, CVV) на сервері. Вся обробка цих даних повинна відбуватися на стороні платіжного шлюзу. Використання HTTPS для всього вебзастосунку для шифрування трафіку.

Модуль Управління замовленнями (для Адміністратора), що дозволяє адміністраторам ефективно керувати замовленнями та передбачає: список замовлень, деталізацію замовлень, пошук замовлень,

					КвРІПЗ.2101085.01.14.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		51

Список Заказов:

- таблиця всіх замовлень з можливістю фільтрації за статусом, датою, користувачем;
- сортування та пагінація.

Деталі Заказа:

- сторінка, що відображає повну інформацію про заказ: дані користувача, адреса доставки, список замовлених товарів, суми;
- можливість змінити статус заказа (наприклад, «В обробці» - «Відправлено» - «Доставлено»);
- можливість додати коментар до заказа (для внутрішнього використання);
- посилання на профіль користувача, що зробив заказ.

Пошук замовлень здійснюється за номером заказа, іменем користувача, назвою товару. Експорт звітів надає можливість експортувати дані замовлень у форматах CSV або Excel для аналізу.

Модуль Управління знижками та акціями (для Адміністратора) дозволяє адміністраторам створювати та керувати маркетинговими кампаніями, передбачає застосування знижок.

CRUD-операції для знижок:

- Форма для створення нового промокоду, а саме Code, Description, DiscountType, DiscountValue, StartDate, EndDate, MinimumOrderAmount, AppliesToProductId;
- список всіх активних та неактивних знижок;
- можливість редагувати та видаляти знижки.

Модуль відгуків та рейтингів, що дозволяє користувачам ділитися враженнями про товари.

Додавання відгуку:

- форма на сторінці товару для авторизованих користувачів;
- поля для оцінки (Rating) та коментаря (Comment);

					КвРІПЗ.2101085.01.14.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		52

- валідація, тобто забезпечення того, що користувач може залишити лише один відгук на товар (або дозволити кілька, але з відстеженням);

- збереження в таблиці Reviews.

Модерація відгуків (для Адміністратора):

- адміністративна панель для перегляду та схвалення або ж відхилення відгуків;

- можливість видаляти небажані відгуки.

Відображення відгуків:

- виведення відгуків на сторінці товару;

- розрахунок та відображення середнього рейтингу товару.

Модуль Адміністрування тобто загальний модуль для управління всією системою передбачає панель керування, управління користувачами, журнали. Панель керування (Dashboard) містить огляд основних метрик (кількість замовлень, нові користувачі, дохід). Журнали (Logs) допомагають здійснювати перегляд системних журналів для відстеження помилок або важливих подій. Конфігурація магазину дозволяє здійснити налаштування загальних параметрів (наприклад, налаштування електронної пошти для сповіщень, умови доставки).

Управління Користувачами:

- список користувачів, пошук, фільтрація;

- можливість блокувати/розблокувати користувачів;

- змінювати ролі користувачів (наприклад, призначити адміністратором).

Ці модулі забезпечують комплексний функціонал для вебзастосунку з торгівлі туристичним спорядженням. Кожен модуль має свою зону відповідальності, а їхня взаємодія, побудована на базі даних та ASP.NET Core, формує цілісну та ефективну систему.

3.3 Реалізація інтерфейсу. Інструкція користувача

					КвРІПЗ.2101085.01.14.ПЗ	Арк.
						53
Зм.	Арк	№ докум.	Підпис	Дата		

Для вірного функціонування розроблюваного програмного продукту доцільно користуватись інструкцією для користувача. Як і у більшості випадків користування вебзастосунків не потребує якихось спеціальних навичок та є простою і лаконічною. Набираючи відповідну адресу в адресному рядку користувач переходить на головну сторінку, вигляд якої зображено на рисунку 3.1.

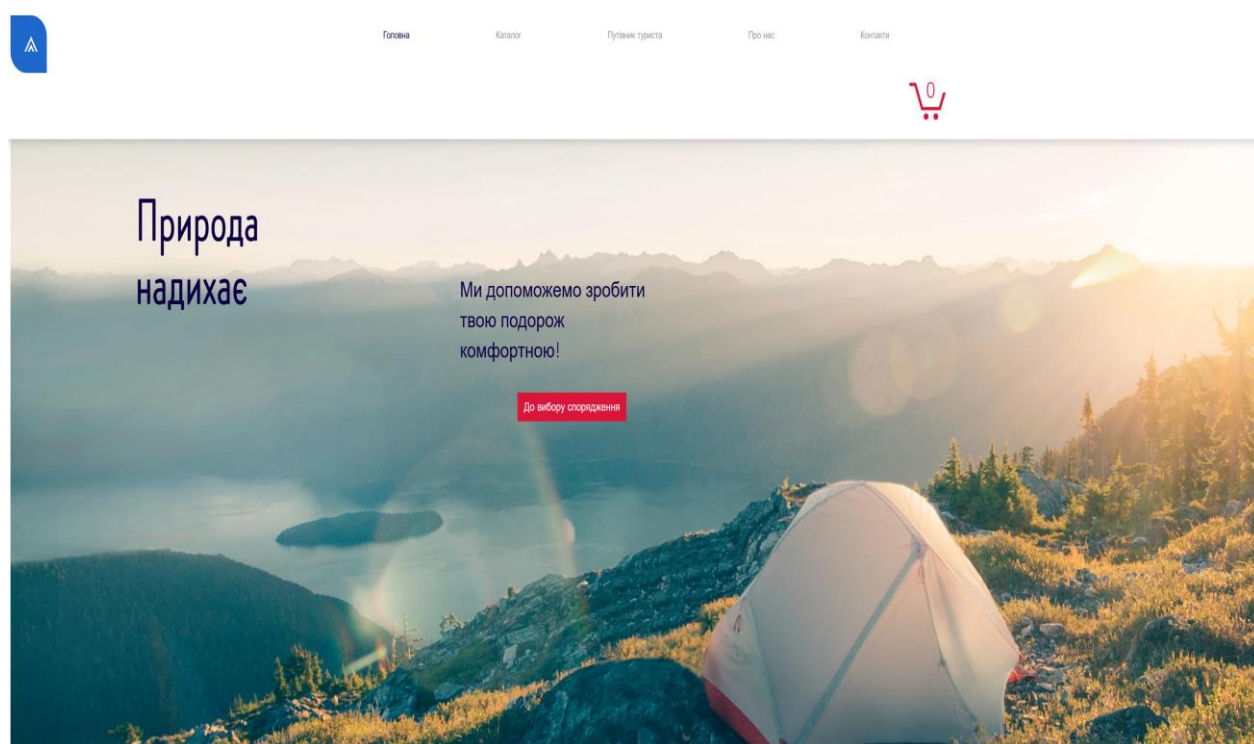


Рисунок 3.1 - Вигляд Головної сторінки вебзастосунку

На Головній сторінці містить логотип компанії, що займається продажем туристичного спорядження, меню, що складається із таких пунктів: Головна, Каталог, Путівник туриста, Про нас, Контакти. Також на головній сторінці є іконка кошика.

При переході на меню Каталог відкриється сторінка із переліком товарів та можливістю здійснити вибір та покупку необхідного товару туристичного спрямування. Зображення даної сторінки Каталогу показано на рисунку 3.2.

					КвРІПЗ.2101085.01.14.ПЗ	Арк.
						54
Зм.	Арк	№ докум.	Підпис	Дата		

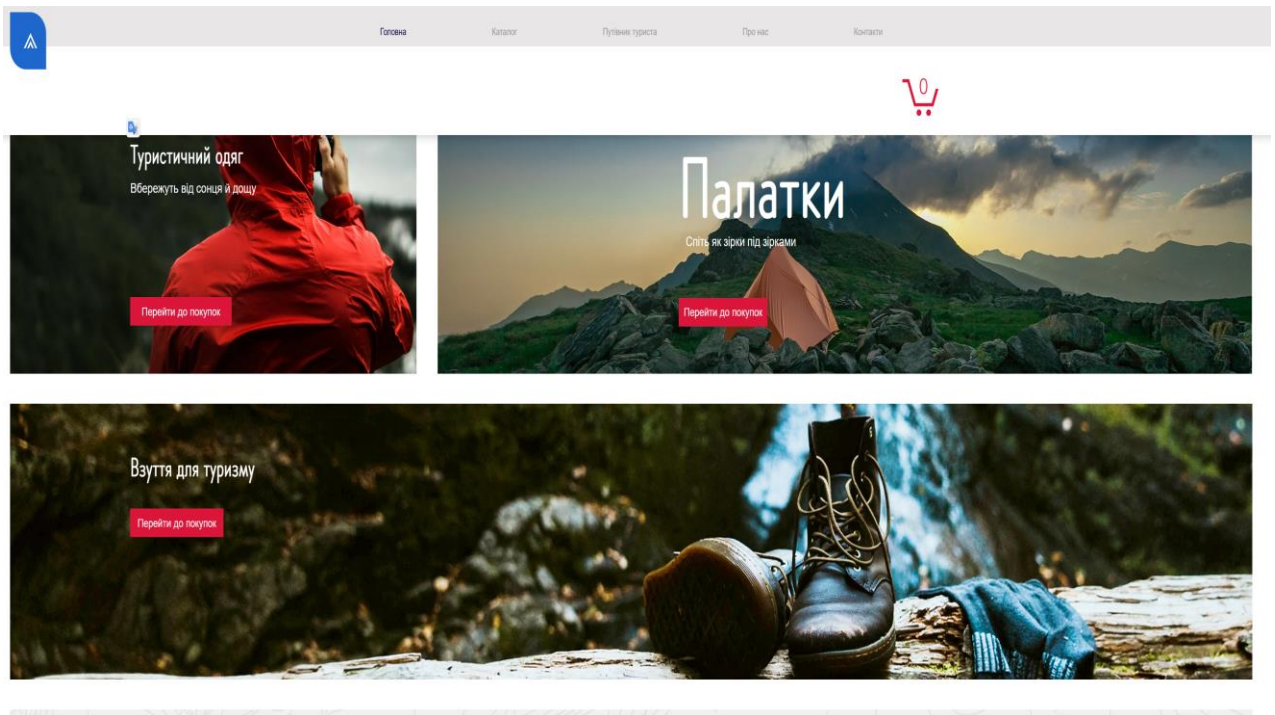


Рисунок 3.2 – Сторінка меню Каталог

Для користування ресурсом можна зареєструватись, що дає можливість запам'ятовувати дані для доставки, оплати, можливості залишати відгук тощо. Якщо користувач зареєструвався, то він може увійти до системи, якщо введе свою електронну пошту та пароль. На рисунку 3.3 показано сторінку із можливістю здійснити авторизацію.

Login

Email

Password

Login

Рисунок 3.3 – Сторінка авторизації

Також у застосунку передбачена можливість залишати запитання та заповняти форму для зворотнього зв'язку, зображення форми показано на рисунку 3.4.

Ваше ім'я

Ваш email

Ваш коментар або пропозиція

Надіслати

Натискаючи на кнопку, ви даєте згоду на обробку персональних даних та погоджуєтесь з політикою конфіденційності

Рисунок 3.4 – Форма для зворотного зв'язку

3.4 Тестування вебзастосунку

Важливим способом перевірки адекватності та правильності функціонування будь-якого програмного забезпечення є його тестування. Загалом є дуже багато типів тестування.

Тестування програмного забезпечення завжди відповідає вимогам замовника, яких він хоче досягти. Тестування програмного забезпечення – це важливий процес, який використовується для підвищення якості програмного забезпечення та надійності застосунку. Важливо розуміти ключовий принцип тестування програмного забезпечення, який керує вами протягом усього процесу розробки програмного забезпечення.

Ці принципи допоможуть тестувальнику виявити проблему програмного забезпечення раніше та перевірити, чи відповідає програмне забезпечення очікуванням.

					КвРІПЗ.2101085.01.14.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		56

- Тестування показує наявність дефектів
- Вичерпне тестування неможливе
- Раннє тестування
- Кластеризація дефектів
- Парадокс пестицидів
- Тестування залежить від контексту
- Помилка відсутності помилок

В контексті розробки та тестування вебзастосунку доцільно використовувати юніт-тестування, АВ-тестування, ручне тестування.

Модульне тестування або юніт-тестування - це метод тестування окремих блоків або компонентів програмного застосунку. Зазвичай його виконують розробники, щоб переконатися, що окремі блоки програмного забезпечення працюють належним чином. Модульні тести зазвичай автоматизовані та призначені для тестування певних частин коду, таких як певна функція або метод. Модульне тестування проводиться на найнижчому рівні процесу розробки програмного забезпечення, де окремі блоки коду тестуються ізольовано.

Для даного тестування у розроблюваному застосунку було здійснено чи вірно та адекватно працює рендеринг головної сторінки, чи працює функціонал додавання товарів у корзину, чи можливо зробити та оформити замовлення туристичного спорядження, а також чи може користувач перевірити, що він добавив у корзину.

					КвРІПЗ.2101085.01.14.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		57

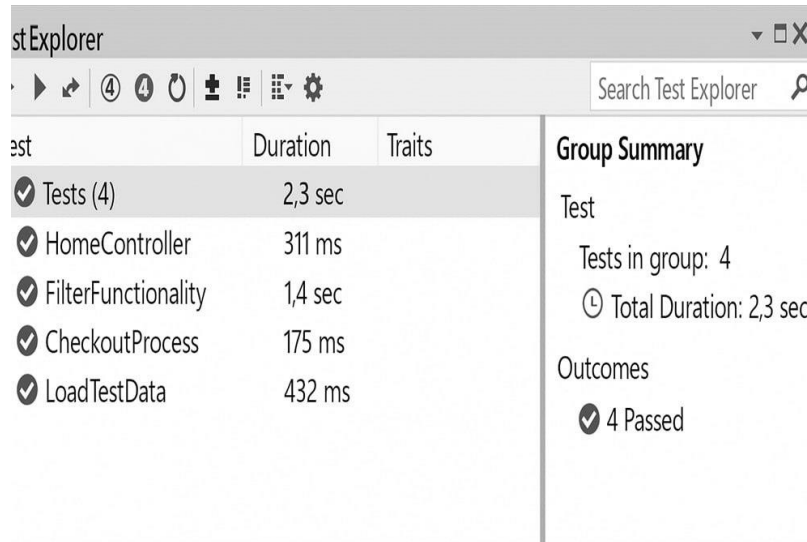


Рисунок 3.5 – Виведені результати юніт-тестування

На рисунку 3.5 подано варіант роботи юніт-тестів, що демонструє правильність роботи функціоналу та відсутність дефектів чи неточностей.

Висновки до 3-го розділу

Отже, у третьому розділі описано реалізацію бази даних, основних модулів вебзастосунку, в тому числі реалізацію інтерфейсу та подано інструкцію для користувача. Також проаналізовано методи тестування та показано результати тестування даного вебзастосунку.

ВИСНОВКИ

Отже, на основі отриманих результатів роботи можна зробити наступні висновки.

Метою кваліфікаційної роботи була розробка вебзастосунку для продажу туристичного спорядження, що має можливість реалізувати функціонал різних категорії користувачів.

Також потрібно було вирішити такі завдання задля досягнення поставленої мети:

– здійснити детальний аналіз предметної області, а саме онлайн-сервісів, платформ та загалом електронної комерції з продажу туристичного спорядження;

– проаналізувати існуючі на ринку програмно-технічні рішення для торгівлі товарів для туризму та супутніх товарів, туристичного спорядження, визначити ключові характеристики, недоліки та переваги цих ресурсів;

– проаналізувати та визначити вимоги до розроблюваного програмного продукту із виокремленням функціональних та нефункціональних вимог;

– на основі проведеного аналізу та визначення вимог до програмного забезпечення оформити необхідну документацію для розробки, а саме технічне завдання;

– здійснити аналіз типів архітектури та обрати відповідне архітектурне рішення для розроблюваного вебзастосунку;

– здійснити реалізацію з деталізованим описом програмного забезпечення, а саме вебзастосунку для торгівлі туристичним спорядженням;

– протестувати розроблений вебзастосунок для продажу туристичного спорядження.

Перед початком роботи було проаналізовано літературу з даної теми, виділено головні дослідження та практики, що стосуються проектування та розробки вебзастосунків різного призначення.

					КвРІПЗ.2101085.01.14.ПЗ	Арк.
						59
Зм.	Арк	№ докум.	Підпис	Дата		

У першому розділі було визначено актуальність даної розробки, визначено поняття вебзастосунку та його видів. Також проаналізовано предметну область, а саме з розробки вебзастосунків для торгівлі товарів туристичного призначення.

У другому розділі показано процес проектування модулів системи, архітектурного рішення, бази даних, інтерфейсу, а також огляд та аналіз існуючих технологій для веброзробки. В подальшому було здійснено вибір технологій для реалізації даного проекту.

У третьому розділі показано процес реалізації бази даних, основних модулів, що були спроектовані у другому розділі, подано інструкцію користувача та здійснено тестування вебзастосунку на базовому рівні.

У результаті написання кваліфікаційної роботи отримано вебзастосунок для торгівлі туристичним спорядженням, що дозволяє користуватись ним у будь-якій частині світу, де є доступ до мережі Інтернет та підключені відповідні платіжні системи та способи доставки, що працюють у регіоні.

					КвРІПЗ.2101085.01.14.ПЗ	Арк.
						60
Зм.	Арк	№ докум.	Підпис	Дата		

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Qahtan M. Yas. A Compr A Comprehensive Review of Software Development Life Cycle methodologies: Pros, Cons, and Future Directions. Iraqi Journal for Computer Science and Mathematics. - 2023.
2. Інтернет-ресурс з продажу Rozetka. URL: <https://rozetka.com.ua/> (дата звернення 12.02.2025).
3. Інтернет-ресурс з продажу olx.ua. URL: https://www.olx.ua/uk/?srsrtid=AfmBOorqQHCH9n9LyshmtTKxz_ZvYYvXyIFK1morvYvmdaYySObg6Z2s (дата звернення 12.02.2025).
4. Атаманчук П.С., Мендерецький В.В., Панчук О.П., Чорна О.Г. Інтегрований курс безпеки життєдіяльності (теоретичні основи): Навч. посіб. – Кам'янець-Подільський: Буйницький О .А., 2009. – 200 с.
5. Що таке CSS/CSS3? [Електронний ресурс] – Режим доступу до ресурсу: <https://artjoker.ua/tech/web-stack/csscss3/>
6. NUnit. URL: <https://nunit.org/> (дата звернення 19.11.2024).
7. What is SQL Server? URL: <https://learn.microsoft.com/en-us/sql/sql-server/what-is-sql-server?view=sql-server-ver16> (дата звернення 23.02.25).
8. Архітектура програмного забезпечення: все що треба знати – URL: <https://wezom.com.ua/ua/blog/arhitektura-programmn> (дата звернення 23.02.25).
9. What Information Do User Guides Have – Essential Elements of User Guides. URL: <https://www.proprofskb.com/blog/user-guide/> (дата звернення 03.02.25).
10. What is Unit Testing? URL: <https://aws.amazon.com/what-is/unit-testing/> (дата звернення 16.02.25).
11. Діаграма зв'язків сутностей URL: <https://www.mindonmap.com/uk/blog/relationship-diagram/> (дата звернення 25.02.25).

					КвРІПЗ.2101085.01.14.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		61

12. Use Case. URL: <https://training.qatestlab.com/blog/technical-articles/what-is-a-use-case-and-what-are-they-for/> (дата звернення 31.03.25).

13. Тестування програмного забезпечення: типи, види та застосування. URL: <https://foxminded.ua/testuvannia-prohramnoho-zabezpechennia/> (дата звернення 06.04.25).

14. Що таке ASP.NET Core, URL: <https://tqm.com.ua/ua/likbez/ua-articles/chomu-asp-net> (дата звернення 06.04.25).

15. Що таке ASP.NET і для чого використовується? URL: <https://hyperhost.ua/info/uk/shho-take-aspnet-i-dlya-cogo-vikoristovujetsya> (дата звернення 06.04.25).

16. Діаграма послідовності (Sequence Diagrams). URL: <https://www.maxzosim.com/sequence-diagrams/> (дата звернення 10.04.25).

17. Software Design Basics. URL: https://www.tutorialspoint.com/software_design_basics (дата звернення 10.04.2025).

18. MVC Framework – Introduction. URL: https://www.tutorialspoint.com/mvc_framework/mvc_framework_introduction.htm (дата звернення 28.04.25).

19. The Software Architect: Demystifying 18 Software Architecture Patterns. URL: <https://medium.com/@amitvsolutions/the-software-architect-deep-dive-into-17-essential-design-patterns-a0fb5a4726ab> (дата звернення 30.04.25).

20. What is Software Architecture in Software Engineering? URL: <https://www.computer.org/resources/software-architecture> (дата звернення 05.05.25).

21. Software Testing – What is it? Everything to Know. URL: <https://www.globalapptesting.com/blog/software-testing> (дата звернення 06.05.2025).

22. ASP.NET Core Documentation. URL: <https://dotnet.microsoft.com/en-us/apps/aspnet> (дата звернення 23.05.25).

					КвРІПЗ.2101085.01.14.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		62

23. MVC. URL: <https://brander.ua/technologies/mvc> (дата звернення 23.05.25).

24. Graph Neural Network for Source Code Defect Prediction. URL: <http://ieeexplore.ieee.org/abstract/document/9684879> (дата звернення 23.05.25).

25. Utilizing Static Analysis and Code Generation to Accelerate Neural Networks. URL: <https://arxiv.org/abs/1206.6466> (дата звернення 23.05.25).

26. Severity Classification of Code Smells Using Machine-Learning Methods. URL: <https://ouci.dntb.gov.ua/en/works/4MYnYrx4/> (дата звернення 23.05.25).

27. ДСТУ 8302:2015. Бібліографічне посилання. Загальні положення та правила складання. Київ : ДП «УкрНДНЦ», 2016. 17 с.

28. Кваліфікаційна робота : Методичні настанови для здобувачів першого (бакалаврського) рівня вищої освіти спеціальності 121 «Інженерія програмного забезпечення» / Л. П. Бедратюк, Г. І. Радельчук. Хмельницький : ХНУ, 2023. 60 с.

29. Tomohiro S., Yutaka W. Learning Path Recommendation System for Programming Education Based on Neural Networks. *International Journal of Distance Education Technologies (IJDET)*. 2020. № 18(1). P.29-44.

30. Інтернет-магазин туристичного спорядження Adventure. URL: <https://www.adventurer.com.ua/> (дата звернення 23.05.25).

31. Client Server Architecture Explained. URL: <https://net2connect.unaux.com/client-server-architecture-explained/?i=1> (дата звернення 23.05.25).

32. Основні типи архітектури програмного забезпечення. URL: <https://www.artofba.com/uk/post/main-types-of-software-architecture> (дата звернення 23.05.25).

					КвРІПЗ.2101085.01.14.ПЗ	Арк.
						63
Зм.	Арк	№ докум.	Підпис	Дата		

Додаток А
(обов'язковий)

ПРЕЗЕНТАЦІЙНІ МАТЕРІАЛИ

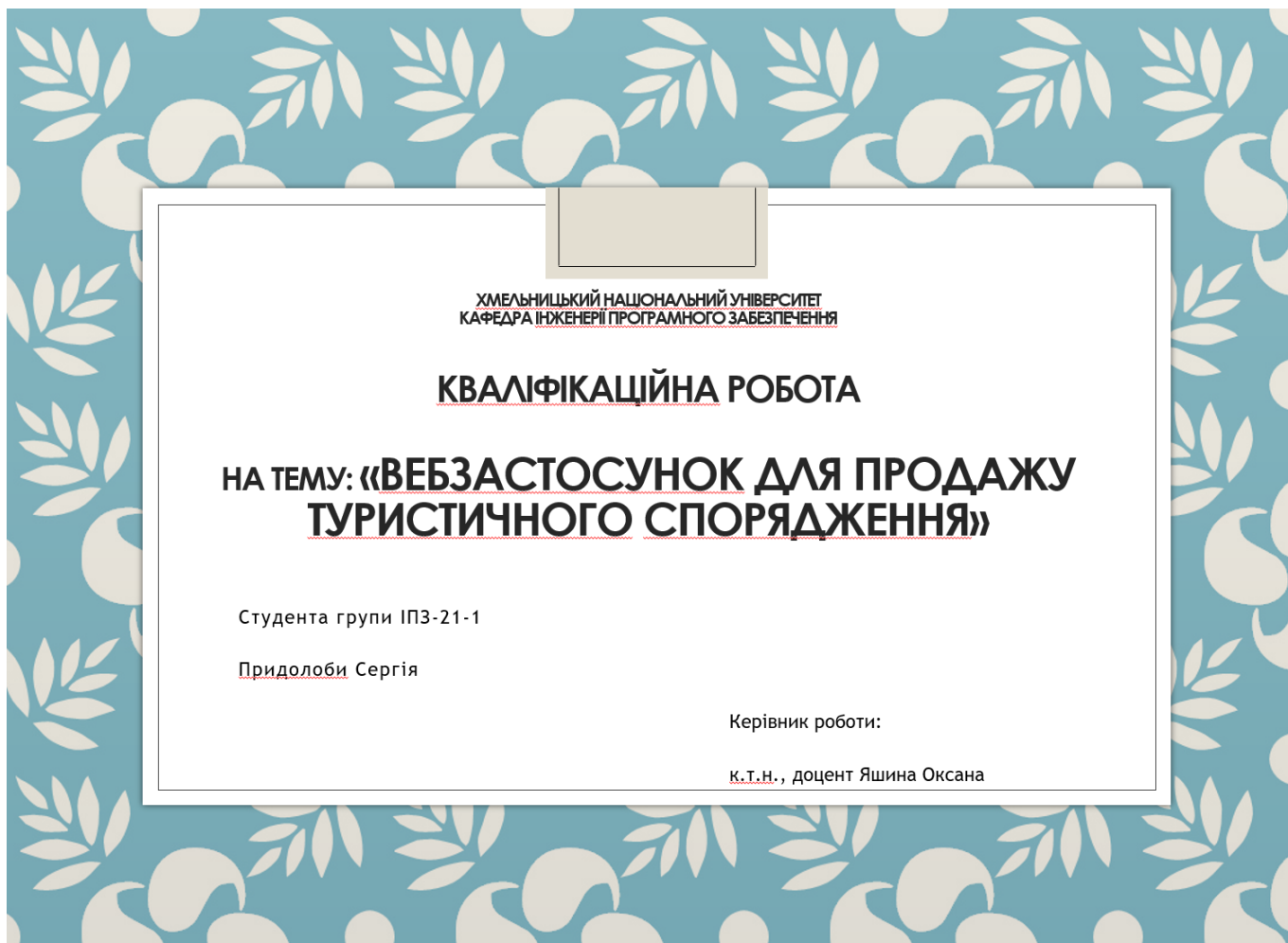


Рисунок А.1 – Титульний слайд, представлення

Мета та завдання кваліфікаційної роботи

Мета роботи: визначення вимог, проектування та розроблення вебзастосунку для продажу туристичного спорядження.

Задачі:

- здійснити аналіз предметної області;
- зробити аналіз існуючих актуальних рішень з даної предметної області;
- вказати переваги та недоліки існуючих рішень;
- визначити функціональні та нефункціональні вимоги;
- розробити технічне завдання;
- здійснити проектування архітектури, бази даних проекту;
- здійснити виконання програмної реалізації;
- протестувати розроблений продукт.

Рисунок А.2 – Мета і завдання кваліфікаційної роботи

АКТУАЛЬНІСТЬ

3

- Пандемія COVID-19 та воєнні дії в Україні зумовили динамічні зміни в моделях споживчої поведінки при здійсненні покупок. Окрім того, нестабільна ситуація у сфері охорони здоров'я може створювати ризики для безпеки користувачів, які відвідують фізичні магазини для вибору товарів.
- Враховуючи цей контекст, створення онлайн-бізнесу туристичного спорядження є розумним кроком, оскільки це безпечний та ефективний спосіб охопити покупців.
- Окрім практичних міркувань безпеки та ефективності, онлайн роздрібна торгівля створює для людей можливість відкрити для себе бренди, з якими вони б ніколи не зіткнулися в іншому місці.
- Продавець має можливість достукатися до клієнтів у будь-якій точці світу, незалежно від того, де вони знаходяться.

Рисунок А.3 – Актуальність теми

Предметна область

Розроблюване програмне забезпечення створюється для здійснення автоматизації у режимі ручного вибору туристичного спорядження, а також замовлення із можливістю оплати у режимі онлайн.

Предметною областю даного дослідження є торгівля туристичним спорядженням та супутніми товарами за допомогою вебзастосунку в мережі Інтернет.

Рисунок А.4 – Предметна область

Аналіз стану інших рішень

При порівнянні різних вебресурсів з торгівлі туристичним спорядженням увага націлена на такі чинники:

- робота застосунку, а саме зручність користування, доцільність та продуманість навігації, розміщення різноманітних елементів (кнопок переходу, пунктів меню, кнопок заклику до дій тощо);
- наявний асортимент – наскільки широко пропонується покупцеві товари з даної сфери бізнесу;
- ціни – порівняння вартості товарів з конкурентами, наявність спецпропозицій;
- підтримка та консультації – швидкість відповідей на звернення покупця, опрацювання замовлення, допомога у вирішенні незрозумілих моментів та конфліктних ситуацій;
- зручність оплати – великий вибір способів оплати (можливість розтермінування та кредиту);
- доставка – швидкість опрацювання, кількість способів доставки;
- акції, знижки, розпродажі – можливість купити дешевше в мережі інтернет;
- гарантійні зобов'язання – повернення товару, ремонт та обслуговування.

Рисунок А.5 – Аналіз стану існуючих рішень

Порівняльна таблиця існуючих застосунків

Сайт	Спеціалізація	Переваги	Недоліки
OLX.ua	Оголошення	Велика аудиторія, простота, вживане спорядження	Шахрайство, відсутність API, ручне керування доставкою
Prom.ua	Маркетплейс	Інструменти продажу, міні-сайт, велика аудиторія	Комісія, конкуренція, обмежена кастомізація
Epicentrk.ua	Гіпермаркет	Довіра, фізичні магазини, доставка	Не спеціалізується, складний сайт, мало інтеграцій
Rozetka.ua	Універсальний магазин	Великий вибір, ціни, доставка, відгуки	Не завжди вузький асортимент, складний пошук
Gorgany.com	Туризм	Спеціалізація, фахівці, якість, магазини	Вищі ціни, менший вибір
Alp.com.ua	Альпінізм, туризм	Вузька спеціалізація, якість, консультації	Високі ціни, мало товарів для загального туризму
Adventure.com.ua	Активний відпочинок	Великий вибір, бренди, зручний сайт	Вищі ціни, не весь асортимент у наявності

Рисунок А.6 – Порівняльна таблиця існуючих застосунків

Аналіз вимог до вебресурсу

7

Функціональні вимоги

Можливість реєстрації користувача з присвоєнням ролі (покупець, адміністратор)

Користувач без реєстрації — гість з обмеженим доступом

Розробка структури вебзастосунок та інтерфейсу користувача

Наявність сторінки каталогу товарів

Можливість збереження товарів у кошику до оформлення замовлення

Наявність можливості зворотного зв'язку з адміністрацією

Перевірка валідності введених даних

Нефункціональні вимоги

Вебзастосунок має бути доступним 24/7

Інтерфейс повинен бути зручним і адаптивним для мобільних і десктопів

Забезпечення безпеки персональних даних користувачів

Швидкість завантаження сторінок не більше 3 секунд

Підтримка масштабованості при збільшенні користувачів і товарів

Рисунок А.7 – Аналіз вимог до вебресурсу

Діаграми варіантів використання для користувача та адміна

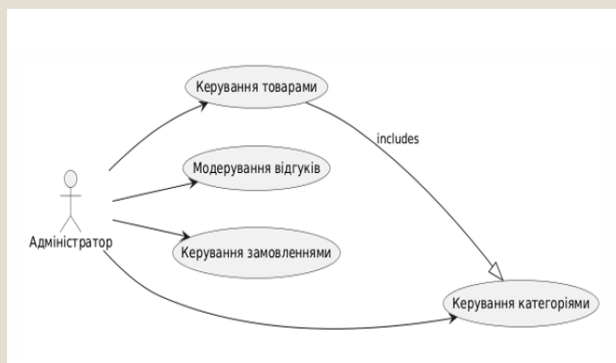


Рисунок А.8 – Діаграми варіантів використання для користувача та адміна

Діаграма декомпозиції для процесу замовлення

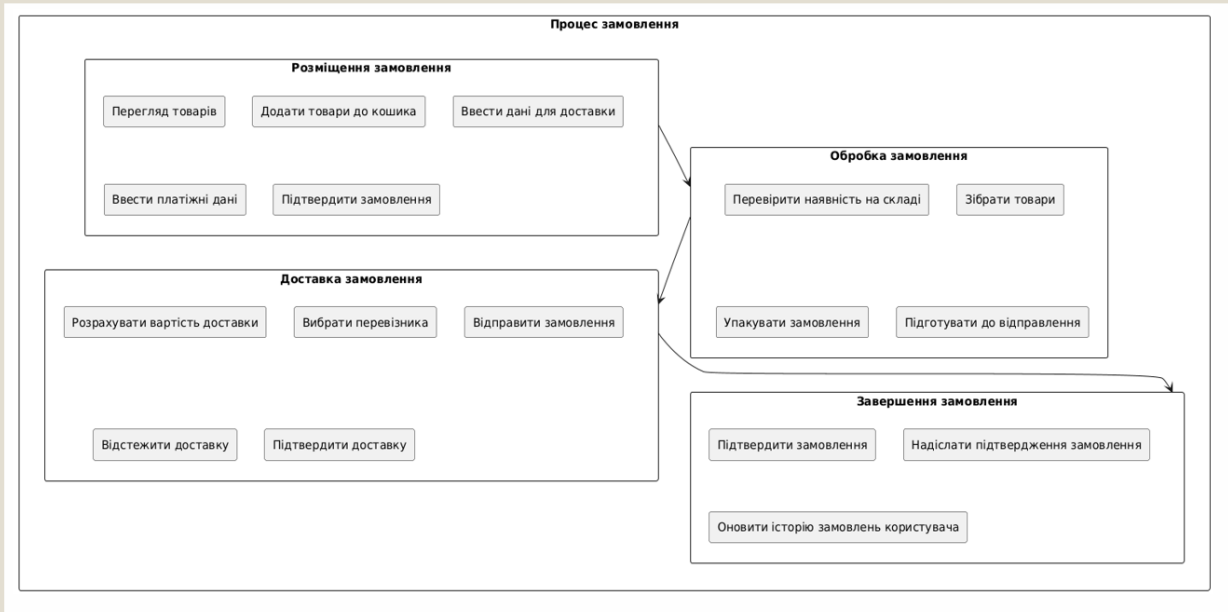


Рисунок А.9 – Діаграма декомпозиції для процесу замовлення

Архітектура вебсайту

Клієнт-серверна архітектура — це модель, де клієнт надсилає запити, а сервер їх обробляє і повертає відповіді. Вона розділяє функції між користувачем і системою для ефективної роботи.

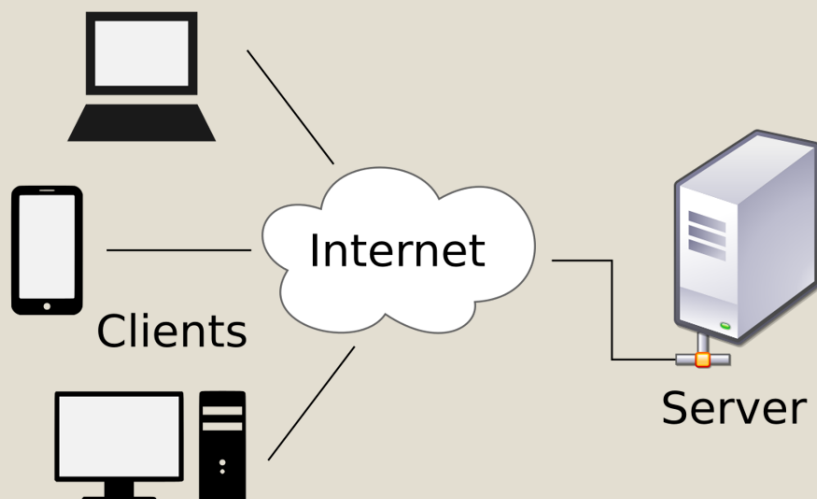


Рисунок А.10 – Архітектура вебсайту

Модель бази даних

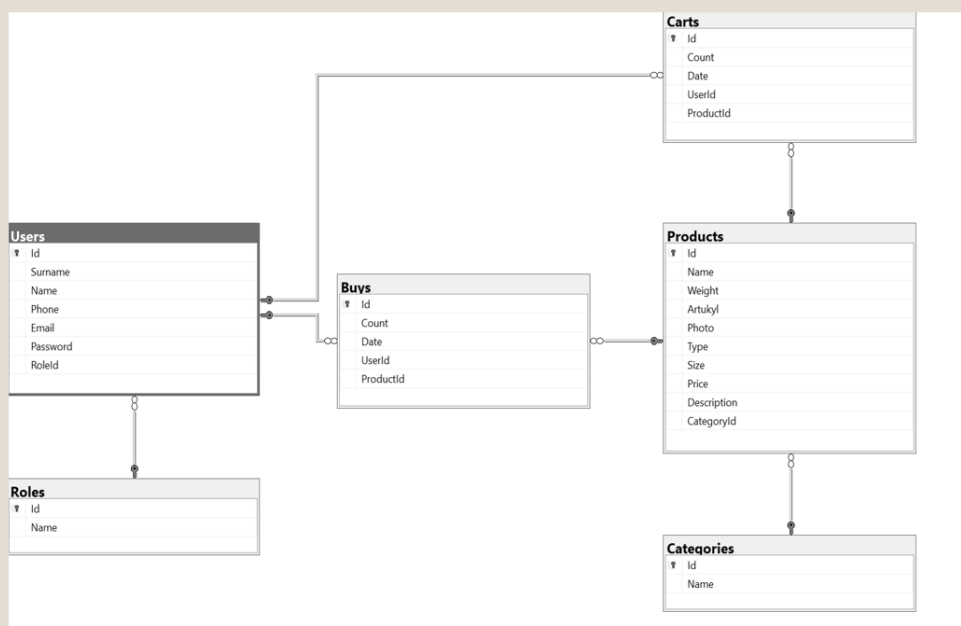


Рисунок А.11 – Модель бази даних

Реалізація модулів і база даних



ASP.NET - це фреймворк з відкритим вихідним кодом, який багато веброзробників використовують для створення сучасних вебзастосунків.



Bootstrap - фреймворк CSS з відкритим кодом. Він містить шаблони дизайну на основі HTML, CSS та JavaScript для типографіки, форм, кнопок, навігації та інших компонентів інтерфейсу.

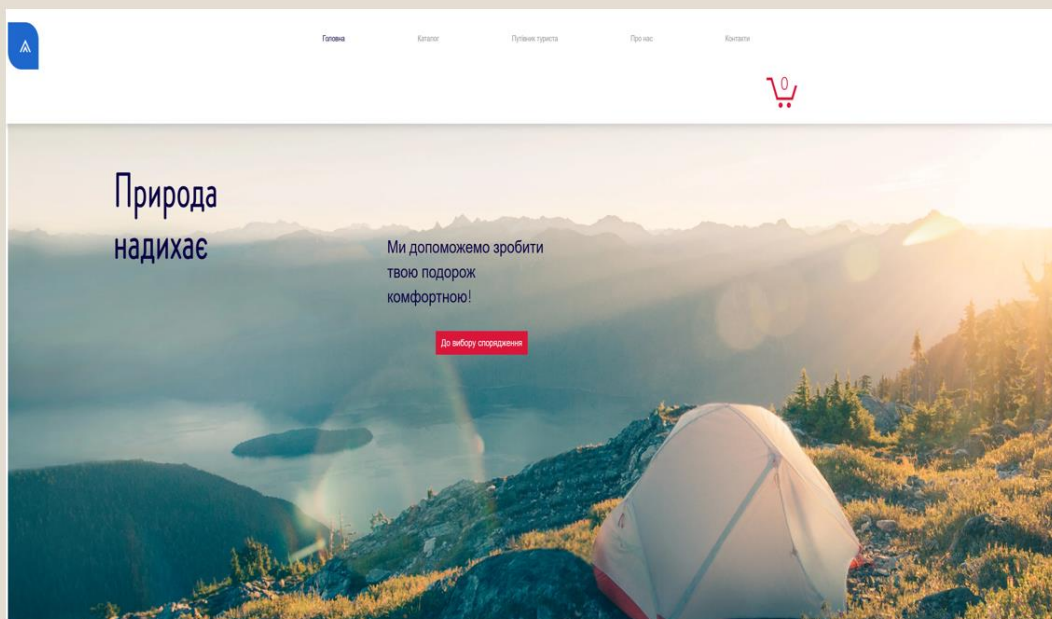


MS SQL Server – система управління базами даних від компанії Microsoft, що використовується для зберігання та обробки інформації.

Рисунок А.12 – Реалізація модулів і база даних

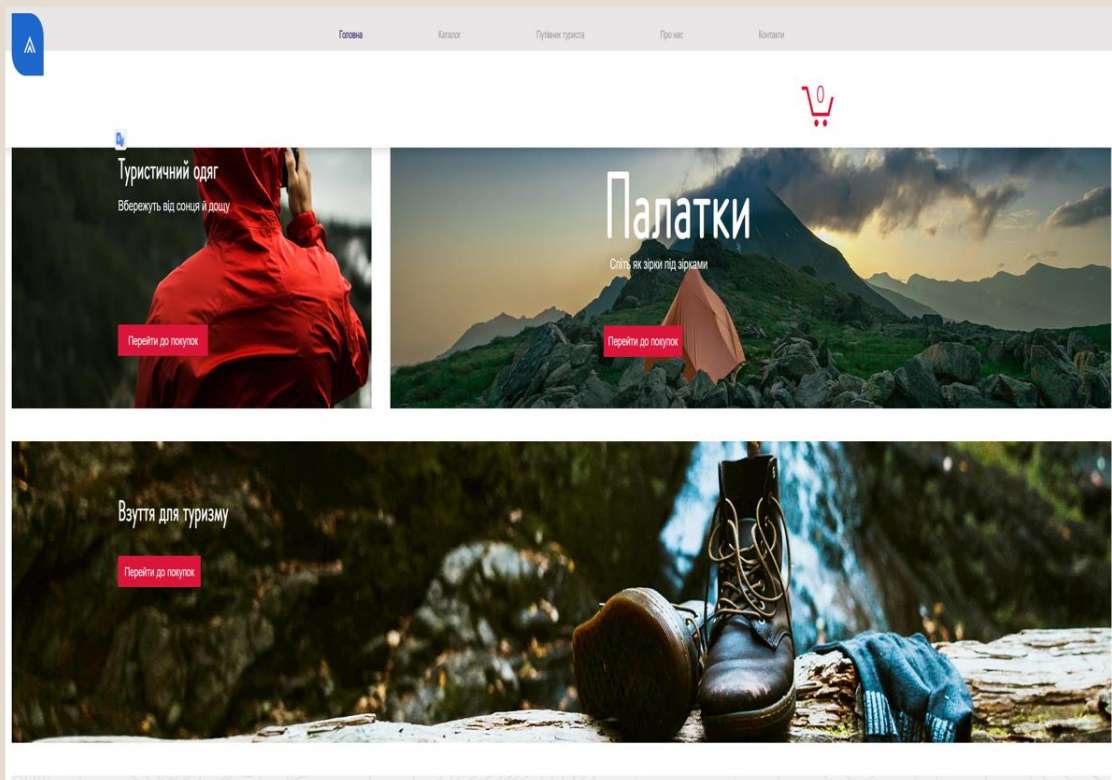
Результати роботи

13



Головна сторінка

Рисунок А.13 – Результати роботи головної сторінки



Сторінка каталогу

Рисунок А.14 – Результати роботи сторінки каталогу

Вимоги до технічного та програмного забезпечення

Параметр	Мінімальні вимоги
Браузер	Microsoft Edge, Google Chrome, Mozilla Firefox, Safari та інші сучасні браузери
Дисплей	Мінімальна роздільна здатність 1024×768 пікселів
Процесор	Щонайменше 1,6 ГГц (рекомендовано 2,0 ГГц або вище для стабільної роботи)
Оперативна пам'ять	Від 1 ГБ (рекомендовано 2 ГБ і більше для комфортної взаємодії)
Місце на диску	Мінімум 100 МБ для збереження кешу та тимчасових файлів браузера
Операційна система	Windows 7/8/10/11, macOS, Linux, Android, iOS (для мобільної версії сайту)
Підключення до Інтернету	Стабільне підключення зі швидкістю від 512 Кбіт/с (рекомендовано 1 Мбіт/с і вище)
JavaScript	Увімкнений JavaScript для роботи інтерактивних елементів
Cookies	Дозвіл на збереження файлів cookies для персоналізації та збереження сесій

Рисунок А.15 – Вимоги до технічного та програмного забезпечення

Тестування

16

Було проведено функціональне ручне тестування, перевірка валідації форм та UI-тестування

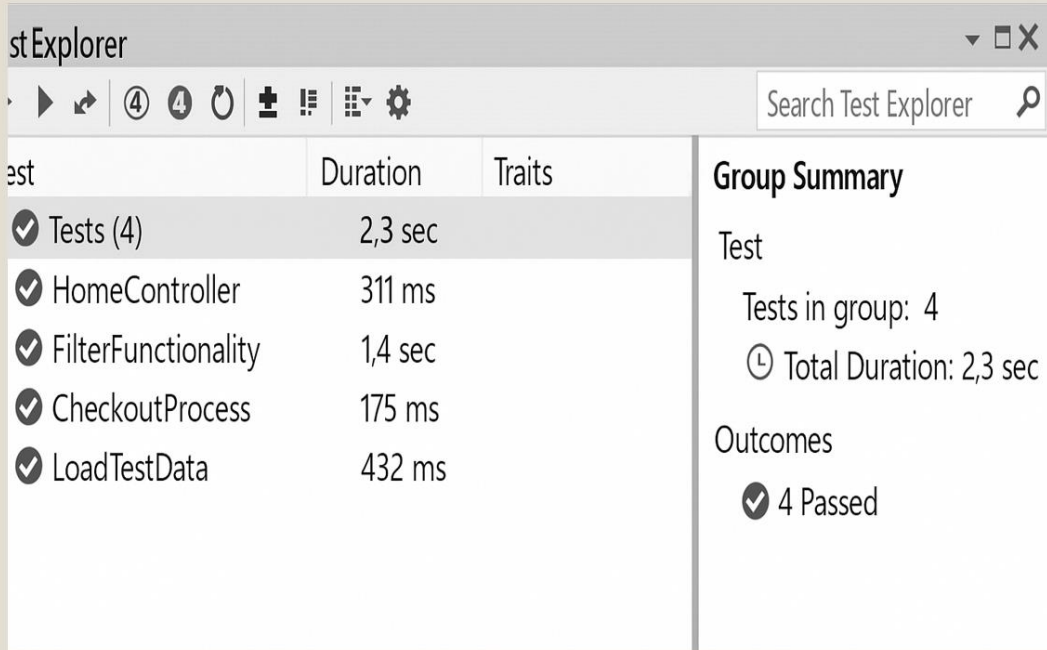


Рисунок А.16 – Тестування

ВИСНОВКИ

У процесі виконання кваліфікаційної роботи було проведено повний цикл розробки програмного забезпечення, зокрема: здійснено аналіз предметної області,

- здійснено детальний аналіз предметної області, що охоплює існуючі рішення на ринку та особливості електронної комерції у сфері туристичного спорядження;
- сформульовано функціональні та нефункціональні вимоги до майбутнього вебзастосунок з урахуванням потреб цільової аудиторії; виконано проєктування архітектури системи, структури бази даних, користувацького інтерфейсу та логіки взаємодії;
- реалізовано вебзастосунок для онлайн-торгівлі туристичним спорядженням з адаптивним дизайном, підтримкою особистих кабінетів, пошуку товарів, оформлення замовлень, системою відгуків і інтеграцією з платіжними сервісами.

Рисунок А.17 – Висновки



Дякую за увагу!

Рисунок А.18 – Останній слайд

Додаток Б
(обов'язковий)

ТЕХНІЧНЕ ЗАВДАННЯ

Вступ

Робота виконується в рамках кваліфікаційної роботи на тему «Вебзастосунок для торгівлі туристичним спорядженням».

1 Підстава для розробки

Підставою для розробки є «Завдання на кваліфікаційну роботу», що затверджене завідувачем кафедри інженерії програмного забезпечення.

Найменування розробки: Вебзастосунок для торгівлі туристичним спорядженням.

2 Призначення розробки

Вебзастосунок для торгівлі туристичним спорядженням призначений для забезпечення користувачів можливістю швидкого пошуку, огляду та купівлі товарів для туризму, зокрема палаток, мішків, карематів тощо в будь-якій точці світу та з будь-якої точки світу, не переживаючи за свою безпеку та безпеку своїх даних.

Користувачами системи є звичайні люди, що люблять подорожувати самостійно або в компанії.

Пошук необхідного товару здійснюється шляхом вказання безпосередньої адреси ресурсу (коли SEO спрацювала добре та ресурс достатньо знає людей) або ж коли вказуються параметри як фірма-виробник, розмір та колір, а також діапазон бажаної ціни, технічні характеристики. Що стосується палаток, то достатнім є вказання такого параметру як розмір, колір, виробник.

Також передбачено форму для зворотного зв'язку, можливість добавляти товар у кошик, здійснювати замовлення, обирати спосіб оплати та доставки

Щодо експлуатаційного призначення, система може використовуватися на будь-якому персональному комп'ютері, у будь-якому встановленому браузері, ніяких додаткових налаштувань не потребує.

3 Вимоги до програми

3.1 Вимоги до функціональних характеристик

Вебзастосунок для продажу туристичного спорядження повинен мати можливість:

- публікації, редагування та видалення інформації щодо продукції, а саме туристичного спорядження;

- зручного пошуку у категоріях продукції;

- авторизації та реєстрації користувачів;

- налаштування особистого кабінету та деактивації профілю;

- керування вебзастосунком адміністратором;

Вимоги до інтерфейсу:

- простий, сучасний та зрозумілий інтерфейс користувача;

- актуальна інформація щодо товарів.

3.2 Вимоги до надійності

Розроблюване ПЗ повинно мати:

- можливість самовідновлення після збоїв, таких як відключення електроживлення;

- обмеження несанкціонованого доступу до даних;

- розмежування прав користувачів у системі;

- контроль інформації, що вводиться, та блокування некоректних дій користувача при роботі з системою.

3.3 Вимоги до складу та параметрів технічних засобів

Мінімальні вимоги для функціонування системи повинні відповідати вимогам будь-якого браузера. Нижче наведено приклад мінімальних вимог для браузера Google Chrome:

- стабільне з'єднання з Інтернетом;

- операційна система: Windows 10, 8, 8.1, 7;

- розрядність: x86 (32-bit) або x64 (64-bit);
- процесор: Pentium 4 з SSE2;
- вінчестер: 350 Mb;
- відеоадаптер: 3D адаптер nVidia, Intel, AMD / ATI;
- оперативна пам'ять: 512 Mb;
- будь-яка аудіокарта;
- роздільна здатність екрану: SVGA 800x600;
- контролери: клавіатура, миша.

3.4 Вимоги до інформаційної та програмної сумісності

Для створення серверної частини будуть використовуватися такі технології:

- об'єктно-орієнтована мова програмування C#;
- фреймворк ASP.NET Core;
- патерн проектування Model-View-Controller;
- технологія доступу та керування реляційними даними, та перетворення їх

у об'єктно-орієнтовану структуру Entity Framework Core.

Для створення клієнтської частини використовуватимуться такі технології:

- мова розмітки HTML;
- таблиці стилів CSS;
- мова програмування JavaScript.

Для роботи з базою даних буде використовуватися мова SQL та система керування базами даних SQL Server.

3.5 Вимоги до транспортування та зберігання

Програма, її документація поставляються у цифровому вигляді. Умови експлуатації програмного забезпечення збігаються з умовами експлуатації серверу, на якому буде розміщене ПЗ.

3.6 Спеціальні вимоги

Програма повинна мати дружній інтерфейс, розрахований на користувача середньої кваліфікації (з точки зору комп'ютерної грамотності). Мова програмування визначається вибором виконавця.

4 Вимоги до програмної документації

В ході розробки програми повинні бути підготовлені:

- технічне завдання;
- опис програми;
- текст програми з коментарями та поясненнями;
- методика випробувань;
- відомості про функціонування програми;
- керівництво користувача;
- керівництво програміста.

5 Стадії та етапи розробки

Розробка програмного продукту проходить декілька стадій і етапів:

- розробка технічного завдання;
- створення ескізного проекту;
- створення технічного проекту;
- створення робочого проекту;
- розробка програмної документації;
- проведення тестування онлайн-ресурсу;
- впровадження.

6 Порядок контролю та приймання

Контроль здійснюється кінцевими користувачами вебзастосунком, підключеними на етапі тестування. Прийом вебзастосунку здійснюється після його повного розгортання та встановлення на хостинг і налаштування для нормального функціонування.

Після закінчення розробки вебзастосунку повинні бути проведені тестування на захист від некоректного введення.

Додаток В

ЛІСТИНГ КОДУ

Конфігурація та Додавання Сервісів (Program.cs)

```
// Program.cs (для .NET 6+), або Startup.cs (для старіших версій)

using Microsoft.EntityFrameworkCore;
using YourProjectName.Data; // Простір імен для вашого DbContext
using YourProjectName.Models; // Простір імен для ваших моделей даних
using Microsoft.AspNetCore.Identity;

var builder = WebApplication.CreateBuilder(args);

// Додаємо DbContext для роботи з базою даних
var connectionString =
builder.Configuration.GetConnectionString("DefaultConnection") ?? throw new
InvalidOperationException("Connection string 'DefaultConnection' not
found.");
builder.Services.AddDbContext<ApplicationDbContext>(options =>
options.UseSqlServer(connectionString)); // Використовуємо SQL
Server, можна змінити на UsePostgreSQL, UseMySQL

// Додаємо ASP.NET Core Identity
builder.Services.AddDefaultIdentity<IdentityUser>(options =>
options.SignIn.RequireConfirmedAccount = true)
.AddRoles<IdentityRole>() // Дозволяє використовувати ролі
.AddEntityFrameworkStores<ApplicationDbContext>(); //
Використовуємо наш DbContext для зберігання Identity даних

// Додаємо підтримку контролерів та представлень (MVC)
builder.Services.AddControllersWithViews();

// Додаємо Razor Pages (якщо використовуєте для Identity UI або інших
сторінок)
builder.Services.AddRazorPages();

var app = builder.Build();

// Налаштування HTTP запитів (middleware pipeline)
if (app.Environment.IsDevelopment())
```

```

    {
        app.UseMigrationsEndPoint(); // Для відображення помилок міграції в
розробці
    }
else
{
    app.UseExceptionHandler("/Home/Error");
    app.UseHsts();
}

app.UseHttpsRedirection();
app.UseStaticFiles();

app.UseRouting();

// Аутентифікація та Авторизація
app.UseAuthentication();
app.UseAuthorization();

// Мапінг контролерів
app.MapControllerRoute(
    name: "default",
    pattern: "{controller=Home}/{action=Index}/{id?}");

// Мапінг Razor Pages (для Identity UI)
app.MapRazorPages();

app.Run();

```

Моделі Даних (Models)

```

// Models/Product.cs
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

namespace YourProjectName.Models
{
    public class Product
    {
        public int Id { get; set; }

        [Required(ErrorMessage = "Назва товару обов'язкова.")]

```

```

        [StringLength(200, MinimumLength = 3, ErrorMessage = "Назва
повинна бути від 3 до 200 символів.")]
        public string Name { get; set; }

        public string Description { get; set; }

        [Range(0.01, 100000.00, ErrorMessage = "Ціна повинна бути між
0.01 та 100000.00.")]
        [Column(TypeName = "decimal(18, 2)")] // Вказуємо точність для
десятькового типу
        public decimal Price { get; set; }

        [Range(0, 10000, ErrorMessage = "Кількість на складі повинна
бути невід'ємною.")]
        public int StockQuantity { get; set; }

        [Display(Name = "Категорія")]
        public int CategoryId { get; set; }
        public Category Category { get; set; } // Навігаційна властивість

        [Display(Name = "Виробник")]
        public int ManufacturerId { get; set; }
        public Manufacturer Manufacturer { get; set; } // Навігаційна
властивість

        [Display(Name = "Зображення")]
        public string imageUrl { get; set; } // URL до зображення

        public bool IsActive { get; set; } = true; // Чи активний товар

        // Колекція відгуків
        public ICollection<Review> Reviews { get; set; } = new
List<Review>();
    }
}

// Models/Category.cs
using System.ComponentModel.DataAnnotations;

namespace YourProjectName.Models
{
    public class Category
    {

```

```

public int Id { get; set; }

[Required(ErrorMessage = "Назва категорії обов'язкова.")]
[StringLength(100, MinimumLength = 2, ErrorMessage = "Назва
повинна бути від 2 до 100 символів.")]
public string Name { get; set; }

public string Description { get; set; }

public int? ParentCategoryId { get; set; } // Для ієрархічних
категорій
public Category ParentCategory { get; set; } // Навігаційна
властивість

public ICollection<Category> SubCategories { get; set; } = new
List<Category>();
public ICollection<Product> Products { get; set; } = new
List<Product>();
}
}

// Models/Manufacturer.cs
using System.ComponentModel.DataAnnotations;

namespace YourProjectName.Models
{
    public class Manufacturer
    {
        public int Id { get; set; }

        [Required(ErrorMessage = "Назва виробника обов'язкова.")]
        [StringLength(100, MinimumLength = 2, ErrorMessage = "Назва
повинна бути від 2 до 100 символів.")]
        public string Name { get; set; }

        public string CountryOfOrigin { get; set; }
        public string Website { get; set; }

        public ICollection<Product> Products { get; set; } = new
List<Product>();
    }
}

```

```

// Models/Review.cs
using System.ComponentModel.DataAnnotations;

namespace YourProjectName.Models
{
    public class Review
    {
        public int Id { get; set; }

        public int ProductId { get; set; }
        public Product Product { get; set; }

        public string UserId { get; set; } // IdentityUser.Id - string!
        // public IdentityUser User { get; set; } // Можна додати
        навігаційну властивість до IdentityUser, якщо потрібно

        [Range(1, 5, ErrorMessage = "Рейтинг повинен бути від 1 до 5.")]
        public int Rating { get; set; }

        [StringLength(1000, ErrorMessage = "Коментар не може
        перевищувати 1000 символів.")]
        public string Comment { get; set; }

        public DateTime ReviewDate { get; set; } = DateTime.UtcNow;
    }
}

```

DbContext (Data/ApplicationDbContext.cs)

```

// Data/ApplicationDbContext.cs
using Microsoft.AspNetCore.Identity.EntityFrameworkCore;
using Microsoft.EntityFrameworkCore;
using YourProjectName.Models;

namespace YourProjectName.Data
{
    // Успадковуємося від IdentityDbContext, щоб отримати Identity таблиці
    public class ApplicationDbContext : IdentityDbContext
    {
        public ApplicationDbContext(DbContextOptions<ApplicationDbContext> options)
    }
}

```

```

        : base(options)
    {
    }

    public DbSet<Product> Products { get; set; }
    public DbSet<Category> Categories { get; set; }
    public DbSet<Manufacturer> Manufacturers { get; set; }
    public DbSet<Review> Reviews { get; set; }
    // ... Додайте інші DbSet'и для Order, OrderItem, ShoppingCart тощо

    protected override void OnModelCreating(ModelBuilder modelBuilder)
    {
        base.OnModelCreating(modelBuilder); // Важливо для IdentityDbContext

        // Приклад налаштування Many-to-Many через проміжну таблицю (для
OrderItems)
        // (Якщо у вас є Order та OrderItem моделі)
        // modelBuilder.Entity<OrderItem>()
        //     .HasKey(oi => new { oi.OrderId, oi.ProductId }); // Композитний
ключ

        // Налаштування для ієрархічних категорій
        modelBuilder.Entity<Category>()
            .HasOne(c => c.ParentCategory)
            .WithMany(c => c.SubCategories)
            .HasForeignKey(c => c.ParentCategoryId)
            .IsRequired(false) // Батьківська категорія не обов'язкова
            .OnDelete(DeleteBehavior.Restrict); // Запобігаємо каскадному
видаленню батьківської категорії при видаленні дочірньої

        // Налаштування для Reviews
        modelBuilder.Entity<Review>()
            .HasOne(r => r.Product)
            .WithMany(p => p.Reviews)
            .HasForeignKey(r => r.ProductId)
            .OnDelete(DeleteBehavior.Cascade); // Видаляти відгуки при видаленні
товару

```

```

// Індекси для прискорення запитів
modelBuilder.Entity<Product>()
    .HasIndex(p => p.CategoryId);
modelBuilder.Entity<Product>()
    .HasIndex(p => p.ManufacturerId);
modelBuilder.Entity<Product>()
    .HasIndex(p => p.Name); // Для пошуку
modelBuilder.Entity<Review>()
    .HasIndex(r => r.ProductId);
    }
}
}

```

Контролер для Управління Каталогом (ProductsController)

// Controllers/ProductsController.cs

```

using Microsoft.AspNetCore.Mvc;
using Microsoft.EntityFrameworkCore;
using YourProjectName.Data;
using YourProjectName.Models;
using Microsoft.AspNetCore.Authorization; // Для авторизації
using System.IO; // Для роботи з файлами (завантаження зображень)

```

namespace YourProjectName.Controllers

```

{
    public class ProductsController : Controller
    {
        private readonly ApplicationDbContext _context;
        private readonly IWebHostEnvironment _webHostEnvironment; // Для доступу до
        шляхів файлів

        public ProductsController(ApplicationDbContext context, IWebHostEnvironment
        webHostEnvironment)
        {
            _context = context;
            _webHostEnvironment = webHostEnvironment;
        }
    }
}

```

```

// GET: /Products (Список товарів для користувачів)

public async Task<IActionResult> Index(int? categoryId, string searchTerm,
int pageNumber = 1, int pageSize = 10)
{
    IQueryable<Product> products = _context.Products
        .Include(p => p.Category)
        .Include(p => p.Manufacturer)
        .Where(p => p.IsActive); // Показуємо тільки активні товари

    if (categoryId.HasValue)
    {
        products = products.Where(p => p.CategoryId == categoryId.Value);
    }

    if (!string.IsNullOrEmpty(searchTerm))
    {
        products = products.Where(p => p.Name.Contains(searchTerm) ||
p.Description.Contains(searchTerm));
    }

    // Пагінація
    var totalProducts = await products.CountAsync();
    var pagedProducts = await products
        .Skip((pageNumber - 1) * pageSize)
        .Take(pageSize)
        .ToListAsync();

    ViewBag.Categories = await _context.Categories.ToListAsync();
    ViewBag.CurrentCategory = categoryId;
    ViewBag.SearchTerm = searchTerm;
    ViewBag.PageNumber = pageNumber;
    ViewBag.PageSize = pageSize;
    ViewBag.TotalPages = (int)Math.Ceiling(totalProducts / (double)pageSize);

    return View(pagedProducts);
}

```

```

// GET: /Products/Details/5 (Детальна сторінка товару)
public async Task<IActionResult> Details(int? id)
{
    if (id == null)
    {
        return NotFound();
    }

    var product = await _context.Products
        .Include(p => p.Category)
        .Include(p => p.Manufacturer)
        .Include(p => p.Reviews) // Завантажуємо відгуки
        .FirstOrDefaultAsync(m => m.Id == id);

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

    return View(product);
}

// === Адмін-панель для управління товарами ===

// GET: /Products/AdminIndex (Список товарів для адмінів)
[Authorize(Roles = "Admin")] // Тільки для адміністраторів
public async Task<IActionResult> AdminIndex()
{
    var products = await _context.Products
        .Include(p => p.Category)
        .Include(p => p.Manufacturer)
        .ToListAsync();

    return View(products);
}

// GET: /Products/Create

```

```

[Authorize(Roles = "Admin")]
public IActionResult Create()
{
    ViewBag.Categories = _context.Categories.ToList();
    ViewBag.Manufacturers = _context.Manufacturers.ToList();
    return View();
}

// POST: /Products/Create
[HttpPost]
[ValidateAntiForgeryToken] // Захист від CSRF атак
[Authorize(Roles = "Admin")]
public async Task<IActionResult>
Create([Bind("Id,Name,Description,Price,StockQuantity,CategoryId,ManufacturerId,IsActive")] Product product, IFormFile imageFile)
{
    if (ModelState.IsValid)
    {
        if (imageFile != null && imageFile.Length > 0)
        {
            // Зберігаємо зображення
            string wwwRootPath = _webHostEnvironment.WebRootPath;
            string fileName = Guid.NewGuid().ToString() +
Path.GetExtension(imageFile.FileName);
            string path = Path.Combine(wwwRootPath, "images", "products",
fileName);

            using (var stream = new FileStream(path, FileMode.Create))
            {
                await imageFile.CopyToAsync(stream);
            }

            product.ImageUrl = "/images/products/" + fileName; // Зберігаємо
відносний шлях у базі
        }

        _context.Add(product);
        await _context.SaveChangesAsync();
        return RedirectToAction(nameof(AdminIndex));
    }
}

```

```

    }
    ViewBag.Categories = _context.Categories.ToList();
    ViewBag.Manufacturers = _context.Manufacturers.ToList();
    return View(product);
}

// GET: /Products/Edit/5
[Authorize(Roles = "Admin")]
public async Task<IActionResult> Edit(int? id)
{
    if (id == null)
    {
        return NotFound();
    }

    var product = await _context.Products.FindAsync(id);
    if (product == null)
    {
        return NotFound();
    }
    ViewBag.Categories = _context.Categories.ToList();
    ViewBag.Manufacturers = _context.Manufacturers.ToList();
    return View(product);
}

// POST: /Products/Edit/5
[HttpPost]
[ValidateAntiForgeryToken]
[Authorize(Roles = "Admin")]
public async Task<IActionResult> Edit(int id,
[Bind("Id,Name,Description,Price,StockQuantity,CategoryId,ManufacturerId,ImageUrl,IsActive")] Product product, IFormFile imageFile)
{
    if (id != product.Id)
    {
        return NotFound();
    }
}

```

```

if (ModelState.IsValid)
{
    try
    {
        if (imageFile != null && imageFile.Length > 0)
        {
            // Видаляємо старе зображення, якщо воно існує
            if (!string.IsNullOrEmpty(product.ImageUrl))
            {
                string oldImagePath =
Path.Combine(_webHostEnvironment.WebRootPath, product.ImageUrl.TrimStart('/'));
                if (System.IO.File.Exists(oldImagePath))
                {
                    System.IO.File.Delete(oldImagePath);
                }
            }

            // Зберігаємо нове зображення
            string wwwRootPath = _webHostEnvironment.WebRootPath;
            string fileName = Guid.NewGuid().ToString() +
Path.GetExtension(imageFile.FileName);
            string path = Path.Combine(wwwRootPath, "images", "products",
fileName);

            using (var stream = new FileStream(path, FileMode.Create))
            {
                await imageFile.CopyToAsync(stream);
            }
            product.ImageUrl = "/images/products/" + fileName;
        }
        // Якщо imageFile == null, то ImageUrl залишається тим самим

        _context.Update(product);
        await _context.SaveChangesAsync();
    }
    catch (DbUpdateConcurrencyException)

```

```

        {
            if (!ProductExists(product.Id))
            {
                return NotFound();
            }
            else
            {
                throw;
            }
        }

        return RedirectToAction(nameof(AdminIndex));
    }

    ViewBag.Categories = _context.Categories.ToList();
    ViewBag.Manufacturers = _context.Manufacturers.ToList();
    return View(product);
}

// GET: /Products/Delete/5
[Authorize(Roles = "Admin")]
public async Task<IActionResult> Delete(int? id)
{
    if (id == null)
    {
        return NotFound();
    }

    var product = await _context.Products
        .Include(p => p.Category)
        .Include(p => p.Manufacturer)
        .FirstOrDefaultAsync(m => m.Id == id);

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

    return View(product);
}

```

```

}

// POST: /Products/Delete/5
[HttpPost, ActionName("Delete")]
[ValidateAntiForgeryToken]
[Authorize(Roles = "Admin")]
public async Task<IActionResult> DeleteConfirmed(int id)
{
    var product = await _context.Products.FindAsync(id);
    if (product != null)
    {
        // Видаляємо зображення з файлової системи
        if (!string.IsNullOrEmpty(product.ImageUrl))
        {
            string imagePath = Path.Combine(_webHostEnvironment.WebRootPath,
product.ImageUrl.TrimStart('/'));
            if (System.IO.File.Exists(imagePath))
            {
                System.IO.File.Delete(imagePath);
            }
        }
        _context.Products.Remove(product);
    }

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

private bool ProductExists(int id)
{
    return _context.Products.Any(e => e.Id == id);
}
}

Представлення (View)
@model IEnumerable<YourProjectName.Models.Product>

```

```

@{
    ViewData["Title"] = "Каталог товарів";
}

<h1>Каталог товарів</h1>

<div class="row">
    <div class="col-md-3">
        <h3>Категорії</h3>
        <ul class="list-group">
            @foreach (var category in ViewBag.Categories)
            {
                <li class="list-group-item @(ViewBag.CurrentCategory == category.Id ?
"active" : "")">
                    <a href="@Url.Action("Index", "Products", new { categoryId =
category.Id, searchTerm = ViewBag.SearchTerm })">
                        @category.Name
                    </a>
                </li>
            }
            <li class="list-group-item @(ViewBag.CurrentCategory == null ? "active"
: "")">
                <a href="@Url.Action("Index", "Products", new { searchTerm =
ViewBag.SearchTerm })">
                    Усі товари
                </a>
            </li>
        </ul>
    </div>
    <div class="col-md-9">
        <form asp-controller="Products" asp-action="Index" method="get" class="mb-3">
            <div class="input-group">
                <input type="text" name="searchTerm" class="form-control"
placeholder="Пошук товарів..." value="@ViewBag.SearchTerm" />
                <button type="submit" class="btn btn-primary">Пошук</button>
            </div>
        </form>
    </div>

```

```

<div class="row">
    @foreach (var item in Model)
    {
        <div class="col-md-4 mb-4">
            <div class="card h-100">
                @if (!string.IsNullOrEmpty(item.ImageUrl))
                {
                    
                }
                else
                {
                    
                }
                <div class="card-body">
                    <h5 class="card-title">@item.Name</h5>
                    <p class="card-text text-muted">@item.Category.Name</p>
                    <p class="card-
text"><strong>@item.Price.ToString("C")</strong></p>
                    @if (item.StockQuantity > 0)
                    {
                        <p class="card-text text-success">В наявності:
@item.StockQuantity шт.</p>
                    }
                    else
                    {
                        <p class="card-text text-danger">Немає в
наявності</p>
                    }
                    <a asp-action="Details" asp-route-id="@item.Id"
class="btn btn-info">Детальніше</a>
                    <button class="btn btn-primary add-to-cart-btn" data-
product-id="@item.Id" data-product-name="@item.Name" data-product-
price="@item.Price.ToString("F2")">Додати в кошик</button>
                </div>
            </div>
        </div>
    </div>
</div>

```

```

    }
</div>

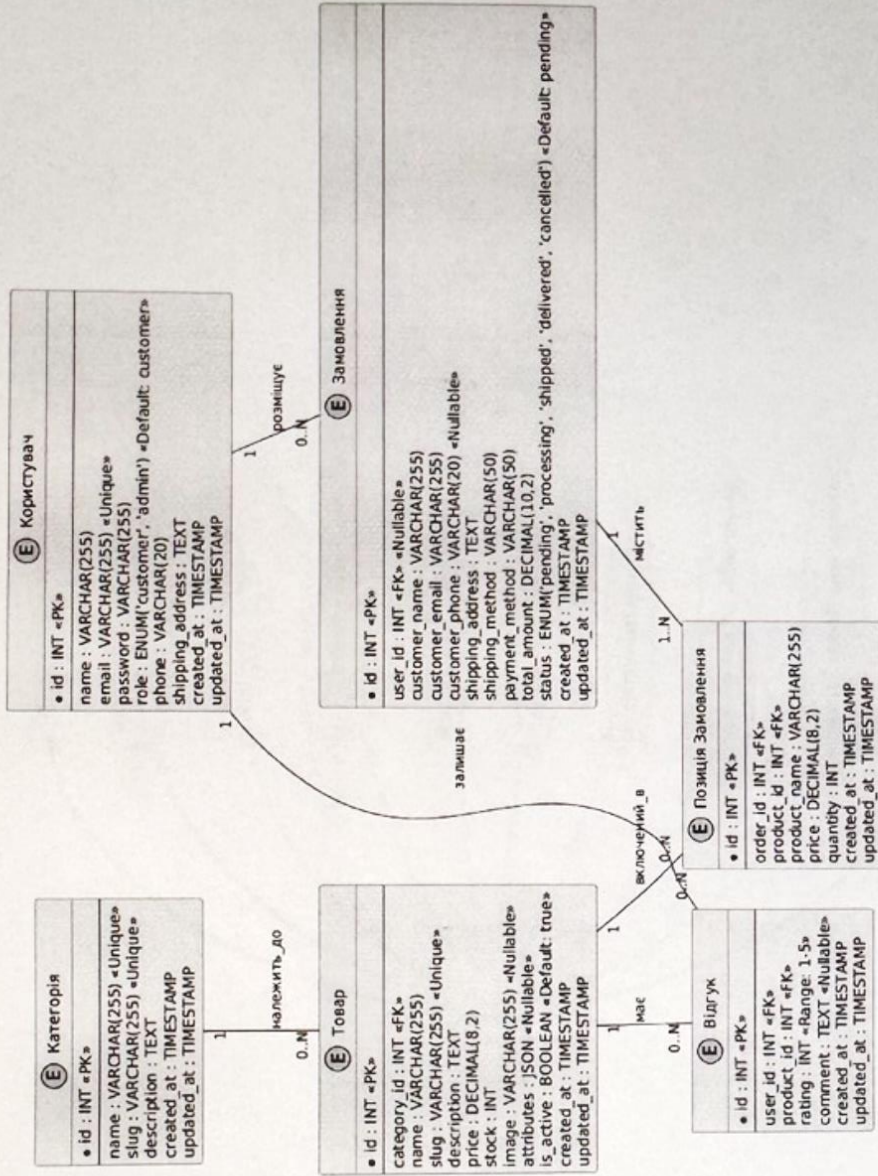
@* Пагінація *@
<nav aria-label="Product Page Navigation">
    <ul class="pagination justify-content-center">
        @if (ViewBag.PageNumber > 1)
        {
            <li class="page-item">
                <a class="page-link" href="@Url.Action("Index", "Products",
new { categoryId = ViewBag.CurrentCategory, searchTerm = ViewBag.SearchTerm,
pageNumber = ViewBag.PageNumber - 1 })">Попередня</a>
            </li>
        }
        @for (int i = 1; i <= ViewBag.TotalPages; i++)
        {
            <li class="page-item @(ViewBag.PageNumber == i ? "active" : "")">
                <a class="page-link" href="@Url.Action("Index", "Products",
new { categoryId = ViewBag.CurrentCategory, searchTerm = ViewBag.SearchTerm,
pageNumber = i })">@i</a>
            </li>
        }
        @if (ViewBag.PageNumber < ViewBag.TotalPages)
        {
            <li class="page-item">
                <a class="page-link" href="@Url.Action("Index", "Products",
new { categoryId = ViewBag.CurrentCategory, searchTerm = ViewBag.SearchTerm,
pageNumber = ViewBag.PageNumber + 1 })">Наступна</a>
            </li>
        }
    </ul>
</nav>
</div>
</div>

@section Scripts {
    <script>
        // Приклад обробки кнопки "Додати в кошик" за допомогою JavaScript (для AJAX
або перенаправлення)

```

```
$(document).ready(function () {  
    $('.add-to-cart-btn').on('click', function () {  
        var productId = $(this).data('product-id');  
        var productName = $(this).data('product-name');  
        // Тут можна зробити AJAX запит до контролера кошика  
        alert('Товар "' + productName + '" (ID: ' + productId + ') додано до  
кошика (логіка додавання AJAX не реалізована в цьому прикладі).');  
        // Наприклад:  
        // $.post('/Cart/AddToCart', { productId: productId, quantity: 1 },  
function(response) {  
            //     if (response.success) {  
            //         alert('Товар додано!');  
            //     } else {  
            //         alert('Помилка: ' + response.message);  
            //     }  
            // });  
        });  
    });  
</script>  
}
```

ГРАФІЧНА ЧАСТИНА



КвРІПЗ.2101085.01.14.E8			
Вебзастосунок для продажу туристичного спорядження		Літера	Маса
Схема бази даних		Аркуш 2	Аркуш 3
ХНУ, ІПЗ-21-1			

Зм. Арх.	№. Додум.	Підпис	Дата
Розробник	Продумана С.В.	<i>[Signature]</i>	1.02
Керівник	Рашна О. М.	<i>[Signature]</i>	1.02
Консульта			
Н. Кошмар	Праворосна Н.У.	<i>[Signature]</i>	1.02
Зав. каф.	Бадрачек Л. П.	<i>[Signature]</i>	1.02

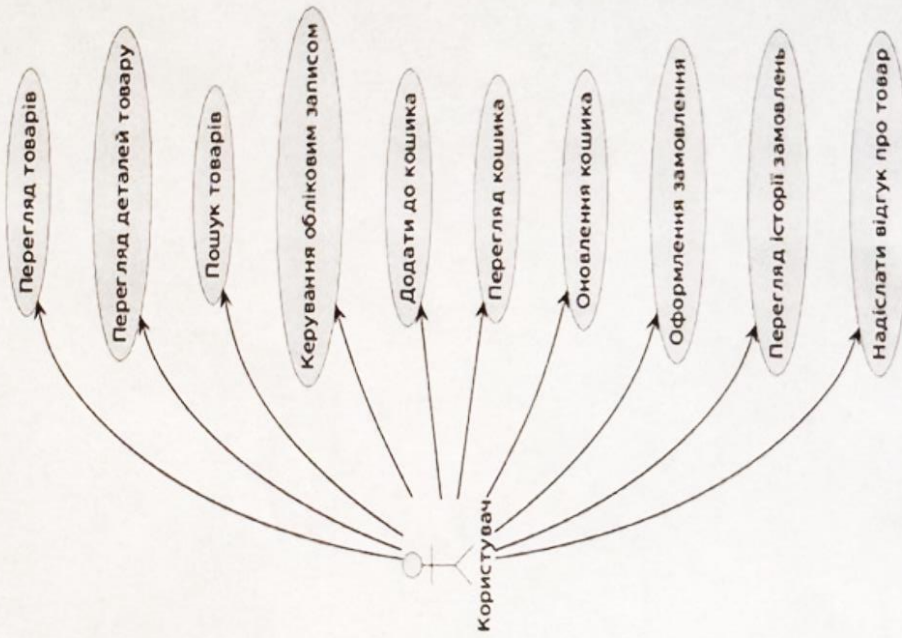
Clients



Server



КвРІПЗ.2101085.14.Е8		Листопад	Місяць	Масштаб
Вебзастосунок для продажу туристичного спорядження				
Архитектура				
Зм. Арк.	№ док-м.	Підпис	Дата	
Розробник	Покшолова С.В.	<i>[Signature]</i>	27.11.2023	
Керівник	Ришак О.М.	<i>[Signature]</i>		
Консультант				
Н. Контр.	Правороса Н.І.	<i>[Signature]</i>		
Зав. каф.	Боднарчук Л.П.	<i>[Signature]</i>		
		ХНУ, ІПЗ-21-1		



КвР ІПЗ.2101085.01.14.Е8				
Зміст	Арх.	№ докум.	Підпис	Дата
Везаастосунок для продажу туристичного спорядження				
Діаграма варіантів використання				
Літера	Маса	Масштаб		
Аркуш 1	Аркуше 3			
ХНУ, ІПЗ-21-1				

СУПРОВІДНІ ДОКУМЕНТИ

Завідувачу кафедри інженерії програмного
забезпечення проф. Леоніду БЕДРАТЮКУ
здобувача вищої освіти
Придолоби Сергія Віталійовича
факультет ІТ, ІV курс, група ІПЗ-21-1

ЗАЯВА

З правилами чинного Положення про систему забезпечення академічної доброчесності в Хмельницькому національному університеті, згідно з яким виявлення академічного плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту і застосування заходів дисциплінарної та академічної відповідальності, ознайомлений. Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на наявність академічного плагіату оповіщений та надаю свою згоду на обробку й збереження університетом моєї роботи в інституційному репозитарії Хмельницького національного університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-обчислювального комплексу StrikePlagiarism та/або програмно-технічного засобу AntiPlagiarism і використання роботи для виявлення академічного плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення текстових збігів у роботах.

Робота надається для перевірки в електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

02.01.2025
дата


підпис

Протокол аналізу звіту подібності експертом

Заявляю, що я ознайомився (-лась) з Повним звітом подібності, який був згенерований Системою виявлення і запобігання плагіату щодо роботи:

Автор: Придолоба Сергій

Співавтор:

Назва: БКР_Вебзастосунок для продажу туристичного спорядження

Експерт:

Підрозділ: Кафедра інженерії програмного забезпечення

Коефіцієнт подібності 1:6.5%

Коефіцієнт подібності 2:0.3%

Мікропробіли: 0

Заміна букв: 1

Інтервали: 0

Білі знаки: 0

Дата створення звіту: 2025-06-07 13:40:18.0

Після аналізу Звіту подібності констатую наступне:

Запозичення, виявлені в роботі є законними і не є плагіатом. Рівень подібності не перевищує допустимої межі. Таким чином робота незалежна і приймається.

Запозичення не є плагіатом, але перевищено граничне значення рівня подібностей. Таким чином робота повертається на доопрацювання.

Виявлено запозичення і плагіат або навмисні текстові спотворення (маніпуляції), як передбачувані спроби укриття плагіату, які роблять роботу невідповідною вимогам законодавства (Ст. 32. ЗУ Про вищу освіту, пункт 3.1, Ст. 42. ЗУ Про освіту) та вимог НАЗЯВО (Критерій 5), а також кодексу етики і процедур. Таким чином робота не приймається.

Обґрунтування:

Дата 8, 06, 25 р.

експерт

Anti-Plagiarism (UA) v-15.281 Educational

The maximum coincidence with one document 4.0%

Dictionaries check: en_US, ru_RU, ua_UA. Errors in the documents: 13%

ID: 244355 Title: БКР - Вебастосунок для продажу туристичного спорядження Added in a DB: 2025-06-09 Authors: Придолоба Сергій Heads: ЯШПІНА Оксана канд. техн. наук, доцент Consultants: Opponents:	Document		Sum coincidence on the DB	
	Symbols	Lexemes	Symbols	Lexemes
	53459	832	4924 (9%)	65 (8%)

Plagiarism sources

ID	Description	Plagiarism presence in the document	
		Symbols	Lexemes

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ
освітнього ступеня «Бакалавр»

Дипломник Придолоба Сергій Віталійович

Тема Вебзастосунок для продажу туристичного спорядження

Спеціальність 121 – Інженерія програмного забезпечення

Обсяг кваліфікаційної роботи:

Кількість листів креслень 3; кількість сторінок записки 107

1. Короткий зміст пояснювальної записки та прийнятих рішень У кваліфікаційній роботі здійснено аналіз предметної області та визначено вимоги до розроблюваного програмного продукту, а також проаналізовано існуючі програмні рішення, що функціонують на ринку. Проведено аналіз та вибір технологій для вирішення поставлених задач, а саме проектування та розробку бази даних, інтерфейсу та всіх програмних модулів. Важливим моментом є тестування розроблено програмного застосунку.

2. Висновок про відповідність роботи поставленому завданню Кваліфікаційна робота виконана відповідно до поставленого завдання та з дотриманням всіх вимог.

3. Характеристика виконання кожного розділу роботи, ступінь використання останніх досягнень науки і техніки та передових методів роботи У вступі показано актуальність теми, визначено мету та завдання кваліфікаційної роботи. У першому розділі проаналізовано предметну область, існуючі рішення, вимоги до розроблюваного програмного забезпечення. Другий розділ містить аналітичний огляд сучасних архітектур, із виокремленням їх переваг та недоліків розглянуто їх переваги і недоліки та визначено, що система буде відповідати архітектурі клієнт-сервер. У третьому розділі підготовлено всі залежності для написання коду та виконано практичну розробку програмних модулів і описано їх особливості, в результаті чого створено програмний продукт. Також у цьому розділі виконано функціональне та модульне тестування та виявлено правильність роботи програмного вебзастосунку.

4. Позитивні сторони роботи Тема кваліфікаційної роботи є актуальною, оскільки на сьогодні в Україні активно розвиваються продажі в мережі Інтернет на різноманітних платформах, онлайн-ресурсах тощо. Також було застосовано новітні технології для побудови програмного продукту та актуальні архітектурні рішення.

5. Негативні сторони роботи Недостатньо висвітлена реалізація інтерфейсу вебзастосунку, відсутні прототипи чи мокапи під час проектування.

6. Оцінка графічного оформлення та пояснювальної записки Графічне оформлення виконано відповідно до теми кваліфікаційної роботи та подано у вигляді діаграм і рисунків. Пояснювальна записка оформлена згідно вимог чинних стандартів.

7. Відгук про кваліфікаційну роботу в цілому Кваліфікаційна робота заслуговує позитивної оцінки. Матеріал пояснювальної записки структурований, послідовний, чіткий та простий, що дозволяє чітко зрозуміти викладений матеріал у рамках тематики проектування. Графічний матеріал дає можливість наочно побачити деталі проектування системи.

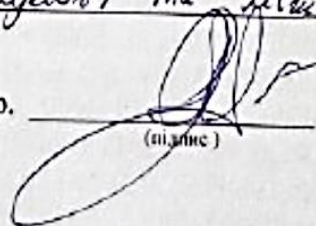
8. Інші зауваження _____

9. Оцінка кваліфікаційної роботи Кваліфікаційна робота виконана у повному обсязі, відповідає поставленій задачі та заслуговує на оцінку «добре».

РЕЦЕНЗЕНТ Лисенко С. М. д. т. н., професор, заступник декана факультету інформаційних технологій з інженерії та міжнародної роботи

"09" серпня

2025 р.



(підпис)

РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ КАФЕДРИ Інженерії програмного забезпечення

ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Назва кваліфікаційної роботи Вебзастосунок для продажу туристичного спорядження

Автор Придолоба Сергій Віталійович

Освітня програма 121 – Інженерія програмного забезпечення

Рівень вищої освіти Перший (бакалаврський)

Спеціальність Інженерія програмного забезпечення

Науковий керівник: Яшина Оксана Миколаївна, канд. техн. наук, доцент

На основі аналізу кваліфікаційної роботи на дотримання вимог академічної доброчесності (у т.ч. відсутності ознак академічного плагіату) з урахуванням результатів перевірки роботи спеціалізованим програмним засобом(ами) комісія зробила такий висновок:

№	Висновок	Позначка про відповідність
1	Ознаки академічного плагіату	
1.1	Запозичення, виявлені в роботі, є законними і не є академічним плагіатом (далі – зазначаються підстави віднесення запозичень до правомірних, якщо потрібно). Робота приймається до захисту.	відповідає
1.2	Виявлені запозичення не є академічним плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи (далі – зазначаються детальні та аргументовані підстави віднесення запозичень до правомірних). Робота приймається до захисту, але має бути відкоригована.	
1.3	Виявлені запозичення не є академічним плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота може бути допущена до захисту після того як буде відкоригована та доопрацьована і успішно пройде повторну перевірку на академічний плагіат.	
1.4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття текстових запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	
2	Інші види порушень академічної доброчесності	

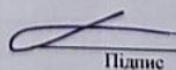
Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

- у тексті кваліфікаційної роботи системами перевірки на плагіат виявлено схожість з деякими документами в частині загальноживаних обов'язкових словосполучень у стандартних бланках (титулка, завдання, анотація, відомість документів), у структурі змісту, назвах розділів/підрозділів тощо, у назвах публікацій та у переліку джерел посилання;
- в якості запозичень системою було зафіксовано деякі послідовності вихідного коду і посилання на бібліотеки, які є стандартними мовними конструкціями програмування та не можуть розглядатися як об'єкт авторських прав і, відповідно, їх порушення;
- усі запозичення є фрагментарними або мають належним чином оформленні посилання;
- виявлені модифікації тексту не впливають на відсоток схожості. Сумарний обсяг всіх запозичень, визначений системою виявлення збігів ідентичності/ схожості, складає 6,5%, що, з урахуванням наведених обґрунтувань, відповідає характеру теми і свідчить на користь кваліфікаційної роботи

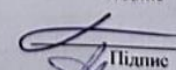
Дата 10.01.25

Завідувач кафедри


Підпис

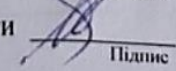
Леонід Федоратюк
Ім'я, ПРІЗВИЩЕ

Гарант освітньої програми


Підпис

Леонід Федоратюк
Ім'я, ПРІЗВИЩЕ

Керівник кваліфікаційної роботи


Підпис

Оксана Яшина
Ім'я, ПРІЗВИЩЕ