

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

**Відокремлений структурний підрозділ
«Тернопільський фаховий коледж
Тернопільського національного технічного університету імені Івана Пулюя»
Відділення телекомунікацій та електронних систем
Циклова комісія комп'ютерних наук**

ПОЯСНЮВАЛЬНА ЗАПИСКА

до кваліфікаційної роботи

фахового молодшого бакалавра

на тему: Розробка інтернет-магазину велосипедів «Biker»

Виконав: студент IV курсу, групи КН-421
спеціальності: 122 «Комп'ютерні науки»

Владислав ХРОМОВ

Керівник **Галина МАРЦЯШ**

Рецензент **Юрій ЛЕЩИШИН**

(ім'я та прізвище)

ВІДОКРЕМЛЕНИЙ СТРУКТУРНИЙ ПІДРОЗДІЛ
«ТЕРНОПІЛЬСЬКИЙ ФАХОВИЙ КОЛЕДЖ
ТЕРНОПІЛЬСЬКОГО НАЦІОНАЛЬНОГО ТЕХНІЧНОГО УНІВЕРСИТЕТУ ІМЕНІ ІВАНА
ПУЛЮЯ»

Відділення телекомунікацій та електронних систем
Циклова комісія комп'ютерних наук
Освітньо-професійний ступінь «фаховий молодший бакалавр»
Спеціальність 122 «Комп'ютерні науки»
Галузь знань 12 «Інформаційні технології»

ЗАТВЕРДЖУЮ

Голова циклової комісії
комп'ютерних наук
_____ Галина МАРЦІЯШ
« 02 » березня 2026 року

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Хромову Владиславу Юрійовичу

(прізвище, ім'я, по батькові)

1. Тема роботи: Розробка інтернет-магазину велосипедів «Viker»

керівник роботи Марціяш Галина Ярославівна,

(прізвище, ім'я, по батькові)

затверджені наказом вищого навчального закладу № 4/9-132 від 27.02.2026 р.

2. Строк подання студентом роботи: 19.06.2026 р.

3. Вихідні дані до роботи: технічне завдання на розробку програмного забезпечення, мови програмування: TypeScript, PostgreSQL, React, HTML, стандарти IEEE 830-1998, IEEE 29148-2018, IEEE 29119, ГОСТ 34.602-89.

4. Зміст розрахунково-пояснювальної записки

1 Загальний розділ

1.1 Аналітичний огляд існуючих рішень

1.2 Технічне завдання

1.2.1 Найменування та область застосування

1.2.2 Призначення розробки

1.2.3 Вимоги до функціоналу вебсайту

1.2.4 Вимоги до програмної документації

1.2.5 Техніко-економічні показники

1.2.6 Стадії та етапи розробки

1.2.7 Порядок тестування та прийому

- 2 Розробка технічного та робочого проєкту
 - 2.1 Розробка структури сайту і вебсторінок
 - 2.2 Створення та верстка сторінок сайту
 - 2.3 Розробка структури бази даних сайту
 - 2.4 Програмування сайту
 - 2.4.1 Написання клієнтської частини
 - 2.4.2 Написання адміністративної частини
 - 2.5 Тестування вебсайту
- 3 Спеціальний розділ
 - 3.1 Інструкція з розміщення сайту в Інтернеті
 - 3.2 Інструкція з обслуговування та наповнення сайту
 - 3.3 Інструкція з популяризації та підтримки сайту
- 4 Економічний розділ
 - 4.1 Визначення стадій технологічного процесу та загальної тривалості проведення НДР
 - 4.2 Визначення витрат на оплату праці та відрахувань на соціальні заходи
 - 4.3 Розрахунок витрат на електроенергію
 - 4.4 Розрахунок суми амортизаційних відрахувань інтернет-магазину велосипедів «Viker»
 - 4.5 Обчислення накладних витрат
 - 4.6 Складання кошторису витрат та визначення собівартості інтернет-магазину велосипедів «Viker»
 - 4.7 Розрахунок ціни інтернет-магазину велосипедів «Viker»
 - 4.8 Визначення економічної ефективності і терміну окупності капітальних вкладень
- 5 Охорона праці, техніка безпеки та екологічні вимоги
 - 5.1 Система управління охороною праці
 - 5.2 Мікроклімат виробничих приміщень та його параметри
- 6 Висновки

Додаткові вказівки:

Виконання роботи із розробкою інтернет-магазину велосипедів «Viker».

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, наприклад):

- 1. Структурна схема
- 2. Структура шаблону вебсторінок
- 3. ER діаграма баз даних
- 4. Таблиця техніко-економічних показників.
- 5. UML-діаграми діяльності оформлення замовлення

6. Консультанти розділів роботи

Розділ	Ім'я, прізвище та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Економічний розділ	Любов КАЛУШКА		
Охорона праці, техніка безпеки та екологічні вимоги	Генадій ГОРЯЧЕК		

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Отримання і аналіз технічного завдання	20.03.2026	
2	Збір і узагальнення інформації	01.05.2026	
3	Написання першого розділу	15.05.2026	
4	Розробка технічного та робочого проєкту	29.05.2026	
5	Написання спеціального розділу	05.06.2026	
6	Розрахунок економічної частини	08.06.2026	
7	Написання розділу охорони праці	09.06.2026	
8	Виконання графічної частини	10.06.2026	
9	Оформлення пояснювальної записки	11.06.2026	
10	Погодження нормоконтролю	12.06.2026	
11	Попередній захист кваліфікаційної роботи	10.06.2026	
12	Захист кваліфікаційної роботи	22.06.2026	

7. Дата видачі завдання: 05 березня 2026 р.

Студент

(підпис)

Владислав ХРОМОВ

Керівник роботи

(підпис)

Галина МАРЦІЯШ

ЗМІСТ

АНОТАЦІЯ	7
ANOTATION	8
ВСТУП	9
1 ЗАГАЛЬНИЙ РОЗДІЛ.....	11
1.1 Аналітичний огляд існуючих рішень	11
1.2 Технічне завдання.....	15
1.2.1 Найменування та область застосування	15
1.2.2 Призначення розробки.....	15
1.2.3 Вимоги до функціоналу вебсайту	16
1.2.4 Вимоги до програмної документації.....	23
1.2.5 Техніко-економічні показники.....	24
1.2.6 Стадії та етапи розробки.....	25
1.2.7 Порядок тестування та прийому	26
2 РОЗРОБКА ТЕХНІЧНОГО ТА РОБОЧОГО ПРОЄКТУ.....	29
2.1 Розробка структури сайту і вебсторінок.....	29
2.2 Створення та верстка сторінок сайту	34
2.3 Розробка структури бази даних сайту	38
2.4 Програмування сайту	42
2.4.1 Написання клієнтської частини	46
2.4.2 Написання адміністративної частини.....	47
2.5 Тестування вебсайту.....	49
3 СПЕЦІАЛЬНИЙ РОЗДІЛ	57
3.1 Інструкція з розміщення сайту в Інтернеті.....	57
3.2 Інструкція з обслуговування та наповнення сайту	60
3.3 Інструкція з популяризації та підтримки сайту	64

					2026.KBP.122.421.16.00.00 ПЗ			
Зм.	Арк.	№ докум.	Підпис	Дата	Розробка інтернет-магазину велосипедів «Viker» Пояснювальна записка	Літ.	Арк.	Аркуші
Розроб.		Хромов В. Ю.						
Перевір.		Марціян Г.Я.						
Реценз.								
Н. Контр.		Приймак В.А.						
Затверд.					ВСП ТФК ТНТУ КН-421 м. Тернопіль			

4. ЕКОНОМІЧНИЙ РОЗДІЛ.....	67
4.1. Визначення стадій технологічного процесу та загальної тривалості проведення НДР.....	67
4.2. Визначення витрат на оплату праці та відрахувань на соціальні заходи .	69
4.3. Розрахунок витрат на електроенергію.....	71
4.4. Розрахунок суми амортизаційних відрахувань інтернет-магазину велосипедів «Viker».....	71
4.5. Обчислення накладних витрат.....	72
4.6. Складання кошторису витрат та визначення собівартості інтернет-магазину велосипедів «Viker».....	72
4.7. Розрахунок ціни інтернет-магазину велосипедів «Viker».....	73
4.8. Визначення економічної ефективності і терміну окупності капітальних вкладень.....	73
5 ОХОРОНА ПРАЦІ, ТЕХНІКА БЕЗПЕКИ ТА ЕКОЛОГІЧНІ ВИМОГИ	75
5.1 Система управління охороною праці.....	75
5.2 Мікроклімат виробничих приміщень та його параметри.....	76
ВИСНОВКИ.....	77
ПЕРЕЛІК ПОСИЛАНЬ.....	79
ДОДАТКИ.....	81
Додаток А. Лістинг програмного коду оголошення моделей Prisma	81
Додаток Б. Лістинг програмного коду серверних дій для створення товарів .	86
Додаток В. Лістинг програмного коду реалізації клієнтського стану кошика	91
Додаток Г. Лістинг функції створення замовлення.....	93
Додаток Д. Лістинг програмного коду авторизації	98

АНОТАЦІЯ

Тема кваліфікаційної роботи: Розробка інтернет-магазину велосипедів «Viker».

Метою кваліфікаційної роботи є розробка вебзастосунку електронної комерції для представлення асортименту велосипедів, перегляду детальної інформації про товари, додавання товарів до кошика, оформлення замовлень, оплати покупки та адміністрування товарів і замовлень.

Пояснювальна записка складається з п'яти основних розділів, висновків, переліку посилань та додатків.

У загальній частині описано аналітичний огляд існуючих рішень у сфері інтернет-магазинів, визначено призначення програмного продукту та сформульовано технічне завдання на розробку вебзастосунку.

У другому розділі представлено процес створення вебзастосунку: розробку структури сайту і вебсторінок, створення та верстку сторінок, проектування структури бази даних, програмування клієнтської та адміністративної частин, а також тестування вебзастосунку.

У спеціальній частині описано інструкцію з розміщення сайту в Інтернеті, порядок обслуговування та наповнення сайту, а також заходи з популяризації, резервного копіювання, моніторингу й підтримки актуальності вебзастосунку.

Розрахунок вартості розробки та економічної ефективності наведено в економічній частині.

Основні питання охорони праці, техніки безпеки та екологічних вимог розглянуто в п'ятому розділі.

Обсяг пояснювальної записки — 100.

До складу кваліфікаційної роботи входить графічна частина, яка складається з п'яти аркушів формату А1: структурної схеми вебзастосунку, ER-діаграми бази даних, карти сторінок сайту, UML-діаграми діяльності оформлення замовлення та таблиці техніко-економічних показників.

					2026.КВР.122.421.16.00.00 ПЗ	Арк.
						7
Зм.	Арк.	№ докум.	Підпис	Дата		

ANOTATION

Thesis Topic: Development of the “Biker” Online Bicycle Store.

The objective of this thesis is to develop an e-commerce web application for displaying a range of bicycles, viewing detailed product information, adding items to the shopping cart, placing orders, processing payments, and managing products and orders.

The explanatory note consists of five main sections, conclusions, a list of references, and appendices.

The general section provides an analytical overview of existing solutions in the field of online stores, defines the purpose of the software product, and formulates the technical specifications for developing the web application.

The second section presents the process of creating the web application: developing the site structure and web pages, creating and laying out pages, designing the database structure, programming the client and administrative components, as well as testing the web application.

The technical section describes instructions for hosting the site on the Internet, the procedure for maintaining and updating the site, as well as measures for promotion, backup, monitoring, and ensuring the web application remains up-to-date.

The calculation of development costs and economic efficiency is presented in the economic section.

Key issues related to occupational health and safety and environmental requirements are addressed in the fifth chapter.

The explanatory note is 100 pages long.

The thesis includes a visual component consisting of five A1-size sheets: a web application architecture diagram, a database ER diagram, a site map, a UML diagram of the order placement process, and a table of technical and economic indicators.

					2026.KBP.122.421.16.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		8

ВСТУП

Електронна комерція стала одним із ключових напрямів розвитку сучасних інформаційних систем, оскільки дає змогу підприємствам продавати товари без обмеження фізичним місцем розташування, а покупцям — швидко отримувати інформацію про асортимент, характеристики продукції, ціни, відгуки та умови купівлі. Особливо актуальними такі рішення є для магазинів велосипедів, де користувачеві важливо порівняти різні моделі, переглянути зображення, оцінити характеристики, вибрати колір, додати товар до кошика та оформити замовлення без зайвих дій.

Актуальність теми полягає в необхідності створення гнучкого програмного продукту, який поєднує функції каталогу товарів, системи оформлення замовлень, авторизації користувачів, онлайн-оплати та адміністративного керування. Готові платформи електронної комерції часто обмежують можливості індивідуального налаштування логіки та інтерфейсу, тому розробка власного вебзастосунку дає змогу повністю контролювати структуру даних, сценарії користувача, дизайн і подальший розвиток системи.

Метою кваліфікаційної роботи є розробка інтернет-магазину велосипедів «Viker», який забезпечує перегляд товарів, роботу з каталогом, оформлення замовлень, оплати та адміністрування основних об'єктів системи.

Для досягнення мети необхідно виконати такі завдання:

- проаналізувати предметну область інтернет-магазинів велосипедів;
- визначити функціональні та нефункціональні вимоги до програмного продукту;
- сформулювати технічне завдання на розробку;
- обґрунтувати вибір технологій для клієнтської частини, серверної логіки, бази даних, авторизації та оплати;
- спроектувати структуру вебзастосунку та бази даних;
- реалізувати головну сторінку, каталог, сторінку товару, кошик, оформлення замовлення, профіль користувача та адміністративну панель;

					<i>2026.KBP.122.421.16.00.00 ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		9

- реалізувати авторизацію користувачів і розмежування доступу;
- забезпечити інтеграцію з платіжною системою Stripe;
- описати порядок інсталяції, експлуатації та тестування програмного продукту;
- підготувати пояснювальну записку та графічні матеріали.

Об'єктом роботи є процес онлайн-продажу велосипедів через вебзастосунок. Предметом роботи є методи та програмні засоби реалізації інтернет-магазину з каталогом товарів, кошиком, замовленнями, оплатою, авторизацією та адміністративним керуванням.

У процесі виконання кваліфікаційної роботи будуть використані методи аналізу предметної області, порівняння існуючих рішень електронної комерції, проектування структури вебзастосунку, моделювання бази даних, програмної реалізації клієнтської та серверної частин, а також тестування основних функціональних можливостей системи. Такий підхід дає змогу послідовно перейти від визначення вимог до створення готового програмного продукту.

Користувачами розробленого інтернет-магазину будуть відвідувачі сайту, зареєстровані покупці та адміністратори. Відвідувачі будуть мати можливість переглядати каталог велосипедів, знайомитися з характеристиками товарів і додавати їх до кошика. Адміністратор забезпечуватиме наповнення каталогу, редагування товарів, перегляд замовлень і зміну їх статусів.

Практичне значення роботи полягає у створенні програмного продукту, який може використовуватися як основа для реального інтернет-магазину велосипедів або як навчальний приклад розробки повноцінного вебзастосунку електронної комерції. Розробка демонструє застосування сучасного стеку технологій: Next.js, React, TypeScript, Prisma ORM, PostgreSQL, NextAuth, Stripe, Tailwind CSS і Amazon S3. Оформлення пояснювальної записки й переліку посилань узгоджується з вимогами ДСТУ 3008:2015 [17] та ДСТУ 8302:2015 [16].

Структуру пояснювальної записки сформовано відповідно до вимог методичних вказівок до виконання кваліфікаційної роботи [20].

					<i>2026.KBP.122.421.16.00.00 ПЗ</i>	Арк.
						10
Зм.	Арк.	№ докум.	Підпис	Дата		

1 ЗАГАЛЬНИЙ РОЗДІЛ

1.1 Аналітичний огляд існуючих рішень

Електронна комерція є одним із найпоширеніших напрямів використання веб-технологій, оскільки дає змогу автоматизувати процеси ознайомлення покупця з товаром, вибору потрібної конфігурації, оформлення замовлення, оплати та подальшого супроводу покупки. Для магазинів спортивних товарів і велосипедів це особливо актуально: покупець має швидко отримати характеристики товару, фотографії, ціну, відгуки інших користувачів, варіанти доставки та зручний спосіб оплати.

Інтернет-магазин велосипедів повинен поєднувати функції електронного каталогу, системи приймання замовлень, засобу взаємодії з клієнтом і адміністративної панелі для керування товарами та замовленнями. На відміну від звичайного інформаційного сайту, такий програмний продукт має працювати з динамічними даними, підтримувати авторизацію користувачів, зберігати інформацію про товари, користувачів, відгуки, замовлення та статуси виконання. Отже, розробка інтернет-магазину вимагає не лише створення зручного інтерфейсу, а й реалізації серверної логіки, бази даних, механізмів безпечної взаємодії з платіжною системою та засобів адміністрування.

На ринку існує декілька типів рішень, які можуть використовуватися для продажу велосипедів в Інтернеті. До першої групи належать універсальні маркетплейси, де продавці розміщують товари поруч із пропозиціями інших компаній. Перевагою такого підходу є наявність готової аудиторії, інструментів пошуку, фільтрації, оплати та доставки. Однак маркетплейси обмежують можливості індивідуального оформлення магазину, контролю над користувацьким досвідом і впровадження специфічної логіки, потрібної для окремого бізнесу. Для магазину велосипедів це може бути недоліком, оскільки важливо не лише показати товар, а й сформуванню власний бренд, виділити переваги моделей, надати покупцеві зрозумілий шлях від вибору до оформлення замовлення.

					2026.КВР.122.421.16.00.00 ПЗ	Арк.
						11
Зм.	Арк.	№ докум.	Підпис	Дата		

До другої групи належать SaaS-платформи для електронної комерції. Вони надають готову адміністративну панель, шаблони сторінок, модулі оплати та інтеграції з доставкою. Такі рішення зменшують час запуску магазину, але мають обмеження щодо гнучкості, структури бази даних, зовнішнього вигляду інтерфейсу та можливості розробляти нестандартні функції. Крім того, використання SaaS-платформ часто пов'язане з регулярними платежами, залежністю від постачальника платформи та складністю міграції на власну інфраструктуру.

До третьої групи належать індивідуально розроблені веб-застосунки. Їхньою перевагою є повний контроль над архітектурою, інтерфейсом, бізнес-логікою, структурою даних і майбутнім розширенням функціональності. Розробка такого застосунку потребує більше часу, але дає змогу створити програмний продукт під конкретні вимоги предметної області. Саме цей варіант є доцільним для кваліфікаційної роботи, оскільки дозволяє продемонструвати повний цикл створення вебзастосунку: аналіз предметної області, проєктування бази даних, розробку клієнтської та серверної частин, реалізацію авторизації, кошика, оформлення замовлення, оплати та адміністративного керування.

Аналіз функціональності сучасних інтернет-магазинів показує, що базовими можливостями для такого типу програмного забезпечення є: перегляд головної сторінки з рекламними або рекомендованими товарами; каталог продукції; сторінка окремого товару; фільтрація та сортування товарів; додавання товару до кошика; зміна кількості товарів у кошику; оформлення замовлення; введення контактної інформації та адреси доставки; вибір способу доставки; онлайн-оплата; реєстрація та авторизація користувачів; перегляд і редагування профілю; залишення відгуків; адміністративне керування товарами, замовленнями та статусами.

Для магазину велосипедів особливо важливими є функції представлення товару. Велосипед як товар має набір характеристик, які впливають на рішення покупця: назву моделі, опис, ціну, фотографії, доступні кольори, особливості конструкції, відгуки та оцінку. Тому каталог повинен бути не лише списком

					<i>2026.КВР.122.421.16.00.00 ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		12

позицій, а інструментом пошуку потрібної моделі за параметрами. У програмному продукті «Viker» передбачено подання товарів із зображеннями, описом, ціною, набором характеристик, доступними кольорами та рейтингом на основі відгуків користувачів.

Зручність користування інтернет-магазином значною мірою визначається якістю навігації та інтерфейсу. Користувач повинен швидко перейти від головної сторінки до каталогу, відкрити сторінку товару, переглянути його фотографії, вибрати потрібну кількість, додати товар до кошика й оформити покупку. У разі складної або незрозумілої навігації зростає ймовірність того, що покупець залишить сайт без оформлення замовлення. Тому під час розробки веб-застосунку необхідно забезпечити логічну структуру сторінок, зрозумілі елементи керування, адаптивний інтерфейс і швидке оновлення даних.

Окрему роль відіграє кошик покупця. Він повинен зберігати вибрані товари, відображати кількість і вартість позицій, дозволяти змінювати кількість товарів та переходити до оформлення замовлення. У веб-застосунку «Viker» для роботи з кошиком буде використовуватися клієнтський стан Zustand[11], що дає змогу швидко реагувати на дії користувача та забезпечувати зручну взаємодію без зайвого перезавантаження сторінок.

Процес оформлення замовлення має бути поділений на логічні етапи. Доцільно спочатку отримати контактні дані й адресу доставки, потім запропонувати варіант доставки, після чого перейти до оплати. Така послідовність зменшує кількість помилок і робить процес купівлі зрозумілим для користувача. У розроблюваному інтернет-магазині оформлення замовлення буде реалізовано у вигляді послідовних вкладок: введення інформації для доставки, вибір доставки та оплата.

Для сучасного інтернет-магазину важливою є підтримка онлайн-оплати. Вона дозволяє автоматизувати приймання платежів і скоротити час між вибором товару та підтвердженням замовлення. У програмному продукті «Viker» буде передбачено інтеграцію з платіжним сервісом Stripe, що використовується для

					2026.КВР.122.421.16.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		13

створення платіжного наміру та обробки платежу. Обробка платежу через зовнішній сервіс відокремлює логіку магазину від платіжних реквізитів і зменшує ризики, пов'язані зі зберіганням чутливих фінансових даних у власній системі.

Ще одним обов'язковим компонентом інтернет-магазину є авторизація користувачів. Вона потрібна для персоналізації роботи, збереження даних профілю, зв'язування замовлень із конкретним користувачем і розмежування доступу до адміністративних можливостей. У системі «Viker» буде передбачено роботу з користувачами, сесіями, обліковими записами та ролями. Для авторизації використовуватиметься NextAuth, а дані користувачів зберігатимуться у базі даних.

Адміністративна частина є необхідною для підтримки актуальності каталогу та контролю виконання замовлень. Адміністратор повинен мати змогу створювати, редагувати й видаляти товари, переглядати список замовлень і змінювати статуси їх виконання. У розроблюваному застосунку буде передбачено адміністративні сторінки для керування товарами, замовленнями та статусами, що дає змогу обслуговувати магазин без прямого втручання в базу даних.

З технічної точки зору інтернет-магазин потребує надійної бази даних. У «Viker» структура даних охоплюватиме сутності користувача, велосипеда, замовлення, позиції замовлення, відгуку та статусу. Така модель дозволить зберігати основну інформацію, необхідну для роботи магазину, і підтримувати зв'язки між покупцями, товарами, відгуками та замовленнями. Для взаємодії з базою даних використовуватиметься Prisma ORM, а як система керування базами даних — PostgreSQL.

Під час аналізу існуючих рішень також слід враховувати нефункціональні вимоги. Інтернет-магазин повинен бути продуктивним, адаптивним, зрозумілим для користувача, придатним до розширення та супроводу. Важливими є швидкість завантаження сторінок, коректна робота на різних розмірах екрана, валідація введених даних, захист маршрутів, підтримка повторного використання компонентів і чіткий поділ відповідальності між частинами системи.

Отже, аналіз предметної області показує, що для реалізації інтернет-магазину

					2026.КВР.122.421.16.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		14

велосипедів доцільно створити окремий вебзастосунок із клієнтською частиною, серверною логікою, базою даних, системою авторизації, кошиком, оформленням замовлення, онлайн-оплатою та адміністративною панеллю. Така архітектура забезпечуватиме достатню гнучкість, дозволить адаптувати функціональність до потреб магазину велосипедів і створюватиме основу для подальшого розвитку системи.

1.2 Технічне завдання

1.2.1 Найменування та область застосування

Найменування програмного продукту: інтернет-магазин велосипедів «Viker».

Програмний продукт призначений для автоматизації процесу продажу велосипедів через мережу Інтернет. Областю застосування є електронна комерція, зокрема діяльність магазинів, що реалізують велосипеди та супутні товари. Система може використовуватися як основний веб-ресурс для представлення асортименту, приймання замовлень, взаємодії з клієнтами та адміністрування товарів.

Користувачами системи будуть:

- відвідувачі сайту, які переглядають каталог велосипедів і сторінки товарів;
- зареєстровані покупці, які додають товари до кошика, оформлюють замовлення, оплачують покупку та можуть залишати відгуки;
- адміністратори, які керують товарами, замовленнями та статусами виконання.

Результатом розробки буде веб-застосунок, який працює у браузері та забезпечує взаємодію користувача з серверною частиною і базою даних.

1.2.2 Призначення розробки

Метою розробки є створення зручного та функціонального інтернет-

					2026.КВР.122.421.16.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		15

магазину велосипедів, який забезпечуватиме повний цикл онлайн-продажу: від перегляду товару до оформлення й оплати замовлення.

Основні завдання розробки:

- створити головну сторінку з презентаційним блоком і рекомендованими товарами;
- реалізувати каталог велосипедів із можливістю перегляду списку товарів;
- забезпечити фільтрацію товарів за ціною та кольорами;
- реалізувати сортування товарів за ціною;
- створити сторінку детального перегляду товару;
- реалізувати кошик покупця з можливістю додавання товарів і зміни кількості;
- розробити форму оформлення замовлення з валідацією даних;
- передбачити вибір способу доставки;
- інтегрувати онлайн-оплату через Stripe;
- реалізувати авторизацію користувачів;
- створити профіль користувача та сторінки налаштувань;
- забезпечити можливість залишення відгуків до товарів;
- реалізувати адміністративну панель для роботи з товарами, замовленнями та статусами;
- організувати зберігання даних у реляційній базі даних;
- забезпечити адаптивність і зручність інтерфейсу.

Розробка має практичне значення, оскільки може бути використана як основа для реального інтернет-магазину або як навчальний приклад створення веб-застосунку електронної комерції з використанням сучасного стеку технологій.

1.2.3 Вимоги до функціоналу вебсайту

Функціонал інтернет-магазину «Viker» повинен забезпечувати повний цикл взаємодії користувача з електронним магазином: перегляд асортименту велосипедів, вибір товару, роботу з кошиком, оформлення та оплати замовлення, а

					2026.КВР.122.421.16.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		16

також адміністрування каталогу й замовлень. Вимоги до функціоналу доцільно подати за ролями користувачів вебзастосунку.

Для відвідувача сайту необхідно реалізувати:

- перегляд головної сторінки інтернет-магазину;
- перегляд рекламного або презентаційного блоку;
- перегляд рекомендованих товарів;
- перехід до каталогу велосипедів;
- перегляд списку товарів із короткою інформацією;
- перегляд детальної сторінки товару;
- перегляд фотографій, опису, ціни, характеристик, кольорів і рейтингу товару;
- використання фільтрів каталогу;
- сортування товарів;
- додавання товару до кошика.

Для зареєстрованого користувача необхідно реалізувати:

- реєстрацію та авторизацію;
- підтримку сесії користувача;
- перегляд і редагування даних профілю;
- введення контактних даних та адреси доставки;
- вибір способу доставки;
- оформлення замовлення;
- оплату замовлення;
- створення відгуків до товарів;
- зв'язування замовлень із обліковим записом користувача.

Для адміністратора необхідно реалізувати:

- доступ до адміністративної панелі;
- перегляд списку товарів;
- створення нового товару;
- редагування інформації про товар;

					2026.КВР.122.421.16.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		17

- видалення товару;
- завантаження зображень товару;
- перегляд замовлень;
- перегляд інформації про покупця та товари в замовленні;
- зміну статусу замовлення;
- керування переліком статусів.

До вимог щодо даних належать:

- зберігання інформації про користувачів;
- зберігання даних авторизації та сесій;
- зберігання інформації про велосипеди;
- зберігання зображень товарів;
- зберігання відгуків і рейтингу;
- зберігання замовлень і позицій замовлення;
- зберігання статусів замовлень.

Програмний продукт повинен використовувати реляційну базу даних PostgreSQL. Для роботи з базою даних використовуватиметься Prisma ORM, що забезпечить опис моделей даних, виконання запитів і взаємодію серверної частини з таблицями бази даних.

Клієнтська частина повинна бути реалізована з використанням React і Next.js. Застосунок має використовувати маршрутизацію Next.js, серверні дії для роботи з даними, компонентний підхід до побудови інтерфейсу та адаптивну верстку. Мова програмування — TypeScript, що дозволяє підвищити надійність коду завдяки статичній типізації.

Інтерфейс користувача повинен бути зрозумілим, сучасним і придатним для використання на різних пристроях. Для побудови інтерфейсу використовуватиметься компонентні UI-рішення, зокрема Radix UI[10], а стилізація виконуватиметься засобами Tailwind CSS[9]. Система повинна мати єдину візуальну структуру, повторно використовувані компоненти та логічно згруповані сторінки.

					2026.KBP.122.421.16.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		18

Для автентифікації користувачів буде використано NextAuth. Система повинна підтримувати роботу з обліковими записами, сесіями та ролями користувачів. Розмежування доступу має забезпечувати, щоб адміністративні функції були доступні лише користувачам із відповідними правами.

Для оплати замовлень буде використано Stripe. Система повинна створювати платіжний намір, передавати необхідні дані для оплати та після успішного платежу створювати замовлення в базі даних. Зберігання повних платіжних реквізитів у власній базі даних не передбачається.

Для роботи із зображеннями товарів передбачено завантаження файлів до об'єктного сховища Amazon S3. У базі даних зберігаються посилання на зображення, що дозволить відокремити медіафайли від основних даних магазину.

До нефункціональних вимог належать:

- адаптивність інтерфейсу для різних розмірів екрана;
- зручність навігації;
- валідація даних, що вводяться користувачем;
- надійне зберігання інформації у базі даних;
- повторне використання компонентів інтерфейсу;
- логічне розділення клієнтської та серверної частин;
- можливість подальшого розширення функціональності;
- підтримуваність коду;
- захист адміністративних сторінок від несанкціонованого доступу.

Деталізація ролей користувачів і прав доступу

У системі «Viker» доцільно розглядати три основні ролі: неавторизований відвідувач, зареєстрований покупець і адміністратор. Неавторизований відвідувач матиме доступ до публічної частини сайту: головної сторінки, каталогу, сторінки товару, пошуку, фільтрів і базової навігації. Для нього система повинна бути зрозумілою без попередньої реєстрації, оскільки перше ознайомлення з асортиментом зазвичай відбувається до створення облікового запису.

Зареєстрований покупець отримуватиме доступ до персональних функцій.

					2026.КВР.122.421.16.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		19

До них належать оформлення замовлення, збереження контактних даних, перегляд профілю, зміна налаштувань і можливість залишати відгуки. Наявність облікового запису дозволить пов'язати замовлення з конкретним користувачем і надалі відображати історію взаємодії з магазином. Для покупця важливо, щоб авторизація не ускладнювала процес купівлі, тому сторінки входу й реєстрації мають бути простими й містити зрозумілі повідомлення про помилки.

Адміністратор матиме доступ до службової частини сайту. Він керуватиме товарами, переглядатиме замовлення, змінюватиме статуси й підтримує актуальність каталогу. Доступ адміністратора повинен бути обмежений роллю користувача. Якщо звичайний покупець або неавторизований відвідувач намагатиметься відкрити адміністративний маршрут, система повинна заблокувати доступ або перенаправити користувача до сторінки входу.

Розмежування доступу є важливою нефункціональною вимогою, оскільки адміністративні операції змінюють дані магазину. Неконтрольований доступ до створення, редагування або видалення товарів може призвести до втрати інформації, некоректних цін, пошкодження каталогу або помилок у замовленнях. Тому перевірка ролі користувача має виконуватися не лише на рівні інтерфейсу, а й на рівні серверної логіки.

Деталізація вимог до каталогу товарів

Каталог є основною сторінкою, через яку користувач взаємодітиме з асортиментом. Він повинен відображати перелік велосипедів у вигляді карток, де подано мінімально необхідну інформацію: зображення, назву, ціну, рейтинг або інший індикатор якості, короткі характеристики та доступну дію для переходу до детального перегляду. Картка товару не повинна бути перевантаженою, але має містити достатньо даних для первинного вибору.

Фільтрація за ціною потрібна для того, щоб користувач міг швидко обмежити асортимент відповідно до бюджету. Фільтрація за кольором важлива для велосипедів, оскільки зовнішній вигляд товару часто впливатиме на рішення покупця. Сортування за ціною дозволить переглядати товари від дешевших до

					2026.КВР.122.421.16.00.00 ПЗ	Арк.
						20
Зм.	Арк.	№ докум.	Підпис	Дата		

дорожчих або навпаки. Поєднання фільтрів і сортування повинно працювати передбачувано: зміна одного параметра не має випадково скидати інші активні параметри.

Пагінація потрібна для ситуацій, коли кількість товарів зростає. Вона зменшує навантаження на сторінку й полегшує перегляд результатів. Якщо каталог містить багато позицій, завантаження всіх товарів одразу може погіршити швидкодію. Тому система повинна підтримувати обмеження кількості товарів на сторінці та коректно обчислювати загальну кількість результатів.

Сторінка каталогу повинна коректно обробляти порожні стани. Якщо товари відсутні або фільтри не повертають жодного результату, користувач має побачити зрозуміле повідомлення й можливість змінити параметри пошуку. Відсутність пояснення може бути сприйнята як помилка сайту.

Деталізація вимог до сторінки товару

Сторінка детального перегляду товару повинна надавати повнішу інформацію, ніж картка каталогу. Вона має містити назву велосипеда, галерею зображень, опис, ціну, доступні кольори, характеристики, рейтинг і відгуки. Таке подання дозволить користувачу прийняти рішення про покупку без звернення до зовнішніх джерел.

Вибір кольору й кількості повинен бути пов'язаний із дією додавання до кошика. Якщо товар має кілька кольорів, користувач повинен явно вибрати потрібний варіант або бачити встановлений за замовчуванням колір. Кількість товару повинна змінюватися в допустимих межах. Система не повинна дозволити від'ємну кількість або некоректне значення, яке неможливо обробити під час оформлення замовлення.

Відгуки користувачів допомагатимуть оцінити товар. Для магазину велосипедів це особливо корисно, оскільки покупці можуть звертати увагу на комфорт, якість матеріалів, відповідність опису й практичні враження від використання. Система повинна зв'язувати відгук із користувачем і товаром, щоб відображати його в правильному контексті.

					2026.КВР.122.421.16.00.00 ПЗ	Арк.
						21
Зм.	Арк.	№ докум.	Підпис	Дата		

Якщо товар не знайдено або ідентифікатор сторінки некоректний, система повинна обробити таку ситуацію без аварійного завершення роботи. Користувач має отримати зрозуміле повідомлення або бути перенаправлений до каталогу. Це підвищує надійність і зручність вебсайту.

Деталізація вимог до кошика та оформлення замовлення

Кошик повинен дозволяти користувачу накопичувати вибрані товари перед оформленням покупки. У ньому мають відображатися назви товарів, кількість, вибраний колір і можливість змінити або видалити позицію. Якщо користувач додає той самий товар із тим самим кольором повторно, система може збільшити кількість існуючої позиції. Якщо колір відрізняється, доцільно створювати окремий елемент кошика.

Оформлення замовлення повинно бути поділене на зрозумілі етапи. Перший етап — введення даних для доставки. Другий етап — вибір способу доставки або підтвердження умов отримання товару. Третій етап — оплата. Така послідовність відповідає очікуванням користувача й дозволяє перевіряти дані поступово.

Під час введення даних необхідно виконувати валідацію. Обов'язкові поля не повинні залишатися порожніми, а контактна інформація має бути подана у придатному форматі. Повідомлення про помилки мають бути зрозумілими й розміщуватися поруч із відповідними полями. Це дає змогу користувачу швидко виправити помилку.

Після успішної оплати система повинна створити замовлення в базі даних. Замовлення має містити адресу, користувача, статус, платіжний ідентифікатор і перелік позицій. Якщо оплата не завершилася успішно, замовлення не повинно відображатися як виконане. Така логіка запобігає плутанині між фактично оплаченими й незавершеними покупками.

Деталізація вимог до адміністративної панелі

Адміністративна панель повинна бути зручною для регулярного обслуговування магазину. Вона має надавати адміністратору можливість

					2026.КВР.122.421.16.00.00 ПЗ	Арк.
						22
Зм.	Арк.	№ докум.	Підпис	Дата		

переглядати товари у таблиці, швидко знаходити потрібну позицію, створювати нові товари, редагувати наявні дані та видаляти неактуальні записи. Для кожної операції бажано показувати повідомлення про успішне виконання або помилку.

Під час створення товару адміністратор буде вводити назву, опис, ціну, характеристики й доступні кольори, а також додаватиме зображення. Ці дані повинні бути достатніми для того, щоб товар коректно відображався в каталозі та на сторінці детального перегляду. Якщо частина інформації відсутня, адміністратор повинен бачити повідомлення про необхідність заповнення.

Редагування товару повинно зберігати цілісність даних. Якщо адміністратор змінює ціну або опис, ці зміни мають відобразитися в каталозі. Якщо додаються нові зображення, вони мають бути завантажені до сховища та пов'язані з товаром. Видалення товару потребує особливої обережності, оскільки товар може бути пов'язаний із відгуками або позиціями замовлень.

Робота із замовленнями має дозволяти адміністратору переглядати їхній поточний стан, склад, покупця та адресу доставки. Зміна статусу замовлення повинна бути зрозумілою й контрольованою. Це потрібно для підтримки процесу обробки покупки після її створення.

1.2.4 Вимоги до програмної документації

У межах кваліфікаційної роботи необхідно підготувати програмну документацію, яка описує призначення, структуру, функціональність і порядок використання розробленого інтернет-магазину.

Документація повинна містити:

- опис предметної області та аналіз існуючих рішень;
- технічне завдання на розробку;
- опис структури веб-застосунку;
- опис архітектури клієнтської та серверної частин;
- опис структури бази даних;
- опис основних моделей даних;

					2026.КВР.122.421.16.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		23

- опис основних сторінок і компонентів інтерфейсу;
- опис реалізації авторизації;
- опис роботи з каталогом товарів;
- опис роботи кошика й оформлення замовлення;
- опис інтеграції з платіжною системою;
- опис адміністративної панелі;
- інструкцію з інсталяції програмного забезпечення;
- інструкцію з експлуатації програмного комплексу;
- опис тестування основних функцій.

Документація має бути оформлена українською мовою в академічному стилі відповідно до вимог до кваліфікаційної роботи. Усі назви розділів, підрозділів, рисунків і таблиць мають бути узгоджені зі структурою пояснювальної записки.

1.2.5 Техніко-економічні показники

Розробка інтернет-магазину «Viker» спрямована на підвищення ефективності продажу велосипедів за рахунок автоматизації основних процесів взаємодії з покупцем. Очікуваний ефект від упровадження програмного продукту полягає у зменшенні ручної обробки замовлень, прискоренні оформлення покупки, розширенні доступності магазину для клієнтів і підвищенні зручності адміністрування асортименту.

До основних техніко-економічних показників розробки можна віднести:

- витрати на оплату праці розробника програмного продукту;
- нарахування на заробітну плату працівника, який виконує розробку;
- витрати на електроенергію, спожиту під час розробки та тестування вебсайту;
- витрати на амортизацію комп'ютерної техніки, що використовується для розробки;
- витрати на використання інтернет-з'єднання під час створення та обслуговування системи;

					2026.КВР.122.421.16.00.00 ПЗ	Арк.
						24
Зм.	Арк.	№ докум.	Підпис	Дата		

– витрати на розміщення вебсайту на сервері або хостингу;

– витрати на реєстрацію та підтримку доменного імені;

Кількісні економічні розрахунки, зокрема вартість розробки, витрати на оплату праці, електроенергію, амортизацію, накладні витрати, собівартість, ціну програмного продукту та термін окупності, доцільно навести в економічному розділі кваліфікаційної роботи після погодження вихідних даних.

Таке відокремлення якісних і кількісних показників дає змогу спочатку визначити очікувані переваги вебзастосування, а потім підтвердити їх економічними розрахунками у відповідному розділі.

1.2.6 Стадії та етапи розробки

Розробку інтернет-магазину велосипедів «Biker» доцільно виконувати поетапно, щоб послідовно пройти проєктування, реалізацію та перевірку функціональності.

Основні стадії розробки:

1. Аналіз предметної області.

На цьому етапі визначаються призначення системи, категорії користувачів, основні функції інтернет-магазину, вимоги до каталогу, кошика, замовлень, оплати та адміністративної панелі.

2. Формування технічного завдання.

Визначаються функціональні та нефункціональні вимоги, склад програмної документації, очікувані техніко-економічні показники, порядок контролю та приймання роботи.

3. Проєктування архітектури системи.

Обирається технологічний стек, визначається структура клієнтської та серверної частин, підхід до маршрутизації, робота з базою даних, авторизацією, оплатою та зберіганням зображень.

4. Проєктування бази даних.

Визначаються основні сутності системи: користувач, обліковий запис, сесія,

					2026.КВР.122.421.16.00.00 ПЗ	Арк.
						25
Зм.	Арк.	№ докум.	Підпис	Дата		

велосипед, замовлення, позиція замовлення, відгук, статус. Проєктуються зв'язки між ними.

5. Розробка клієнтської частини.

Створюються сторінки головної, каталогу, детального перегляду товару, кошика, оформлення замовлення, профілю, налаштувань та адміністративної панелі.

6. Розробка серверної логіки.

Реалізуються дії для отримання, створення, редагування та видалення товарів, створення замовлень, оновлення статусів, роботи з відгуками, користувачами та платіжною системою.

7. Інтеграція зовнішніх сервісів.

Налаштовується авторизація, платіжна система Stripe та зберігання зображень в Amazon S3.

8. Тестування програмного продукту.

Перевіряється робота основних сценаріїв: перегляд каталогу, фільтрація, перегляд товару, додавання до кошика, оформлення замовлення, оплата, авторизація, створення й редагування товарів, зміна статусів замовлення.

9. Підготовка програмної документації.

Описуються структура, функції, база даних, інструкція з інсталяції, інструкція користувача та результати тестування.

10. Приймання роботи.

Перевіряється відповідність реалізованого програмного продукту вимогам технічного завдання та готовність до демонстрації.

1.2.7 Порядок тестування та прийому

Тестування вебзастосунку здійснюватиметься шляхом перевірки відповідності реалізованих функцій технічному завданню, контролю роботи клієнтської та серверної частин, перевірки бази даних і виконання основних користувацьких сценаріїв. Метою тестування є підтвердження працездатності

					2026.КВР.122.421.16.00.00 ПЗ	Арк.
						26
Зм.	Арк.	№ докум.	Підпис	Дата		

інтернет-магазину «Viker» перед прийманням роботи.

Під час тестування необхідно перевірити:

- коректність запуску веб-застосунку;
- доступність головної сторінки;
- відображення рекомендованих товарів;
- роботу каталогу велосипедів;
- коректність фільтрації та сортування товарів;
- відкриття сторінки окремого товару;
- відображення фотографій, опису, характеристик, кольорів, ціни та рейтингу;
- додавання товару до кошика;
- зміну кількості товарів у кошику;
- валідацію форми доставки;
- перехід між етапами оформлення замовлення;
- створення платіжного наміру;
- створення замовлення в базі даних;
- роботу авторизації та сесій;
- доступ до профілю користувача;
- створення відгуку;
- роботу адміністративної панелі;
- створення, редагування та видалення товарів;
- завантаження зображень товарів;
- перегляд замовлень адміністратором;
- зміну статусу замовлення;
- коректність зв'язків між таблицями бази даних;
- адаптивність інтерфейсу.

Приймання вебзастосунку здійснюватиметься після демонстрації основних функцій, виконання контрольних сценаріїв тестування та підтвердження відповідності системи вимогам технічного завдання. Результатом приймання є

					2026.КВР.122.421.16.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		27

висновок про те, що інтернет-магазин «Viker» забезпечує заявлену функціональність вебсайту, коректно працює з товарами, користувачами, кошиком, оплатою, замовленнями й адміністративною частиною та може бути представлений як завершена програмна розробка кваліфікаційної роботи.

У загальному розділі проаналізовано предметну область інтернет-магазинів велосипедів, порівняно готові платформи з індивідуальною веброзробкою, сформульовано призначення програмного продукту «Viker», вимоги до функціоналу вебсайту, програмної документації, техніко-економічних показників, етапів розробки та порядку тестування й приймання. Ці вимоги задають основу для подальшого проєктування архітектури, бази даних, інтерфейсу й реалізації програмного забезпечення.

Проведений аналіз предметної області показує, що вебзастосунок для продажу велосипедів повинен поєднувати зручний каталог, зрозумілу навігацію, захищену авторизацію, керування кошиком, оформлення замовлення та адміністративний контроль асортименту. Саме ці функції формують мінімальний набір можливостей, без якого електронний магазин не може повноцінно підтримувати процес продажу товарів.

Вимоги до інтерфейсу і програмної частини взаємопов'язані: покупець має швидко знайти потрібну модель велосипеда, переглянути характеристики, вибрати кількість і завершити оплату, а адміністратор повинен мати інструменти для оновлення товарів та обробки замовлень. Тому під час подальшого проєктування необхідно враховувати не лише окремі сторінки, а й повний маршрут користувача від першого відкриття сайту до підтвердження покупки.

Окрему увагу в наступному проєктуванні слід приділити узгодженості назв сторінок, елементів бази даних і програмних модулів. Якщо ці складові описані однаково в інтерфейсі, схемі даних і коді, подальше тестування стає простішим, а супровід вебзастосунку не потребує постійного уточнення призначення окремих компонентів.

					2026.КВР.122.421.16.00.00 ПЗ	Арк.
						28
Зм.	Арк.	№ докум.	Підпис	Дата		

2 РОЗРОБКА ТЕХНІЧНОГО ТА РОБОЧОГО ПРОЄКТУ

2.1 Розробка структури сайту і вебсторінок

Для вебзастосунку електронної комерції структура сторінок має забезпечувати швидкий доступ до каталогу, детального опису товару, кошика, оформлення замовлення, авторизації, профілю користувача та адміністративної частини.

Логічну карту сторінок вебзастосунку наведено у графічній частині 2026.КВР.122.421.16.00.01 СС. Вона фіксує основні маршрути користувача: перехід від головної сторінки до каталогу, відкриття сторінки товару, додавання позиції до кошика, авторизацію, оформлення замовлення та подальшу роботу з профілем або адміністративною частиною.

Головна сторінка виконує презентаційну і навігаційну функції. На ній розміщується верхня панель із логотипом, основним меню, пошуковим полем, посиланням на кошик та кнопкою входу. Центральний банер привертає увагу до нової колекції, а блок рекомендованих товарів скорочує шлях користувача від першого відвідування сайту до перегляду конкретного велосипеда.

Сторінка каталогу є основним робочим екраном покупця. Вона містить перелік товарів, картки велосипедів, блок фільтрації за ціною і кольором, а також сортування результатів. Каталог повинен підтримувати сценарій швидкого перегляду товарів, порівняння цін і переходу до сторінки детальної інформації. Така структура відповідає типовій логіці інтернет-магазину, де основним об'єктом взаємодії є товарна картка.

Сторінка товару деталізує обрану позицію каталогу. На ній відображаються зображення велосипеда, назва, ціна, рейтинг, опис, характеристики, вибір кольору, кількість товару і кнопка додавання до кошика. Наявність окремої сторінки товару потрібна для того, щоб не перевантажувати каталог і водночас надати користувачеві достатньо інформації для прийняття рішення про покупку.

Кошик і оформлення замовлення утворюють транзакційну частину структури

					2026.КВР.122.421.16.00.00 ПЗ	Арк.
						29
Зм.	Арк.	№ докум.	Підпис	Дата		

сайту. Кошик акумулює вибрані товари, кількість, колір і проміжну вартість, а сторінка оформлення замовлення забезпечує введення адреси доставки, вибір способу доставки та оплату. Якщо користувач не авторизований, вебзастосунок має спрямувати його до входу або реєстрації, оскільки замовлення пов'язується з конкретним обліковим записом.

Блок авторизації містить сторінки входу і реєстрації. Вони забезпечують створення облікового запису, доступ до профілю, перегляд історії замовлень та використання персоналізованих функцій. У межах структури сайту ці сторінки є допоміжними, але критичними для завершення покупки й адміністрування користувацьких даних.

Адміністративна частина відокремлена від публічного інтерфейсу і призначена для керування замовленнями, клієнтами та статусами. Доступ до неї має обмежуватися роллю користувача. Такий розподіл структури дозволяє розмежувати сценарії покупця і адміністратора, а також зменшує ризик випадкового доступу до службових операцій.

Під час побудови структури вебсторінок враховано принципи простоти навігації, логічної послідовності переходів і мінімізації кількості кроків до цільової дії. Користувач повинен перейти від головної сторінки до каталогу, від каталогу до товару, від товару до кошика і від кошика до замовлення без зайвих проміжних екранів.

Інформаційні потоки вебзастосунку пов'язані з такими діями: перегляд товарів, пошук і фільтрація, авторизація, додавання товару до кошика, створення замовлення, оплата, збереження статусу замовлення та адміністрування даних. Ці потоки відображають не лише переходи між сторінками, а й обмін даними між клієнтським інтерфейсом, серверною логікою та базою даних.

Для зручності сприйняття сторінки поділено на публічну, транзакційну та службову групи. До публічної групи належать головна сторінка, каталог і сторінка товару. До транзакційної — кошик, checkout і сторінка успішного замовлення. До службової — сторінки входу, реєстрації, профілю та dashboard адміністратора.

					2026.КВР.122.421.16.00.00 ПЗ	Арк.
						30
Зм.	Арк.	№ докум.	Підпис	Дата		

Такий поділ використовується далі під час опису верстки, бази даних і програмування.

Розроблена структура сайту забезпечує відповідність функціональному призначенню вебзастосунку: демонстрацію асортименту велосипедів, пошук потрібного товару, перегляд характеристик, оформлення покупки та супровід замовлення. Отже, структура сторінок є основою для наступних етапів — верстки, проектування бази даних, програмування та тестування.

Під час деталізації структури сторінок враховано, що вебзастосунок повинен підтримувати не один ізольований сценарій, а повний ланцюг взаємодії покупця. Користувач може почати роботу з головної сторінки, перейти до каталогу, застосувати фільтри, відкрити товар, повернутися до каталогу, додати товар до кошика і перейти до авторизації. Тому структура сторінок проектувалася як послідовність пов'язаних станів, а не як набір незалежних екранів.

Для головної сторінки визначено такі інформаційні блоки: заголовок магазину, коротка пропозиція для користувача, кнопка переходу до каталогу, блок популярних або рекомендованих велосипедів і навігаційна панель. Кожен із цих блоків має окреме призначення. Заголовок формує контекст, кнопка переходу спрямовує до основного сценарію, а рекомендовані товари демонструють асортимент без необхідності одразу використовувати пошук.

Для каталогу визначено структуру, яка підтримує поступове уточнення вибору. Спочатку користувач бачить загальну кількість товарів і сітку карток, далі може застосувати сортування або фільтрацію. Це важливо для магазину велосипедів, оскільки товари відрізняються за ціною, кольором, технічними характеристиками і призначенням. Без фільтрів каталог швидко стає незручним для перегляду.

Сторінка товару структурно пов'язана з каталогом, але має іншу інформаційну щільність. Якщо каталог призначений для порівняння, то сторінка товару призначена для прийняття рішення. Тому на ній розміщуються не лише ціна і назва, а й опис, характеристики, відгуки, рейтинг, вибір кольору і кількості. Такий

					2026.КВР.122.421.16.00.00 ПЗ	Арк.
						31
Зм.	Арк.	№ докум.	Підпис	Дата		

набір даних дає змогу обґрунтувати покупку без переходу до сторонніх джерел.

Кошик виконує роль контрольної сторінки перед оформленням замовлення. Він повинен показувати, які товари вибрано, у якій кількості, з якою ціною і загальною сумою. Якщо користувач змінює кількість або видаляє позицію, структура сторінки має одразу відображати оновлений стан покупки. Це зменшує ризик помилки перед оплатою.

Checkout-сценарій розглядається як окремий логічний блок. Він містить перевірку авторизації, введення адреси доставки, вибір способу доставки, розрахунок вартості і перехід до оплати. У структурі сайту цей блок не повинен бути прихованим або випадковим, оскільки саме він завершує бізнес-процес інтернет-магазину.

Профіль користувача і історія замовлень належать до післяпродажної частини структури. Вони потрібні для того, щоб покупець міг переглянути власні дані, попередні покупки і статуси замовлень. Наявність такого розділу підвищує завершеність вебзастосунку, оскільки покупка не закінчується лише натисканням кнопки оплати.

Адміністративний dashboard у структурі сайту пов'язаний із тими самими даними, що й публічна частина, але має інше призначення. Якщо покупець переглядає товари і створює замовлення, адміністратор контролює замовлення, клієнтів і статуси. Тому службові сторінки мають бути виділені в окремий розділ з обмеженим доступом.

З погляду інформаційної архітектури всі сторінки поділено на чотири групи: презентаційні, товарні, транзакційні та службові. Презентаційні сторінки пояснюють призначення магазину, товарні забезпечують вибір, транзакційні відповідають за покупку, а службові підтримують експлуатацію. Такий поділ робить структуру зрозумілою і для користувача, і для розробника.

Навігаційні переходи повинні мати зворотні маршрути. Користувач, який відкрив сторінку товару, має мати можливість повернутися до каталогу; користувач, який відкрив кошик, має бачити шлях до продовження покупок;

					2026.КВР.122.421.16.00.00 ПЗ	Арк.
						32
Зм.	Арк.	№ докум.	Підпис	Дата		

користувач, який потрапив на форму входу, має розуміти, чому авторизація потрібна для подальшої дії. Це враховано у логіці побудови сторінок.

Для вебзастосунку електронної комерції також важлива узгодженість назв розділів. Назви пунктів меню, заголовки сторінок і підписи кнопок мають відповідати фактичній дії. Наприклад, кнопка переходу до каталогу не повинна вести на сторінку авторизації, а кнопка checkout має з'являтися тоді, коли користувач уже сформував кошик.

Під час аналізу структури враховано можливість подальшого розвитку магазину. До наявних сторінок можна додати категорії велосипедів, сторінку акцій, блог із порадами, сторінку порівняння товарів або розширену аналітику для адміністратора. Поточна структура не суперечить такому розширенню, оскільки основні функціональні групи вже відокремлені.

Отже, структура вебсторінок «Viker» відповідає логіці інтернет-магазину: вона веде користувача від ознайомлення з магазином до вибору велосипеда, формування кошика, авторизації, оформлення замовлення та подальшого супроводу. Схема у графічній частині деталізує ці переходи та показує взаємозв'язок публічних, транзакційних і службових сторінок.

Під час формування структури сайту окремо визначено, які дані переходять між сторінками. Із каталогу на сторінку товару передається ідентифікатор велосипеда, зі сторінки товару до кошика — товар, вибраний колір і кількість, із кошика до checkout — склад замовлення та проміжна сума. Такий опис потоків потрібний для подальшого програмування і тестування.

Структура сторінок також враховує помилкові або неповні дії користувача. Якщо товар не вибрано, кошик порожній або користувач не авторизований, система повинна показати зрозумілий стан і запропонувати наступну дію. Для інтернет-магазину це важливо, оскільки частина користувачів переходить між сторінками не за ідеальним сценарієм.

У публічній частині не передбачено доступу до службових операцій. Користувач бачить лише ті сторінки, які потрібні для вибору і купівлі товару.

					2026.КВР.122.421.16.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		33

Адміністративні сторінки відокремлені, тому їх можна захистити перевіркою ролі й не змішувати з навігацією покупця.

Розроблена структура дає змогу поступово нарощувати функціональність без порушення базової логіки. Наприклад, у майбутньому можна додати сторінку порівняння велосипедів або сторінку акцій, не змінюючи основного маршруту головна сторінка — каталог — товар — кошик — замовлення.

Опис структури показує шлях користувача від головної сторінки до каталогу, товару, кошика й оформлення замовлення, а також відокремлює службові маршрути адміністратора від публічної частини сайту.

2.2 Створення та верстка сторінок сайту

Створення та верстка сторінок вебзастосунку виконуються відповідно до визначеної структури сайту. Основним завданням цього етапу є перетворення логічної структури на зручний інтерфейс, який коректно відображається у браузері, підтримує адаптивність і забезпечує зрозумілу взаємодію користувача з елементами інтернет-магазину.

Інтерфейс застосунку реалізовано у світлому стилі з контрастними кнопками та нейтральними фоновими блоками. Верхня панель містить логотип, навігацію, поле пошуку, кошик і кнопку входу. Така панель розміщується на ключових сторінках і забезпечує єдиний спосіб переходу між основними розділами. Повторюваність навігації підвищує передбачуваність інтерфейсу і зменшує час навчання користувача.

Головна сторінка складається з hero-блоку, кнопок швидкого переходу до каталогу, секції рекомендованих товарів і промо-блоку. Hero-блок виконує рекламну функцію та формує перше враження про магазин. Секція рекомендованих товарів демонструє найважливіші позиції каталогу і дозволяє перейти до детального перегляду без додаткового пошуку.

Каталог побудований за типовою схемою електронної комерції: ліворуч розміщено фільтри, у центральній частині — картки товарів, а над списком —

					2026.КВР.122.421.16.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		34

сортування. Картка товару містить зображення, назву, рейтинг, ціну і кнопку додавання до кошика. Для кожної картки важливо зберігати однакові пропорції, щоб сторінка не втрачала візуального порядку при різній довжині назв або зміні кількості товарів.

Фільтрація в каталозі подана як набір керованих елементів: слайдер діапазону цін і список кольорів. Такі елементи дають змогу звузити перелік товарів без переходу на іншу сторінку. З погляду верстки важливо, щоб панель фільтрів не перекривала картки товарів і залишалася читабельною на різних розмірах екрана.

Сторінка товару має двоколонкову композицію: блок зображення розміщується поруч з інформаційною частиною. В інформаційній частині наведено назву, ціну, рейтинг, опис, вибір кольору, кількість, кнопку додавання до кошика та перелік характеристик. Такий поділ дозволяє одночасно бачити зовнішній вигляд товару і його ключові параметри.

Форма входу побудована у двоколонковому макеті: ліворуч розташовані поля електронної пошти й пароля, кнопка входу та посилання на реєстрацію, праворуч — ілюстративний блок із зображенням велосипеда. Верстка форми має бути простою, оскільки авторизація є проміжним етапом, який не повинен відволікати користувача від основного сценарію покупки.

Для створення сторінок використано компонентний підхід React[2]. Повторювані елементи, зокрема header, картка товару, кнопки, елементи форм, рейтинг і блоки товарів, виділяються в окремі компоненти. Це спрощує підтримку інтерфейсу, дозволяє повторно використовувати однакову логіку і зменшує кількість дубльованого коду.

Адаптивність сторінок забезпечується за рахунок гнучких контейнерів, сітки карток і контрольованих відступів, що належать до базових засобів CSS-верстки [13]. На широкому екрані каталог відображає фільтри й товари паралельно, а на менших екранах структура має перебудовуватися так, щоб зберегти доступність основних дій. Для інтернет-магазину це особливо важливо, оскільки користувач може переглядати товари як з комп'ютера, так і з мобільного пристрою.

					2026.КВР.122.421.16.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		35

Під час верстки враховано візуальну ієрархію: найбільш важливі елементи мають більший розмір, контраст або розташування у верхній частині екрану. Кнопки дії виділяються так, щоб користувач легко знаходив перехід до каталогу, додавання до кошика та оформлення замовлення. Текстові блоки не перевантажуються другорядною інформацією.

Верстка сторінок практично реалізує структуру, визначену у попередньому підрозділі. Інтерфейс поєднує презентаційні сторінки, каталог, товарні картки, checkout-сценарій і форми авторизації в єдиний вебзастосунок.

Додатково під час створення сторінок враховано узгодженість кольорової схеми, однаковість типографіки і повторюваність відступів. Для вебзастосунку інтернет-магазину це важливо тому, що користувач переходить між різними типами сторінок, але повинен сприймати їх як частини одного продукту. Якщо каталог, сторінка товару і форма входу мають різну логіку розміщення елементів, це ускладнює користування сайтом.

Кнопки основних дій розміщуються у місцях, де користувач очікує завершити або продовжити сценарій. На головній сторінці це кнопки переходу до каталогу, у каталозі — кнопки перегляду товару і додавання до кошика, на сторінці товару — кнопка додавання до кошика, а на сторінці входу — кнопка авторизації. Така послідовність формує єдиний поведінковий шаблон.

Під час верстки картки товару потрібно було зберегти баланс між зображенням, назвою, рейтингом і ціною. Зображення привертає увагу, але не повинно витіснити текстові характеристики. Назва і ціна мають бути видимими без додаткового прокручування, оскільки саме ці елементи є ключовими для порівняння товарів у каталозі.

Окремо враховано стан інтерфейсу для неавторизованого користувача. У такому режимі кнопки входу і реєстрації повинні бути легко доступними, а спроба перейти до кошика має супроводжуватися зрозумілим повідомленням. Це дозволяє не блокувати користувача без пояснення і спрямовує його до наступної потрібної дії.

					2026.КВР.122.421.16.00.00 ПЗ	Арк.
						36
Зм.	Арк.	№ докум.	Підпис	Дата		

Верстка сторінки входу повинна підтримувати коректне заповнення форм. Поля введення мають мати зрозумілі підписи, а пароль — можливість показу або приховування. Посилання на відновлення пароля і реєстрацію розміщуються поруч із формою, оскільки вони належать до того самого авторизаційного сценарію.

На етапі створення сторінок також враховано майбутнє наповнення каталогу. Кількість товарів може змінюватися, тому сітка карток не повинна бути жорстко прив'язана до конкретного набору даних. Завдяки цьому магазин можна розширювати без повного перегляду верстки.

Для підтримки доступності[15] елементів інтерфейсу важливо, щоб інтерактивні об'єкти мали достатній розмір і контраст. Користувач повинен безпомилково відрізнити посилання, кнопки, поля вводу, чекбокси і слайдери. Особливо це стосується каталогу, де одночасно відображається велика кількість керованих елементів.

Верстка сторінок визначає не тільки зовнішній вигляд, а й роботу основних сценаріїв: пошук товару, перехід між сторінками, додавання до кошика, авторизацію та оформлення замовлення.

Під час верстки додатково враховано роботу з реальними даними каталогу. Товарні картки можуть мати різну довжину назв, різну кількість зображень і різний набір характеристик, тому елементи сторінки повинні зберігати охайний вигляд незалежно від наповнення. Це досягається однаковими контейнерами, обмеженням розмірів зображень і логічним розміщенням текстових блоків.

Для головної сторінки важливо, щоб перший екран не був перевантажений. Користувач повинен одразу зрозуміти тематику сайту і побачити шлях до каталогу. Саме тому рекламний блок, навігація і рекомендовані товари розміщуються у послідовності, яка не змушує користувача шукати основну дію.

У каталозі фільтри не повинні конкурувати з товарами за увагу. Вони допомагають уточнити вибір, але головним змістом сторінки залишаються велосипеди. Тому панель фільтрації доцільно робити компактною, а картки товарів — достатньо помітними й однаковими за структурою.

					2026.КВР.122.421.16.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		37

На сторінці товару верстка має підтримувати детальне ознайомлення. Зображення, характеристики, опис і блок покупки розміщуються так, щоб користувач міг спочатку оцінити товар, а потім виконати дію. Це відповідає природній послідовності прийняття рішення про покупку.

Форми авторизації і реєстрації мають містити лише необхідні поля. Надмірна кількість елементів на цих сторінках може знизити конверсію, тому структура форми повинна бути короткою і зрозумілою. Додаткові посилання, такі як відновлення пароля або перехід до реєстрації, розміщуються нижче основних полів.

Під час адаптивної верстки важливо не втратити функціональність фільтрів і кошика. На меншому екрані ці елементи можуть змінювати розташування, але повинні залишатися доступними. Це особливо важливо для користувачів, які переглядають каталог зі смартфона.

Підсумовуючи, створення і верстка сторінок забезпечують не лише зовнішній вигляд, а й практичну придатність вебзастосунку. Саме на цьому етапі логічна структура сайту перетворюється на інтерфейс, придатний для щоденного використання покупцем.

2.3 Розробка структури бази даних сайту

База даних вебзастосунку «Biker» проектується відповідно до сутностей предметної області інтернет-магазину. Основними об'єктами є користувачі, велосипеди, замовлення, позиції замовлення, відгуки, статуси, облікові записи та сесії. Такий набір сутностей дозволяє зберігати дані, необхідні для перегляду каталогу, оформлення покупки, авторизації і адміністрування.

Структуру бази даних і зв'язки між основними сутностями подано у графічній частині 2026.КВР.122.421.16.00.00 БД. Діаграма відображає взаємозв'язки між користувачами, товарами, замовленнями, позиціями замовлень, відгуками, статусами та службовими таблицями авторизації.

Для зберігання даних обрано PostgreSQL[5], а взаємодію з базою даних

					2026.КВР.122.421.16.00.00 ПЗ	Арк.
						38
Зм.	Арк.	№ докум.	Підпис	Дата		

реалізовано через Prisma ORM. Prisma[4] дає змогу описати схему даних у файлі `schema.prisma`, генерувати типізований клієнт і виконувати операції створення, читання, оновлення та видалення без написання низькорівневого SQL-коду в більшості типових сценаріїв.

Модель User зберігає інформацію про користувача: ідентифікатор, ім'я, електронну пошту, роль, зображення профілю, пароль, телефон, адресу, місто, країну, дату створення та оновлення. Користувач пов'язаний із сесіями, обліковими записами, відгуками та замовленнями. Наявність ролі дає змогу відокремити звичайного покупця від адміністратора.

Моделі Account, Session і VerificationToken забезпечують роботу механізму авторизації. Account містить дані зовнішнього провайдера авторизації, Session зберігає сесійний токен і час завершення сесії, а VerificationToken може використовуватися для перевірки облікового запису. Ці сутності відокремлені від User, оскільки авторизаційні дані мають власний життєвий цикл.

Модель Bicycle описує товар каталогу. Вона містить назву, опис, ціну, масив зображень, характеристики, кольори, дату створення та оновлення. Зв'язки Bicycle з Review і OrderItem відображають два важливі сценарії: користувач може залишити відгук про товар, а товар може входити до одного або багатьох замовлень.

Модель Order зберігає замовлення користувача. Вона містить посилання на користувача, статус, ідентифікатор платіжного наміру, адресу доставки, спосіб доставки, вартість доставки, загальну суму і дату створення. Через зв'язок із OrderItem замовлення може містити декілька велосипедів із різною кількістю і кольором.

Модель OrderItem виконує роль проміжної сутності між Order і Bicycle. Вона зберігає ідентифікатор замовлення, ідентифікатор велосипеда, кількість і вибраний колір. Така структура потрібна тому, що одне замовлення може містити багато товарів, а один товар може входити до багатьох замовлень.

Модель Review зберігає відгуки користувачів. Вона пов'язана з User і Bicycle,

					<i>2026.KBP.122.421.16.00.00 ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		39

містить заголовок, коментар, рейтинг, кількість корисних позначок і дату створення. Завдяки цьому вебзастосунок може показувати оцінки товарів і відгуки без дублювання даних у таблиці Bicycle.

Модель Status описує можливі стани замовлень. Винесення статусу в окрему таблицю спрощує адміністрування і забезпечує однакове використання назв статусів у всіх замовленнях. У разі розширення системи можна додати нові статуси без зміни структури таблиці Order.

Проектування бази даних виконано з урахуванням нормалізації. Дані користувача, товару, замовлення, позиції замовлення і відгуку не дублюються, а пов'язуються через зовнішні ключі. Це зменшує ризик розсинхронізації даних і спрощує оновлення окремих сутностей.

Фрагмент програмного опису схеми бази даних подано в додатку А. Він показує, як описані сутності реалізовані у файлі `schema.prisma`, зокрема первинні ключі, зовнішні ключі, зв'язки між таблицями та поля, потрібні для роботи каталогу, замовлень і авторизації.

Під час проектування бази даних окремо враховано вимогу збереження історії замовлень. Замовлення не повинно втрачати зв'язок із користувачем і товарами після створення, оскільки ці дані потрібні для перегляду історії покупок, адміністрування і можливого аналізу продажів. Саме тому замовлення виділено в окрему сутність Order.

Використання окремої сутності OrderItem дозволяє уникнути обмеження, коли одне замовлення містить лише один товар. У реальному інтернет-магазині покупець може додати декілька велосипедів або варіантів товару, тому позиція замовлення зберігає конкретну кількість і колір для кожного велосипеда. Це робить структуру бази даних ближчою до практичної моделі електронної комерції.

Для відгуків обрано окрему модель Review, а не текстове поле всередині Bicycle. Це рішення дає змогу зберігати багато відгуків для одного товару, пов'язувати кожен відгук із конкретним користувачем і надалі розширювати функціональність, наприклад додавати модерацію або сортування відгуків за

					2026.КВР.122.421.16.00.00 ПЗ	Арк.
						40
Зм.	Арк.	№ докум.	Підпис	Дата		

датою.

У структурі бази даних передбачено поля часу створення і оновлення для ключових сутностей. Такі поля потрібні для контролю актуальності даних, аналізу активності і технічного супроводу. Наприклад, дата створення замовлення використовується для фільтрації замовлень, а дата оновлення товару — для перевірки актуальності каталогу.

Унікальність електронної пошти користувача запобігає створенню кількох облікових записів з однаковим основним ідентифікатором. Це важливо для авторизації, відновлення доступу і коректного зв'язку замовлень із покупцем. Подібне обмеження на рівні бази даних є надійнішим, ніж перевірка лише в інтерфейсі.

Зовнішні ключі між таблицями забезпечують цілісність даних. Наприклад, позиція замовлення повинна посилатися на існуюче замовлення і на існуючий товар, а відгук — на користувача і велосипед. Якщо ці зв'язки не контролювати, у системі можуть з'явитися записи, які неможливо правильно відобразити в інтерфейсі.

Схема бази даних також враховує можливість подальшого розвитку застосунку. У майбутньому до неї можна додати категорії товарів, знижки, промокоди, обрані товари або розширені параметри доставки. Поточна структура не блокує таке розширення, оскільки основні сутності вже відокремлені одна від одної.

Отже, структура бази даних відповідає логіці інтернет-магазину: користувач переглядає товари, додає їх до кошика, створює замовлення, а система зберігає позиції замовлення, статус, оплату і відгуки. Це забезпечує зв'язок між інтерфейсом, бізнес-логікою і довготривалим зберіганням даних.

При розробці структури бази даних також враховано, що частина даних має довідковий характер. Наприклад, статуси замовлень не повинні вводитися довільним текстом у кожному записі, тому їх доцільно винести в окрему таблицю. Це забезпечує однаковість станів і спрощує подальшу фільтрацію замовлень в

					2026.КВР.122.421.16.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		41

адміністративній частині.

Для товарів важливо зберігати не лише назву і ціну, а й набір характеристик, який може відрізнятися залежно від моделі велосипеда. Тому структура повинна підтримувати гнучке представлення характеристик без дублювання зайвих полів у різних таблицях. У межах реалізованої схеми це забезпечується відповідними полями моделі Bicycle.

Зображення товарів не зберігаються як великі бінарні об'єкти в основних таблицях. Доцільніше зберігати посилання на файли або масив шляхів до зображень. Це зменшує навантаження на базу даних і спрощує використання зовнішнього сховища файлів.

Дані користувача і дані авторизації розділені, оскільки вони мають різне призначення. User описує профіль покупця, а Account і Session забезпечують роботу входу через облікові записи і сесії. Такий розподіл відповідає практиці вебзастосунків, де профіль і механізм авторизації не змішуються в одну таблицю.

У структурі замовлення важливо зберігати суму, адресу доставки і спосіб доставки на момент створення покупки. Якщо ці дані зміняться в профілі користувача пізніше, історичне замовлення повинно залишитися коректним. Це дає змогу відтворити стан покупки на момент її оформлення.

Зв'язки між сутностями дозволяють будувати запити для різних сторінок застосунку. Каталог отримує дані Bicycle, сторінка товару додатково використовує Review, профіль користувача звертається до Order, а адміністративна частина об'єднує Order, User, OrderItem і Status. Тому база даних проектується не ізольовано, а відповідно до потреб сторінок.

Отже, структура бази даних є центральним елементом вебзастосунку. Вона зв'язує користувацький інтерфейс, бізнес-логіку, замовлення, оплату, відгуки і службове адміністрування в єдину систему даних.

2.4 Програмування сайту

Програмування вебзастосунку «Biker» виконано на основі стеку Next.js,

					2026.КВР.122.421.16.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		42

React, TypeScript, Prisma ORM і PostgreSQL. Загальну архітектуру вебзастосунку наведено у графічній частині 2026.KBP.122.421.16.00.02 СС. Програмна структура охоплює основні модулі, взаємодію клієнтської і серверної частин та логіку виконання ключових сценаріїв. Фрагмент серверних дій для роботи з товарами наведено в додатку Б, реалізацію клієнтського стану кошика — у додатку В, серверну дію створення замовлення — у додатку Г, а налаштування авторизації користувачів — у додатку Д.

Застосунок побудовано як вебсистему, що складається з клієнтського інтерфейсу, серверної логіки, бази даних і зовнішніх сервісів. Користувач взаємодіє із застосунком через браузер, клієнтські компоненти відображають сторінки і форми, а серверна частина виконує операції з базою даних, авторизацією, оплатою та файлами.

Next.js[1] використовується як основний фреймворк, який поєднує маршрутизацію, рендеринг сторінок, серверні дії та інтеграцію з React-компонентами. Логіка сторінок розміщується у структурі `src/app`, а окремі функції й компоненти — у відповідних каталогах проєкту.

TypeScript[3] застосовується для типізації даних, параметрів функцій, компонентів і результатів запитів. Для дипломного вебзастосунку це важливо, оскільки помилки типів у кошику, замовленні або роботі з користувачем можуть призвести до некоректної обробки покупки. Типізація зменшує кількість помилок ще на етапі розробки.

Prisma ORM використовується як проміжний рівень між програмним кодом і PostgreSQL. Завдяки Prisma моделі бази даних описуються декларативно, а програмний код працює з типізованим клієнтом. Це спрощує запити до сутностей Bicycle, User, Order, OrderItem, Review і Status.

Серверна частина відповідає за дії, які не повинні виконуватися лише у браузері: створення замовлення, перевірка користувача, доступ до бази даних, робота з оплатою і службові операції адміністратора. Такий поділ зменшує ризик несанкціонованої зміни даних з клієнтського інтерфейсу.

					2026.KBP.122.421.16.00.00 ПЗ	Арк.
						43
Зм.	Арк.	№ докум.	Підпис	Дата		

Інтеграція з NextAuth[6] використовується для авторизації користувачів. Вона забезпечує створення сесій, перевірку поточного користувача і розмежування доступу. У сценаріях покупки авторизація потрібна для прив'язки замовлення до користувача, а в адміністративній частині — для контролю прав доступу.

Оплата реалізується через Stripe. У процесі оформлення замовлення серверна логіка створює платіжний намір і пов'язує його із замовленням. Це дозволяє не зберігати платіжні реквізити користувача у власній базі даних вебзастосунку, а передати обробку платежу спеціалізованому сервісу.

Для роботи із зображеннями товарів передбачено використання Amazon S3. Зберігання файлів у зовнішньому сховищі дає змогу відокремити медіаконтент від основного коду застосунку, спростити масштабування і не перевантажувати базу даних великими бінарними файлами.

Блок-схему оформлення замовлення наведено у графічній частині 2026.КВР.122.421.16.00.00 БС. Вона відображає послідовність дій від перегляду каталогу і вибору товару до авторизації, введення адреси, оплати та створення замовлення. Такий сценарій є ключовим для інтернет-магазину, оскільки поєднує клієнтський інтерфейс, перевірку користувача, серверну логіку, базу даних і платіжний сервіс.

У програмній структурі важливо відокремлювати компоненти інтерфейсу від бізнес-логіки. Компоненти відповідають за відображення кнопок, форм, карток і списків, а серверні дії — за перевірку даних і зміну стану системи. Такий розподіл полегшує тестування і супровід проєкту.

Загалом програмування сайту спрямоване на забезпечення повного циклу електронної комерції: від демонстрації товару до створення замовлення і адміністрування даних. Опис клієнтської та адміністративної частин подано в наступних підпунктах.

Під час програмування особливу увагу приділено розмежуванню відповідальності між рівнями застосунку. Інтерфейс не повинен напряму змінювати критичні дані, а серверна частина не повинна містити зайвих елементів

					2026.КВР.122.421.16.00.00 ПЗ	Арк.
						44
Зм.	Арк.	№ докум.	Підпис	Дата		

відображення. Така архітектура підвищує керованість проєкту і полегшує пошук помилок.

Маршрути застосунку організовано відповідно до структури сторінок. Окремі сторінки каталогу, товару, авторизації, checkout, профілю і dashboard дають змогу підтримувати зрозумілий зв'язок між URL-адресою та функціональним призначенням екрана. Для користувача це означає передбачувану навігацію, а для розробника — зрозумілу структуру коду.

Серверні дії використовуються там, де потрібно перевірити права доступу, звернутися до бази даних або виконати операцію, що змінює стан системи. Наприклад, створення замовлення має виконуватися на сервері, оскільки воно пов'язане з користувачем, позиціями замовлення, оплатою і перевіркою коректності даних.

Помилки під час роботи застосунку мають оброблятися так, щоб користувач отримував зрозуміле повідомлення, а розробник міг встановити причину. Для цього доцільно розділяти помилки валідації, помилки авторизації, помилки доступу до бази даних і помилки зовнішніх сервісів. Такий поділ полегшує супровід вебзастосунку.

Важливою частиною програмування є підготовка даних для відображення. Дані товарів мають надходити у форматі, зручному для карток каталогу і сторінки товару, а дані замовлень — у форматі, придатному для кошика, checkout і адміністративної панелі. Некоректне перетворення даних може призвести до помилок у ціні, кількості або статусі.

Підключення зовнішніх сервісів виконується через конфігурацію середовища. Ключі Stripe, параметри Amazon S3 і секрети авторизації не повинні записуватися безпосередньо в код. Це зменшує ризик витоку конфіденційних даних і дозволяє використовувати різні налаштування для локального та production-середовища.

Після реалізації основних модулів програмний код перевіряється через запуск локального середовища розробки, відкриття основних сторінок і виконання

					2026.KBP.122.421.16.00.00 ПЗ	Арк.
						45
Зм.	Арк.	№ докум.	Підпис	Дата		

користувацьких сценаріїв. Така перевірка дозволяє виявити не лише синтаксичні помилки, а й логічні проблеми у взаємодії компонентів.

Отже, програмування сайту охоплює не лише написання окремих компонентів, а й побудову цілісної системи, де клієнтська частина, серверні дії, база даних, авторизація, оплата і зовнішні сервіси працюють узгоджено.

2.4.1 Написання клієнтської частини

Клієнтська частина вебзастосунку, що реалізується засобами JavaScript-логіки у браузері [14], реалізує взаємодію покупця з магазином. До неї належать головна сторінка, каталог, сторінка товару, кошик, сторінки входу та реєстрації, профіль користувача і checkout-сценарій. Основним завданням клієнтської частини є відображення даних у зручній формі та передавання коректних дій користувача на сервер.

Компоненти клієнтської частини реалізують повторювані елементи інтерфейсу: header, картку товару, рейтинг, кнопки, елементи форми, фільтри, панель кошика і повідомлення. Повторне використання компонентів забезпечує єдність стилю і зменшує обсяг коду, який потрібно підтримувати.

У каталозі клієнтська логіка відповідає за відображення списку товарів, роботу фільтрів, сортування і перехід до детального перегляду. Користувач може обрати ціновий діапазон і колір, після чого інтерфейс має показати відповідні результати. Така логіка покращує зручність пошуку товару.

На сторінці товару клієнтська частина забезпечує вибір кольору і кількості, перегляд характеристик, відгуків та додавання товару до кошика. При цьому інтерфейс повинен не лише показати кнопку додавання, а й контролювати, щоб користувач зробив необхідний вибір перед продовженням сценарію.

Кошик виконує роль проміжного сховища вибраних товарів до створення замовлення. Він показує склад покупки, кількість, ціну і подальшу дію checkout. Якщо користувач не авторизований, застосунок відображає вимогу увійти до облікового запису.

					2026.KBP.122.421.16.00.00 ПЗ	Арк.
						46
Зм.	Арк.	№ докум.	Підпис	Дата		

Клієнтська частина також відповідає за повідомлення про помилки й успішні дії. Наприклад, некоректне заповнення форми, відсутність авторизації або успішне додавання товару мають бути зрозуміло відображені. Це зменшує кількість непередбачуваних ситуацій для користувача.

Отже, клієнтська частина вебзастосунку забезпечує основні сценарії покупця: перегляд, пошук, вибір, додавання до кошика, авторизацію і перехід до замовлення. Її якість безпосередньо впливає на практичну придатність інтернет-магазину.

Під час реалізації клієнтської частини важливо забезпечити миттєвий візуальний відгук на дії користувача. Натискання кнопок, вибір кольору, зміна кількості, відкриття кошика або перехід до іншої сторінки мають супроводжуватися зрозумілою зміною інтерфейсу. Це підвищує довіру до застосунку і зменшує кількість повторних натискань.

Стан окремих елементів інтерфейсу повинен бути узгоджений з даними, які надходять із серверної частини. Наприклад, якщо товар недоступний або користувач не авторизований, кнопки і повідомлення мають відображати саме цей стан, а не створювати враження, що операція може бути виконана без обмежень.

У клієнтській частині важливо підтримувати єдину структуру обробки форм. Поля введення, повідомлення про помилки і кнопки підтвердження повинні мати однакову логіку на сторінках входу, реєстрації, профілю і checkout. Це спрощує користування сайтом і полегшує подальше розширення функціональності.

Для каталогу особливо важлива продуктивність відображення. Якщо кількість товарів збільшується, сторінка не повинна ставати незручною для перегляду. Тому сітка товарів, фільтри і сортування мають бути реалізовані так, щоб користувач швидко отримував потрібний результат.

2.4.2 Написання адміністративної частини

Адміністративна частина вебзастосунку призначена для керування службовими даними магазину. Вона відокремлена від публічного інтерфейсу і має

					2026.КВР.122.421.16.00.00 ПЗ	Арк.
						47
Зм.	Арк.	№ докум.	Підпис	Дата		

бути доступна лише користувачам з відповідними правами. Основними об'єктами адміністрування є замовлення, клієнти, статуси та, за потреби, товарні позиції.

Для admin-частини важливим є контроль доступу. Перед відкриттям службових сторінок система повинна перевірити авторизацію і роль користувача. Якщо доступ не підтверджено, користувач не повинен бачити адміністративні операції. Це є базовою вимогою безпеки для вебзастосунку електронної комерції.

Сторінка замовлень дозволяє переглядати створені покупки, їхній стан, загальну суму, користувача і дату створення. Адміністратор повинен мати змогу контролювати процес обробки замовлення, змінювати або перевіряти статус, а також аналізувати активність покупців.

Сторінка клієнтів відображає користувачів, які взаємодіють із магазином. Такі дані потрібні для супроводу замовлень, комунікації, аналізу повторних покупок і перевірки коректності профілів. У службовому інтерфейсі не повинні відображатися зайві або чутливі дані, якщо вони не потрібні адміністратору.

Сторінка статусів використовується для підтримки однакового переліку станів замовлення. Окрема модель Status у базі даних дозволяє не дублювати назви статусів у кожному замовленні. Це спрощує підтримку системи і робить дані більш узгодженими.

Admin-частина має використовувати ті самі серверні механізми доступу до бази даних, що й інші службові операції. Пряме редагування даних з клієнтського інтерфейсу без перевірки на сервері є небезпечним, тому зміни мають проходити через контрольовані функції.

Адміністративна частина доповнює публічний магазин службовими засобами керування. Вона потрібна для супроводу замовлень, користувачів і статусів після того, як покупець завершив основний сценарій взаємодії з сайтом.

У службовій частині важливо мінімізувати кількість ручних помилок адміністратора. Для цього дані мають бути подані у зрозумілих таблицях або списках, а дії зі зміни статусів повинні мати чітке призначення. Адміністратор не повинен виконувати операції з неочевидними наслідками.

					<i>2026.КВР.122.421.16.00.00 ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		48

Адміністративні сторінки повинні бути відокремлені маршрутизаційно і логічно. Це дозволяє простіше перевіряти доступ, журналювати службові дії і надалі розширювати панель керування без зміни публічного інтерфейсу магазину.

Під час роботи з даними замовлень потрібно враховувати, що вони пов'язані з оплатою і користувацькою інформацією. Тому admin-частина не повинна дозволяти випадково порушити зв'язки між замовленням, користувачем, статусом і позиціями замовлення.

Отже, admin-частина є окремим функціональним блоком, який забезпечує експлуатацію магазину після запуску. Без неї вебзастосунок міг би відобразити товари, але не забезпечував би повноцінний супровід замовлень і користувачів.

2.5 Тестування вебсайту

Тестування вебсайту «Viker» виконується для перевірки відповідності реалізованого застосунку технічному завданню, працездатності основних сценаріїв і коректності взаємодії між інтерфейсом, серверною логікою та базою даних. Для інтернет-магазину критичними є сценарії перегляду каталогу, вибору товару, авторизації, додавання до кошика та оформлення замовлення. План тестування вебсайту наведено в додатку Е.

Під час тестування перевіряються функціональні, інтерфейсні та інтеграційні аспекти. Функціональне тестування підтверджує, що кнопки, форми, фільтри і переходи виконують очікувані дії. Інтерфейсне тестування оцінює зручність розміщення елементів. Інтеграційне тестування перевіряє роботу клієнтської частини із серверними діями, базою даних і зовнішніми сервісами.

Першим перевіряється відкриття головної сторінки. На ній мають бути доступні логотип, головне меню, пошукове поле, кнопки переходу до каталогу, блок рекомендованих товарів і кнопки додавання до кошика. Відображення головної сторінки підтверджує, що базова маршрутизація і завантаження товарних даних працюють коректно.

Другим перевіряється сторінка каталогу. Основні критерії: відображення

					2026.КВР.122.421.16.00.00 ПЗ	Арк.
						49
Зм.	Арк.	№ докум.	Підпис	Дата		

кількості товарів, робота сортування, наявність фільтрів за ціною і кольором, коректність карток товарів і можливість переходу до сторінки товару. Каталог є найважливішою сторінкою для пошуку товару, тому його перевірка має високий пріоритет.

Третім перевіряється сторінка товару. На ній мають відобразитися зображення, назва, ціна, рейтинг, опис, характеристики, вибір кольору, лічильник кількості і кнопка додавання до кошика. Окремо перевіряється, що користувач може вибрати колір і змінити кількість перед додаванням товару.

Четвертим перевіряється сценарій кошика. Якщо користувач не авторизований, вебзастосунок має повідомити про необхідність входу до облікового запису. Така поведінка була зафіксована під час перевірки: після звернення до кошика відображається панель `Sign in to continue`, що пояснює необхідність авторизації для перегляду кошика і здійснення покупки.

П'ятим перевіряється сторінка входу. Вона повинна містити поле електронної пошти, поле пароля, кнопку показу пароля, кнопку входу, посилання на відновлення пароля, кнопку Google і посилання на реєстрацію. Наявність цих елементів підтверджує готовність авторизаційного сценарію до використання.

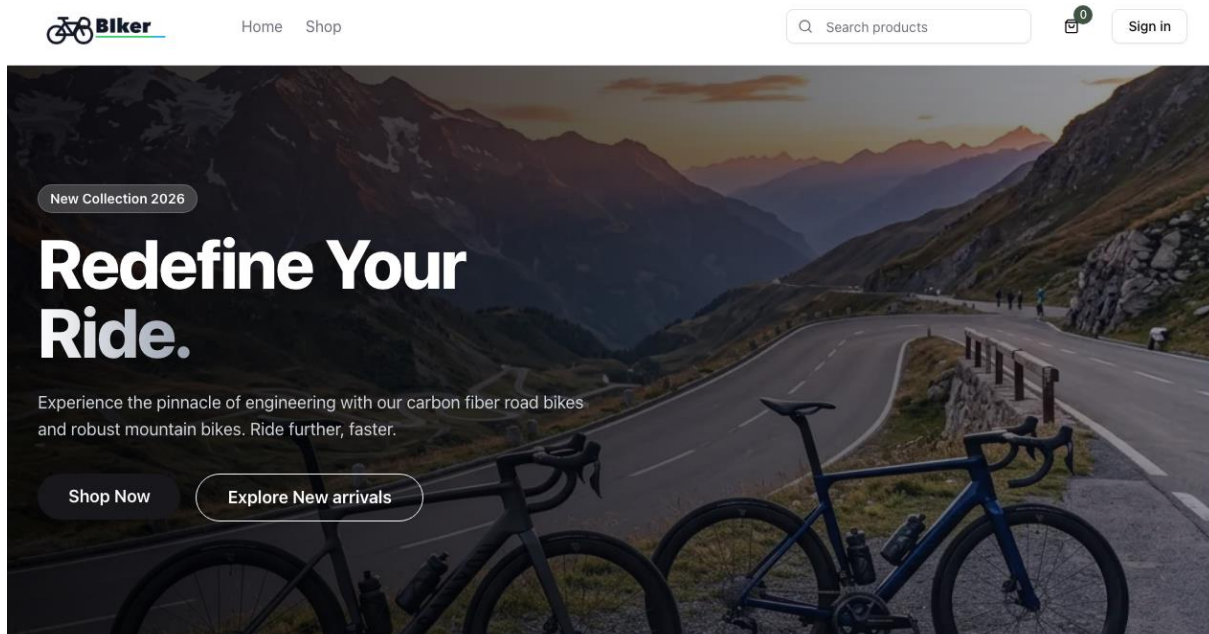
Шостим перевіряється сторінка керування товарами адмін-панелі. На ній мають бути доступні бічне меню, пошукове поле, вкладки станів товарів, кнопка додавання нового товару, таблиця зображень, назв, кількості відгуків, ціни, продажів, дати створення і меню дій. Відображення цієї сторінки підтверджує, що захищений маршрут адміністратора працює коректно, а товарні дані завантажуються у режимі керування.

Сьомим перевіряється сторінка замовлень адмін-панелі. Основні критерії: відображення покупця, адреси доставки, поточного статусу, кількості товарів, дати створення замовлення і меню адміністративних дій. Наявність цих елементів підтверджує, що адміністратор може переглядати оформлені замовлення і контролювати їхній стан.

Для фіксації результатів тестування підготовлено скріншоти реального

					2026.КВР.122.421.16.00.00 ПЗ	Арк.
						50
Зм.	Арк.	№ докум.	Підпис	Дата		

вебзастосунку у світлій темі інтерфейсу. Головну сторінку вебзастосунку наведено на рисунку 2.1.



Featured Rides

Discover our hand-picked selection of top-rated bicycles, designed for unparalleled performance and aesthetic excellence.

[View All Products →](#)

Рисунок 2.1 – Головна сторінка вебзастосунку «Biker» у світлій темі

Сторінку каталогу з фільтрами, сортуванням і картками товарів наведено на рисунку 2.2.

Під час перевірки головної сторінки контролювалися коректність верхньої навігації, доступність переходу до каталогу, відображення товарних карток і стабільність роботи інтерактивних елементів. Окремо перевірено, що сторінка завантажується у світлій темі без зсуву блоків, а основні посилання залишаються видимими для користувача. Це підтверджує придатність головної сторінки як стартової точки для подальшого сценарію купівлі.

Також оцінювалася логічність розміщення інформаційних блоків: користувач повинен одразу бачити призначення магазину, доступні переходи та приклади товарів. Завдяки цьому головна сторінка не лише презентує вебзастосунок, а й

					2026.КВР.122.421.16.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		51

спрямовує покупця до подальшого перегляду каталогу.

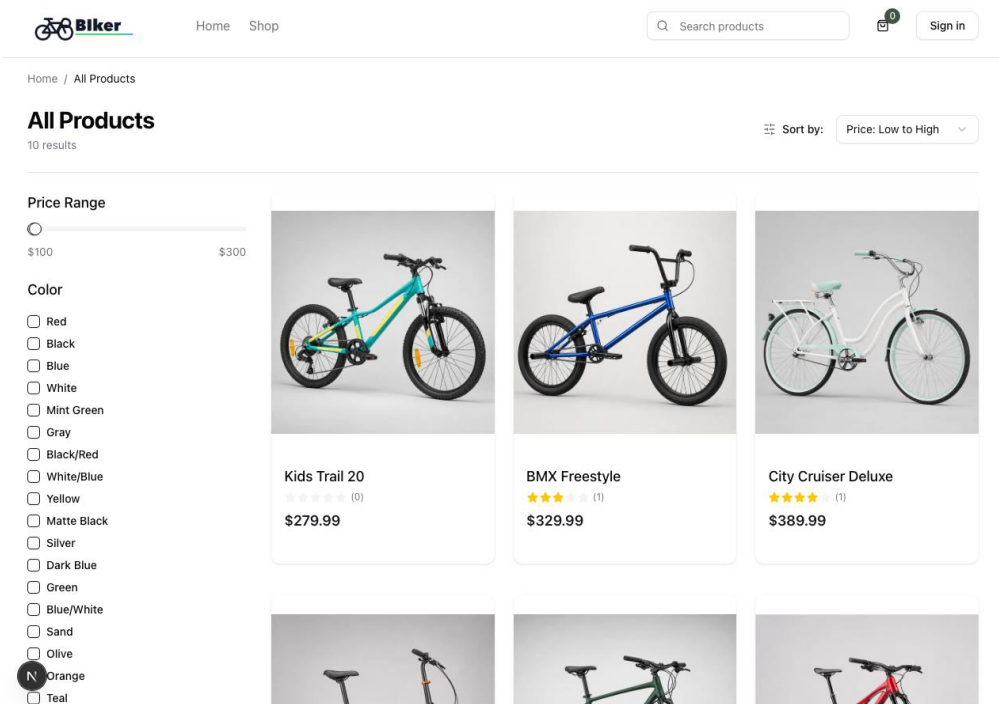


Рисунок 2.2 – Каталог товарів вебзастосунку «Biker» у світлій темі

Сторінку окремого товару з вибором кольору, кількості та кнопкою додавання до кошика наведено на рисунку 2.3.

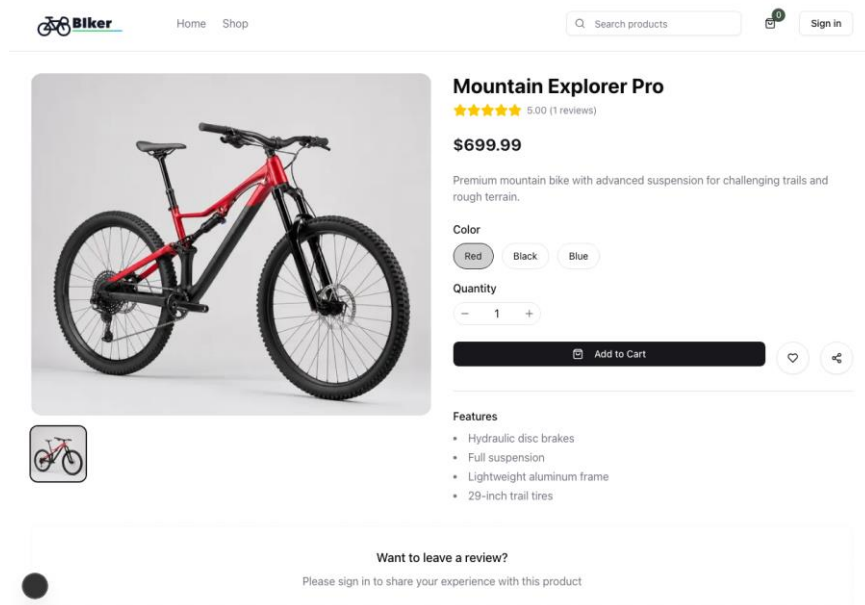


Рисунок 2.3 – Сторінка товару вебзастосунку «Biker» у світлій темі

					2026.КВР.122.421.16.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		52

Сторінку авторизації, яка використовується для перевірки сценарію входу користувача, наведено на рисунку 2.4.

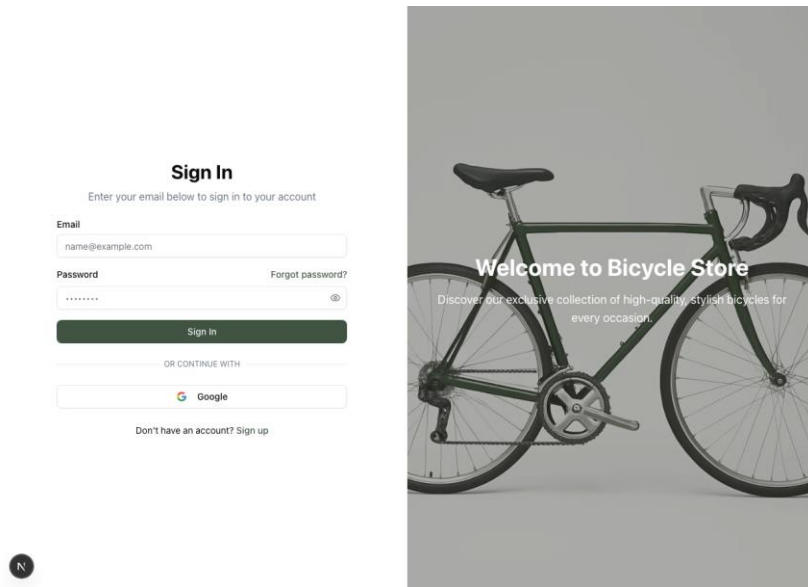


Рисунок 2.4 – Сторінка входу вебзастосунку «Biker» у світлій темі

Сторінку керування товарами адмін-панелі вебзастосунку наведено на рисунку 2.5.

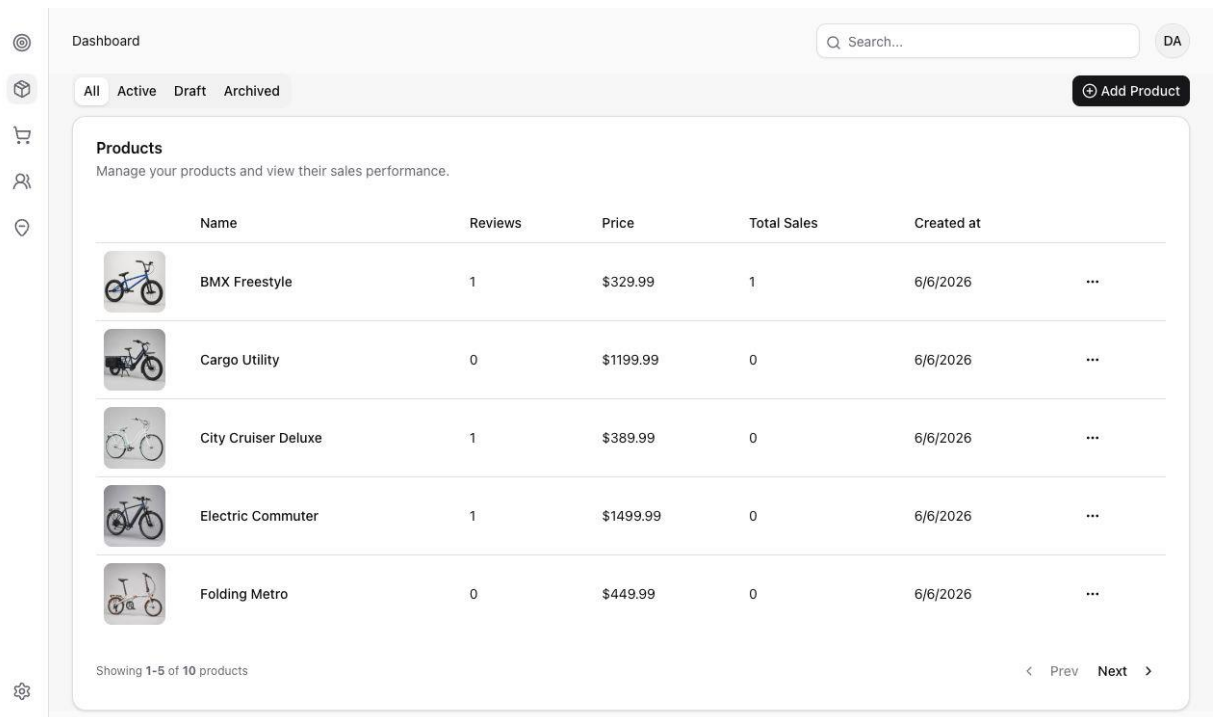


Рисунок 2.5 – Сторінка керування товарами адмін-панелі

Сторінку замовлень адмін-панелі вебзастосунку наведено на рисунку 2.6.

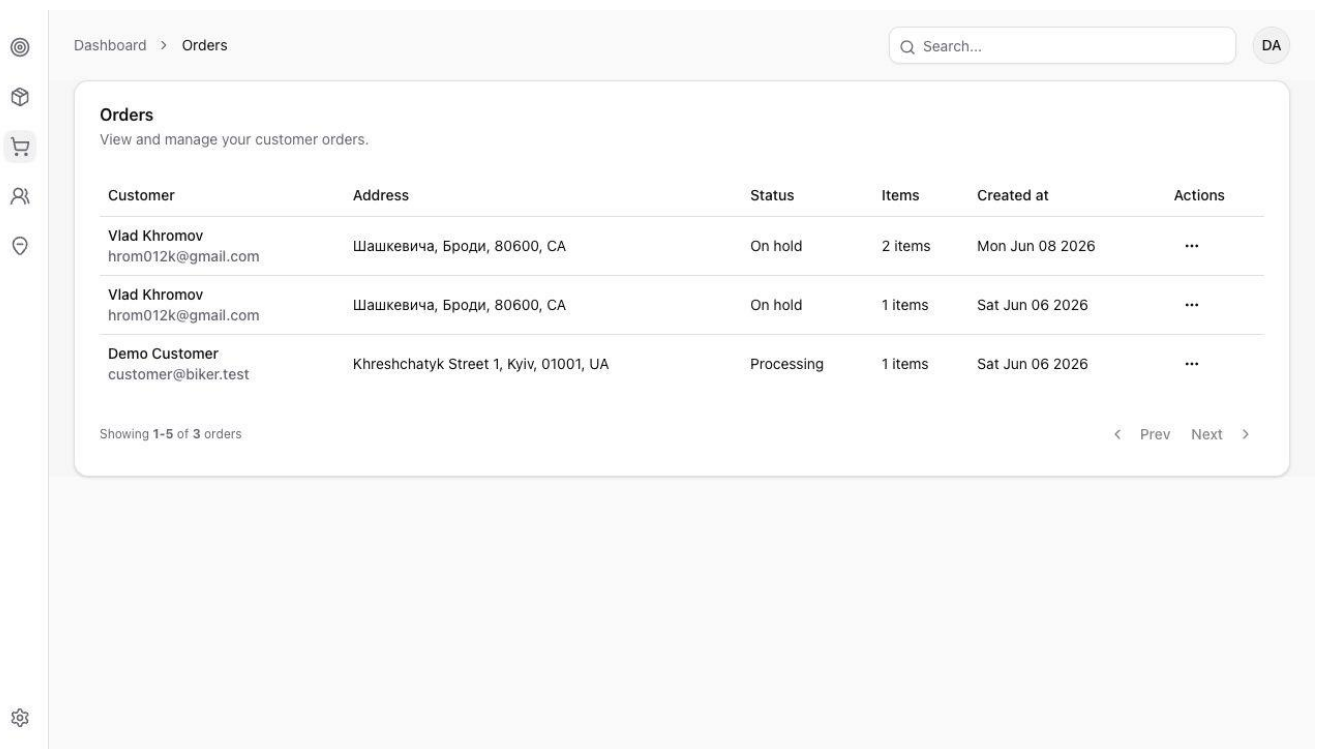


Рисунок 2.6 – Сторінка замовлень адмін-панелі

Сторінку клієнтів адмін-панелі вебзастосунку наведено на рисунку 2.7.

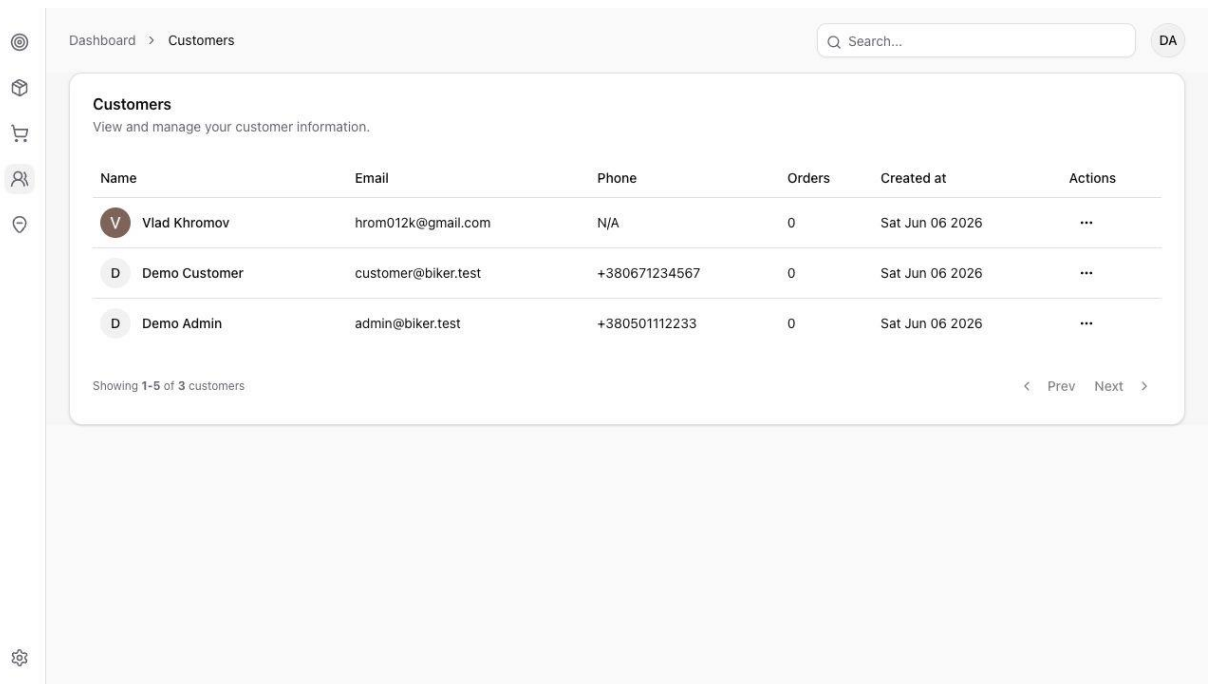


Рисунок 2.7 – Сторінка клієнтів вебзастосунку

Сторінку статусів адмін-панелі вебзастосунку наведено на рисунку 2.8.

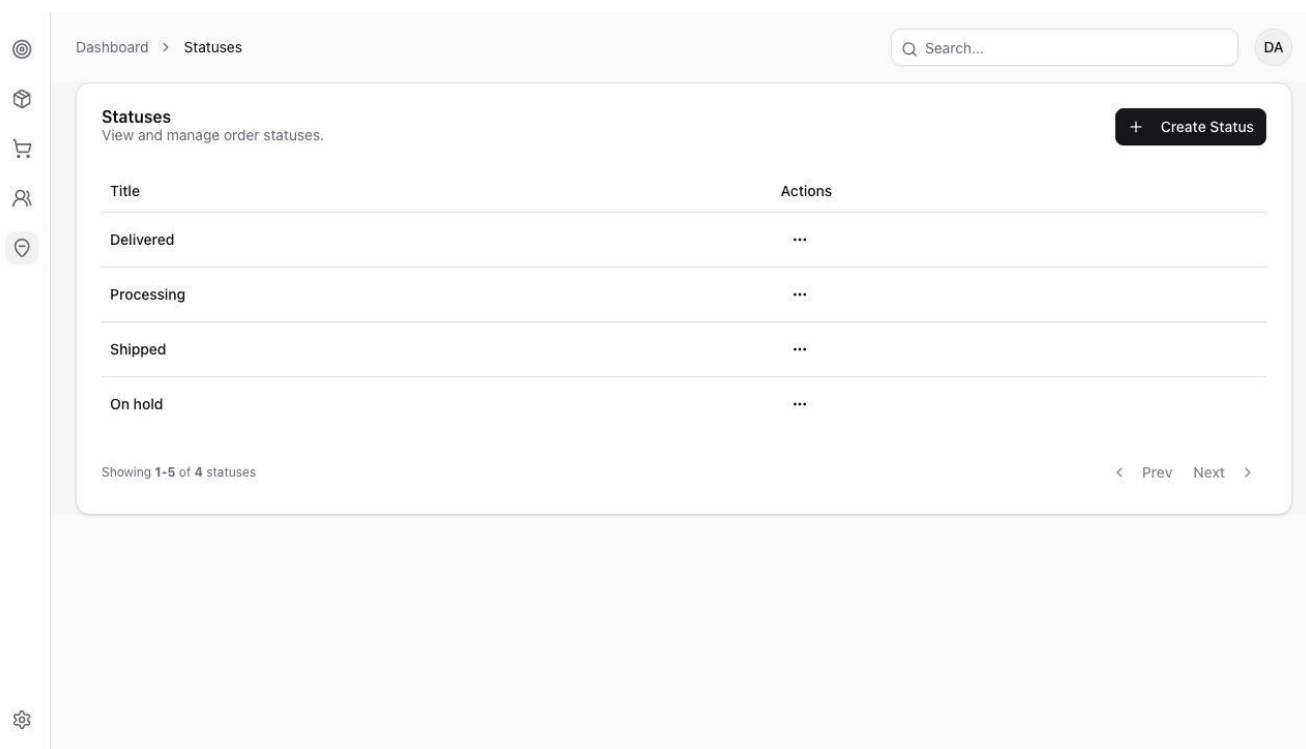


Рисунок 2.8 – Сторінка статусів адмін-панелі

Сторінку налаштувань вебзастосунку наведено на рисунку 2.9.

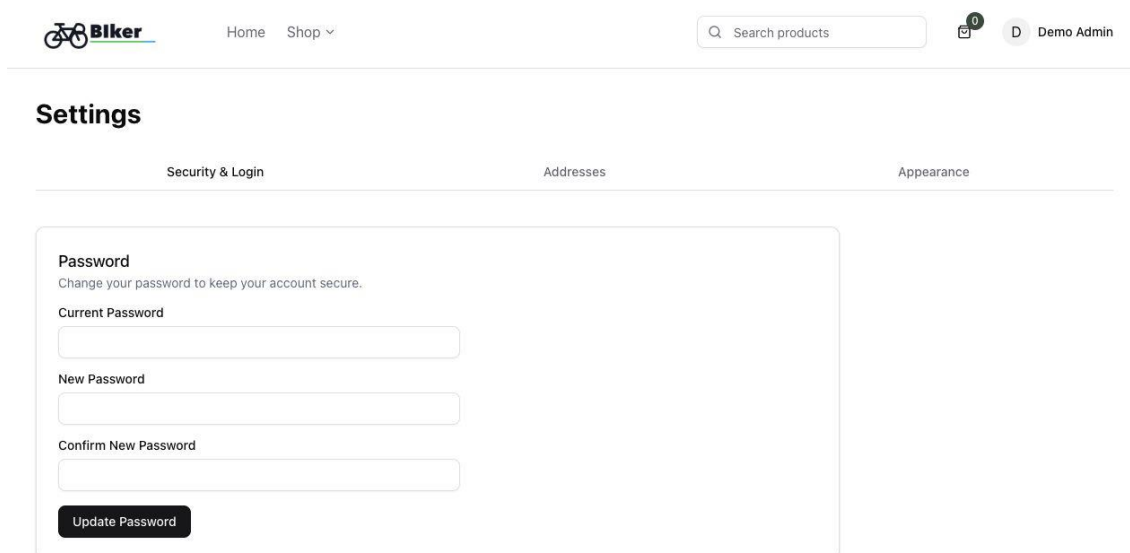


Рисунок 2.9 – Сторінка налаштувань вебзастосунку

Перелік базових тестових сценаріїв вебсайту наведено в таблиці 2.1. Ця таблиця використовується як підсумкова форма перевірки основних функціональних можливостей.

Таблиця 2.1 – Перелік тестових сценаріїв

№	Сценарій	Очікуваний результат
1	Відкрити головну сторінку	Відображаються навігація, промо-блок і рекомендовані товари
2	Перейти до каталогу	Відображається список товарів, фільтри, сортування та пагінація
3	Змінити діапазон цін	Список товарів оновлюється відповідно до фільтра
4	Вибрати колір	У каталозі залишаються товари відповідного кольору
5	Відкрити сторінку товару	Показано галерею, назву, рейтинг, ціну, опис і характеристики
6	Змінити кількість товару	Значення кількості збільшується або зменшується
7	Додати товар до кошика	Для авторизованого користувача товар додається до кошика; для неавторизованого показується вимога входу
8	Авторизуватися	Після успішного входу користувач отримує доступ до персональних функцій
9	Оформити замовлення	Після заповнення даних і оплати створюється замовлення
10	Змінити статус замовлення адміністратором	Статус оновлюється в базі даних

За результатами виконання тестових сценаріїв основні функції вебзастосунку працюють узгоджено: користувач може перейти від перегляду каталогу до сторінки товару, змінити кількість, додати позицію до кошика та завершити оформлення замовлення.

					2026.КВР.122.421.16.00.00 ПЗ	Арк.
						56
Зм.	Арк.	№ докум.	Підпис	Дата		

3 СПЕЦІАЛЬНИЙ РОЗДІЛ

3.1 Інструкція з розміщення сайту в Інтернеті

Розміщення вебзастосунку «Viker» в Інтернеті передбачає підготовку програмного коду, налаштування змінних середовища, підключення бази даних, виконання міграцій і запуск production-версії застосунку. Оскільки застосунок побудовано на Next.js, розгортання може виконуватися на платформі, що підтримує Node.js і серверний рендеринг.

Перед розміщенням потрібно перевірити актуальність вихідного коду, наявність усіх залежностей і відповідність конфігурацій production-середовищу. У репозиторії мають бути файли `package.json`, `package-lock.json`, `prisma/schema.prisma`, конфігураційні файли Next.js і TypeScript, а також каталог `src` із програмним кодом застосунку.

Першим етапом є встановлення залежностей. Для цього у каталозі проєкту виконується команда `npm install`. Вона встановлює бібліотеки, потрібні для роботи Next.js, React, Prisma, NextAuth, Stripe, AWS SDK та інших компонентів. Після встановлення залежностей бажано перевірити, що не виникло конфліктів версій.

Другим етапом є налаштування змінних середовища. До них належать рядок підключення до PostgreSQL, секрети авторизації, параметри Stripe, налаштування Amazon S3 та URL production-сайту. Значення секретів не повинні потрапляти до пояснювальної записки або відкритого репозиторію.

Третім етапом є генерація Prisma Client і виконання міграцій бази даних. Prisma Client потрібний для типізованої взаємодії програмного коду з базою даних. Якщо база даних створюється вперше, потрібно застосувати міграції або синхронізувати схему відповідно до обраної стратегії розгортання.

Четвертим етапом є складання production-версії застосунку командою `npm run build`. Під час складання перевіряється коректність маршрутизації, імпортів, типів, серверних дій і доступу до схеми Prisma. Якщо на цьому етапі виникає

					2026.КВР.122.421.16.00.00 ПЗ	Арк.
						57
Зм.	Арк.	№ докум.	Підпис	Дата		

помилка, розгортання потрібно зупинити й усунути причину.

П'ятим етапом є запуск застосунку командою `npm run start` або розміщення на хмарній платформі, яка самостійно виконує запуск production-збірки. Після запуску перевіряється доступність головної сторінки, каталогу, сторінки товару, авторизації та службових маршрутів.

Після розміщення потрібно перевірити HTTPS-з'єднання, коректність доменного імені, доступ до бази даних, роботу авторизації, платежів і завантаження зображень. Окремо контролюється, щоб production-сайт не використовував тестові або локальні URL-адреси.

У разі використання Vercel або подібної платформи процес розміщення може бути автоматизований через підключення репозиторію. Після кожного оновлення гілки система збирає застосунок і публікує нову версію. Проте навіть за автоматичного розгортання необхідна ручна перевірка основних сценаріїв після релізу.

Отже, розміщення сайту в Інтернеті складається з підготовки коду, налаштування середовища, підключення бази даних, складання production-версії, запуску і післярелізної перевірки. Виконання цих етапів забезпечує доступність вебзастосунку для користувачів.

Перед публікацією необхідно створити production-базу даних і переконатися, що доступ до неї дозволений лише застосунку та уповноваженим адміністраторам. Рядок підключення має зберігатися у змінних середовища платформи розгортання, а не у вихідному коді.

Після налаштування бази даних потрібно перевірити, що Prisma бачить актуальну схему і може згенерувати клієнт. Якщо структура таблиць не відповідає схемі `schema.prisma`, застосунок може зібратися, але помилки з'являться під час виконання запитів. Тому перевірка бази даних виконується до публікації сайту.

Окремо налаштовуються параметри авторизації. Необхідно задати секрет сесій, URL застосунку і параметри провайдерів входу. Якщо ці значення залишити локальними, production-сайт може перенаправляти користувача на неправильну

					2026.КВР.122.421.16.00.00 ПЗ	Арк.
						58
Зм.	Арк.	№ докум.	Підпис	Дата		

адресу або створювати недійсні сесії.

Для інтеграції оплати потрібно використовувати production- або тестові ключі Stripe[7] відповідно до режиму запуску. Не можна змішувати тестові ключі з production-адресами без чіткого розуміння наслідків, оскільки це ускладнює перевірку платежів і може призвести до неправильного статусу замовлення.

Під час розміщення також перевіряється доступність статичних файлів і зображень. Якщо зображення товарів зберігаються у зовнішньому сховищі Amazon S3[8], потрібно переконатися, що production-домен має доступ до них і браузер не блокує завантаження через неправильні CORS-налаштування.

Після першого запуску production-версії виконується контрольний обхід сторінок: головна сторінка, каталог, сторінка товару, сторінка входу, кошик, checkout і службові сторінки. Якщо хоча б один із цих маршрутів не працює, сайт не можна вважати готовим до використання.

Для стабільної експлуатації потрібно налаштувати журналювання помилок. Після розгортання помилки можуть виникати через мережу, базу даних, сторонні сервіси або неправильні дані користувача. Журнали дозволяють швидко виявити джерело проблеми без повторення всього сценарію вручну.

Інструкція з розміщення описує повну підготовку production-середовища: базу даних, секрети, авторизацію, оплату, файли, домен, HTTPS і післярелізну перевірку. Простого запуску команди збірки для готовності інтернет-магазину недостатньо.

Перед перенесенням сайту з локального середовища до production потрібно перевірити файл конфігурації змінних середовища. У ньому мають бути задані окремі значення для бази даних, авторизації, платіжної системи, сховища зображень і базової адреси сайту. Локальні адреси на зразок localhost не повинні залишатися в production-конфігурації.

Після встановлення залежностей доцільно виконати перевірку типів і локальне складання проєкту. Якщо команда складання завершується з помилкою, розміщення сайту потрібно відкласти, оскільки production-середовище зазвичай не

					2026.КВР.122.421.16.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		59

повинно містити незавершених або частково працездатних модулів.

Для бази даних потрібно заздалегідь створити production-екземпляр PostgreSQL і перевірити доступ до нього з платформи розгортання. Якщо база даних недоступна, сайт може відкрити статичні сторінки, але каталог, замовлення, відгуки або авторизація працюватимуть некоректно.

Під час першого запуску необхідно перевірити не лише головну сторінку, а й маршрути, які використовують серверні дії. До таких маршрутів належать каталог, сторінка товару, кошик, checkout, профіль і адміністративні сторінки. Це дозволяє виявити помилки, які не видно під час простого відкриття стартової сторінки.

Після підключення доменного імені перевіряється робота HTTPS. Захищене з'єднання є обов'язковим для інтернет-магазину, оскільки користувач вводить персональні дані, адресу доставки і може переходити до оплати. Наявність HTTPS також впливає на довіру користувачів і коректну роботу частини зовнішніх сервісів.

Остаточним етапом є контрольне тестування production-версії за скороченим сценарієм: відкрити головну сторінку, перейти до каталогу, застосувати фільтр, відкрити товар, перейти до кошика або авторизації та перевірити доступність службових маршрутів. Лише після цього сайт можна вважати розміщеним в Інтернеті.

3.2 Інструкція з обслуговування та наповнення сайту

Обслуговування вебзастосунку «Viker» охоплює регулярне оновлення товарів, контроль замовлень, перевірку працездатності сервісів, резервне копіювання даних і підтримку актуальності програмних залежностей. Для інтернет-магазину обслуговування є постійним процесом, оскільки дані каталогу, замовлення і користувацькі дії змінюються щоденно.

Наповнення сайту починається з підготовки інформації про велосипед. Для кожного товару потрібно вказати назву, опис, ціну, кольори, характеристики та зображення. Інформація має бути достатньо повною, щоб користувач міг порівняти

					2026.КВР.122.421.16.00.00 ПЗ	Арк.
						60
Зм.	Арк.	№ докум.	Підпис	Дата		

товари і прийняти рішення про покупку без додаткових уточнень.

Зображення товарів повинні бути якісними, однакового стилю і розміру. Якщо для зберігання використовується Amazon S3, потрібно контролювати правильність URL-адрес, доступність файлів і відсутність битих посилань. Непрацюючі зображення погіршують сприйняття каталогу і можуть знизити довіру до магазину.

Під час додавання або редагування товару необхідно перевіряти, що дані коректно відображаються в каталозі й на сторінці товару. Після зміни ціни, кольорів або характеристик слід оновити сторінку каталогу, виконати пошук і перейти до детального перегляду, щоб переконатися у відсутності помилок.

Обслуговування замовлень передбачає перевірку нових покупок, контроль статусу, обробку оплати і зв'язок із покупцем у разі потреби. Статуси замовлень повинні оновлюватися своєчасно, оскільки вони відображають стан виконання покупки і впливають на інформування користувача.

Періодично потрібно перевіряти облікові записи користувачів і коректність профільних даних. Якщо користувачі звертаються із проблемами входу, доставки або історії замовлень, адміністратор має перевірити відповідні записи в системі й усунути причину збою.

Резервне копіювання бази даних є обов'язковою частиною обслуговування. Резервні копії мають створюватися за графіком і зберігатися окремо від основного середовища. Періодично потрібно виконувати пробне відновлення, оскільки резервна копія без перевірки відновлення не гарантує захисту від втрати даних.

Оновлення програмних залежностей виконується обережно. Перед оновленням потрібно переглянути зміни у бібліотеках, виконати локальне складання і перевірити основні сценарії. Особливу увагу слід приділяти пакетам, пов'язаним з авторизацією, оплатою, базою даних і UI-компонентами.

Моніторинг сайту має охоплювати доступність сторінок, час відповіді, помилки сервера, помилки авторизації, успішність платежів і доступ до зовнішніх сервісів. Якщо виникають помилки, їх потрібно фіксувати, аналізувати і усувати з

					2026.КВР.122.421.16.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		61

урахуванням впливу на користувача.

Обслуговування та наповнення сайту включають роботу з товарами, замовленнями, користувачами, зображеннями, резервними копіями, залежностями і моніторингом. Регулярне виконання цих дій підтримує стабільну роботу вебзастосунку після розміщення в Інтернеті.

Під час наповнення каталогу потрібно дотримуватися єдиного стилю назв товарів. Назва має бути короткою, але достатньо інформативною: тип велосипеда, модель або призначення. Якщо назви будуть неоднорідними, користувачеві буде складніше порівнювати товари у каталозі.

Опис товару має містити практичну інформацію, а не лише рекламні фрази. Для велосипеда важливими є призначення, тип рами, гальма, підвіска, розмір коліс, особливості використання і переваги для покупця. Такий опис допомагає користувачеві обрати товар без звернення до адміністратора.

Ціну потрібно оновлювати централізовано і перевіряти її відображення в каталозі, на сторінці товару, у кошику і при оформленні замовлення. Якщо ціна в різних частинах сайту відрізняється, це створює помилку бізнес-логіки і може викликати претензії користувача.

Після додавання нових товарів адміністратор повинен перевірити їх наявність у загальному каталозі, роботу фільтрів і коректність переходу до сторінки товару. Це дозволяє виявити помилки ще до того, як їх побачить покупець.

Замовлення потрібно обробляти за встановленим регламентом. Спочатку перевіряється факт створення замовлення і його статус, далі — оплата, адреса доставки і склад замовлення. Після зміни статусу бажано перевірити, що інформація коректно відображається для користувача.

У разі зміни структури каталогу або бізнес-логіки потрібно оновлювати не лише код, а й інструкції для адміністратора. Інструкція має відповідати фактичному інтерфейсу, інакше обслуговування сайту буде залежати від неформальних пояснень розробника.

Обслуговування також включає контроль якості даних. Потрібно періодично

					2026.КВР.122.421.16.00.00 ПЗ	Арк.
						62
Зм.	Арк.	№ докум.	Підпис	Дата		

перевіряти, чи немає порожніх описів, некоректних цін, відсутніх зображень, дубльованих товарів або застарілих статусів. Такі помилки не завжди є технічними, але вони впливають на придатність магазину.

Отже, наповнення і обслуговування сайту повинні виконуватися як регулярний процес. Тільки за умови постійного оновлення товарів, перевірки замовлень, резервного копіювання і контролю якості даних вебзастосунк може залишатися актуальним після захисту та впровадження.

Для обслуговування сайту потрібно вести журнал змін у каталозі. У ньому доцільно фіксувати, коли додано або змінено товар, хто виконав зміну, які поля було оновлено і чи перевірено відображення на сайті. Такий журнал допомагає швидко знайти причину помилки, якщо після оновлення каталогу з'являється некоректна інформація.

Окремо потрібно контролювати якість зображень. Перед публікацією фото перевіряється його розмір, різкість, відповідність товару і коректність відображення на сторінці товару та у каталозі. Якщо одне зображення має інший формат або пропорції, воно може порушити сітку товарних карток.

Під час регулярного обслуговування адміністратор повинен перевіряти роботу зовнішніх сервісів. Якщо недоступний платіжний сервіс, сховище зображень або база даних, користувач не зможе завершити основний сценарій. Тому перевірка інтеграцій має бути частиною стандартного регламенту.

Резервне копіювання потрібно поєднувати з контролем відновлення. Недостатньо створити файл копії; необхідно періодично перевіряти, чи можна з нього відновити базу даних. Це особливо важливо для магазину, де втрата замовлень або користувацьких даних має практичні наслідки.

У разі виявлення помилки на сайті потрібно встановити її рівень критичності. Помилки, що блокують оплату, авторизацію або створення замовлення, усуваються першочергово. Менш критичні візуальні недоліки можуть виправлятися планово, але також мають бути зафіксовані.

Регламент обслуговування має охоплювати контроль каталогу, замовлень,

					2026.КВР.122.421.16.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		63

інтеграцій, резервних копій, залежностей і якості даних. Такий контроль зменшує ризик ситуацій, коли сайт формально працює, але користувач не може коректно знайти товар, оформити замовлення або отримати актуальну інформацію.

3.3 Інструкція з популяризації та підтримки сайту

Популяризація вебзастосунку «Viker» спрямована на залучення користувачів до інтернет-магазину велосипедів і підвищення кількості цільових відвідувань. Для інтернет-магазину важливо не лише розмістити сайт в Інтернеті, а й забезпечити його видимість у пошукових системах, соціальних мережах і тематичних каналах.

Першим напрямом популяризації є пошукова оптимізація. Для сторінок каталогу і товарів потрібно використовувати зрозумілі заголовки, описи, ключові слова, коректні URL-адреси й метадані. Назви товарів мають бути конкретними, а описи — достатніми для індексації пошуковими системами.

Другим напрямом є контентна підтримка. Магазин велосипедів може публікувати матеріали про вибір велосипеда, поради з обслуговування, порівняння моделей, сезонні рекомендації і новини про надходження товарів. Такий контент підвищує довіру до сайту і створює додаткові точки входу з пошуку.

Третім напрямом є робота із соціальними мережами. Для популяризації доцільно публікувати фотографії велосипедів, короткі огляди, акційні пропозиції, відгуки покупців і посилання на нові товари. Соціальні мережі можуть привести користувача безпосередньо на сторінку товару або каталогу.

Четвертим напрямом є рекламні кампанії. Їх можна налаштовувати на аудиторію, яка цікавиться велосипедами, активним відпочинком, спортом або міською мобільністю. Перед запуском реклами потрібно перевірити, що сторінки приземлення швидко відкриваються і мають зрозумілу кнопку цільової дії.

Підтримка сайту після запуску включає аналіз поведінки користувачів. Потрібно відстежувати, які сторінки переглядаються найчастіше, на яких етапах користувачі залишають сайт і які товари викликають найбільший інтерес. Такі дані допомагають покращувати каталог і checkout-сценарій.

					2026.КВР.122.421.16.00.00 ПЗ	Арк.
						64
Зм.	Арк.	№ докум.	Підпис	Дата		

Для підтримки актуальності сайту необхідно регулярно оновлювати асортимент, ціни, інформацію про наявність, описи й зображення. Якщо користувач бачить застарілу інформацію, це може призвести до втрати довіри. Особливо важливо своєчасно оновлювати акції та сезонні пропозиції.

Окрему увагу слід приділяти швидкодії. Повільне завантаження сторінок негативно впливає на конверсію і позиції у пошуку. Для підтримки швидкодії потрібно оптимізувати зображення, контролювати розмір сторінок, перевіряти запити до бази даних і використовувати можливості кешування.

Безпека також є частиною підтримки сайту. Потрібно оновлювати залежності, контролювати секрети, обмежувати доступ до адміністративних сторінок і перевіряти підозрілі помилки авторизації. Для магазину, який працює з оплатою, безпечна обробка даних є критичною умовою експлуатації.

Підтримка якості включає регулярне повторне тестування після оновлень. Після зміни каталогу, checkout, авторизації або адміністративної частини слід перевіряти основні сценарії: відкриття каталогу, вибір товару, додавання до кошика, вхід, оформлення замовлення і перегляд статусу.

Ефективність популяризації оцінюється за відвідуваністю, кількістю переглядів товарів, додаванням до кошика, завершеними замовленнями і повторними відвідуваннями. Ці показники дозволяють зрозуміти, які канали просування є результативними, а які потребують зміни.

Для пошукової оптимізації потрібно контролювати не лише загальні метадані, а й зміст сторінок окремих товарів. Назва велосипеда, короткий опис, характеристики, ціна і наявність повинні відповідати реальному товару. Якщо сторінка товару містить неповні або дубльовані описи, пошукова система гірше визначає її зміст, а користувач отримує менше підстав для переходу до покупки.

Під час просування магазину важливо підтримувати єдину структуру посилань. Рекламні оголошення, публікації в соціальних мережах і матеріали з порадами мають вести не на випадкові сторінки, а на відповідні розділи каталогу або конкретні товари. Наприклад, повідомлення про гірські велосипеди доцільно

					2026.КВР.122.421.16.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		65

спрямовувати на сторінки таких моделей, а не лише на головну сторінку сайту.

Для оцінювання результатів підтримки можна використовувати контрольні періоди. Після оновлення асортименту або запуску рекламної кампанії перевіряється кількість відвідувань, переходів до каталогу, переглядів сторінок товарів, додавань до кошика та завершених замовлень. Якщо один з показників погіршується, потрібно переглянути текст, зображення, ціну, швидкість завантаження або зручність маршруту користувача.

Регулярна підтримка також передбачає перевірку адаптивності сторінок. Оскільки покупці можуть відкривати магазин зі смартфона, планшета або ноутбука, необхідно контролювати відображення головної сторінки, каталогу, сторінки товару, форми входу та checkout-сценарію на різних розмірах екрана. Помилки адаптивності знижують довіру до сайту навіть тоді, коли серверна логіка працює правильно.

У разі додавання нових функцій потрібно оновлювати не лише програмний код, а й матеріали підтримки: короткі інструкції для адміністратора, опис правил оновлення каталогу, порядок перевірки замовлень і перелік типових помилок. Це дозволяє передавати сайт на супровід без залежності від розробника та робить подальшу експлуатацію більш керованою.

Популяризація і підтримка сайту охоплюють SEO, контент, соціальні мережі, рекламу, оновлення каталогу, моніторинг, безпеку і повторне тестування. Після запуску ці роботи підтримують видимість вебзастосунку, актуальність товарів і працездатність основних сценаріїв.

Для інтернет-магазину «Viker» такі роботи мають виконуватися регулярно, а не лише під час першого запуску. Після кожного оновлення каталогу або зміни бізнес-правил необхідно перевіряти шлях користувача від пошуку товару до оформлення замовлення, актуальність описів і відповідність цін. Це дає змогу підтримувати стабільну якість сервісу, своєчасно виявляти помилки та зберігати довіру покупців до вебзастосунку.

					2026.КВР.122.421.16.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		66

4. ЕКОНОМІЧНИЙ РОЗДІЛ

Метою економічної частини даного дипломного проєкту є проведення економічних розрахунків, спрямованих на визначення економічної ефективності розробки інтернет-магазину велосипедів «Viker», прийняття рішення про подальший розвиток і впровадження або ж недоцільність проведення відповідної розробки.

Об'єктом розробки є інтернет-магазин велосипедів «Viker».

Розрахунок вартості розробки виконується в декілька етапів:

- описати технологічний процес розробки із зазначенням трудомісткості кожної операції;
- визначити суму витрат на оплату праці основного і допоміжного персоналу, включаючи відрахування на соціальні заходи;
- обчислити витрати на електроенергію;
- нарахувати суму амортизаційних відрахувань;
- визначити суму накладних витрат;
- скласти кошторис та визначити собівартість робіт;
- розрахувати ціну робіт;
- визначити економічну ефективність та термін окупності.

4.1. Визначення стадій технологічного процесу та загальної тривалості проведення НДР

В цьому підрозділі розглянемо основні етапи технологічного процесу для розробки інтернет-магазину велосипедів «Viker». Для визначення загальної тривалості проведення робіт доцільно дані витрат часу по окремих операціях технологічного процесу звести у таблицю.

					2026.КВР.122.421.16.00.00 ПЗ	Арк.
						67
Зм.	Арк.	№ докум.	Підпис	Дата		

Таблиця 4.1 - Середній час виконання робіт по обслуговуванню та стадії (операції) технологічного процесу

№ п/п	Назва операції (стадії)	Виконавець	Середній час виконання операції, год.
1	Аналіз предметної області	Кер. проєкту Pm	8
		Інженер (11)	8
2	Формування технічного завдання	Кер. проєкту (Pm)	7
		Інженер (11)	4
3	Проектування архітектури та бази даних	Інженер (11)	14
		Інженер (12)	20
4	Розробка головної сторінки та каталогу	Інженер (11)	32
5	Розробка сторінки товару, кошика та оформлення замовлення	Інженер (11)	12
		Інженер (12)	18
6	Реалізація авторизації та профілю користувача	Інженер (11)	6
7	Розробка адміністративної панелі	Інженер (12)	12
8	Інтеграція Stripe та Amazon S3	Інженер (11)	10
9	Тестування	Тестувальник	7
10	Підготовка програмної документації	Кер. проєкту (Pm)	2
Разом			160

Сумарний час виконання операцій технологічного процесу становить 160. години.

4.2. Визначення витрат на оплату праці та відрахувань на соціальні заходи

У даному підрозділі проводиться аналіз і розрахунок витрат, пов'язаних з оплатою праці та відрахуваннями на соціальні заходи, що необхідні для розробки інтернет-магазину велосипедів «Viker».

Розмір заробітної плати залежить від складності та умов виконуваної роботи, професійно-ділових якостей працівника, результатів його праці та діяльності підприємства.

Основна заробітна плата розраховується за формулою:

$$Z_{\text{осн.}} = T_c \cdot K_r \quad (4.1)$$

де: T_c – тарифна ставка, грн. (приймаємо для керівника проєкту (Pm) – 450 грн./год, інженера (I2) – 272 грн./год.), інженера (I1) – 113 грн./год., тестувальник – 100 грн./год.; K_r – кількість відпрацьованих годин.

Отже, основна заробітна плата для:

Керівника проєкту (Pm)	$Z_{\text{осн.}} = 17 \cdot 450 = 7\,650$ грн.
Інженера (I2)	$Z_{\text{осн.}} = 50 \cdot 272 = 13\,600$ грн.
Інженера (I1)	$Z_{\text{осн.}} = 86 \cdot 113 = 9\,718$ грн.
Тестувальник	$Z_{\text{осн.}} = 7 \cdot 100 = 700$ грн.

Сумарна основна заробітна плата становить

$$Z_{\text{осн.}} = 7\,650 + 13\,600 + 9\,718 + 700 = 31\,668 \text{ грн.}$$

Додаткова заробітна плата становить 10 – 15 % від суми основної заробітної плати.

$$Z_{\text{дод.}} = Z_{\text{осн.}} \cdot K_{\text{допл.}} \quad (4.2)$$

де: $K_{\text{допл.}}$ – коефіцієнт додаткових виплат працівникам.

Отже додаткова заробітна плата по категоріях працівників становить:

					2026.КВР.122.421.16.00.00 ПЗ	Арк.
						69
Зм.	Арк.	№ докум.	Підпис	Дата		

Керівника проекту $Z_{\text{дод}2} = 7\,650 * 0,1 = 765$ грн.
 Інженера (12) $Z_{\text{дод}3} = 13\,600 * 0,1 = 1360$ грн.
 Інженера (11) $Z_{\text{дод}4} = 9\,718 * 0,1 = 972$ грн.
 Тестувальник $Z_{\text{дод}4} = 700 * 0,1 = 70$ грн.

Загальна додаткова заробітна плата становить:

$$Z_{\text{дод}} = 765 + 1360 + 972 + 70 = 3\,167 \text{ грн.}$$

Звідси загальні витрати на оплату праці ($B_{\text{о.п.}}$) визначаються за формулою:

$$B_{\text{о.п.}} = Z_{\text{осн.}} + Z_{\text{дод.}} \quad (4.3)$$

$$B_{\text{о.п.}} = 31\,668 + 3\,167 = 34\,835 \text{ грн.}$$

Єдиний соціальний внесок (ЄСВ – 22%) визначається за формулою:

$$B_{\text{ЄСВ}} = B_{\text{оп}} * 0,22 \quad (4.4)$$

$$B_{\text{ЄСВ}} = 34\,835 * 0,22 = 7\,664 \text{ грн.}$$

Проведені розрахунки витрат на оплату праці наведено у таблиці 4.2.

Таблиця 4.2 – Зведені розрахунки витрат на оплату праці

№ п/п	Категорія працівників	Основна заробітна плата, грн.			Додаткова заробітна плата, грн.	ЄСВ, грн.	Всього витрати на оплату праці, грн. $6 = 3+4+5$
		Тарифна ставка, грн.	К-сть годин	Фактично нарах. зарплати, грн.			
		1	2	3	4	5	6
1	Кер. проекту (Pm)	450	17	7 650	765	1 851	10 266
2	Інженера (12)	272	50	13 600	1360	3 291	18 251
3	Інженера (11)	113	86	9 718	972	2 352	13 042
4	Тестувальник	100	7	700	70	170	940
Разом				31 668	3 167	7 664	42 499

Отже, загальні витрати на оплату праці становлять 42 499 грн.

					2026.КВР.122.421.16.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		70

4.3. Розрахунок витрат на електроенергію

Розрахуємо вартість електроенергії. Затрати на електроенергію 1-ці обладнання визначаються за формулою:

$$Z_B = W * T * S \quad (4.5)$$

де: W – необхідна потужність, кВт; T – кількість годин роботи обладнання; S – вартість кіловат-години електроенергії (приймаємо 15, 94 грн).

В нашій системі є 1 ПК. Витрати на електроенергію для цього комп'ютера обчислимо окремо, взявши за основу, що час роботи обладнання обчислюється в залежності від виконуваних робіт (згідно табл. 4.1) і споживані потужності наступні: комп'ютер – 0,82 кВт/год.

$$Z_{ек} = 0,82 * 160 * 15,94 = 2091 \text{ грн.}$$

Витрати на електроенергію становлять 2091 грн.

4.4. Розрахунок суми амортизаційних відрахувань інтернет-магазину велосипедів «Biker»

Характерною особливістю застосування основних фондів у процесі виробництва є їх відновлення. Для відновлення засобів праці у натуральному виразі необхідне їх відшкодування у вартісній формі, яке здійснюється шляхом амортизації.

Амортизація – це процес перенесення вартості основних фондів на вартість новоствореної продукції з метою їх повного відновлення

Комп'ютери та оргтехніка належать до четвертої групи основних фондів.

Амортизація на них нараховується лише в випадку, якщо мінімально допустимі строки їх корисного використання 2 роки. Для визначення амортизаційних відрахувань застосовуємо формулу:

$$A = \frac{B_B * H_A}{100\%} * T, \quad (4.6)$$

де: A – амортизаційні відрахування за звітний період, грн.;

					2026.КВР.122.421.16.00.00 ПЗ	Арк.
						71
Зм.	Арк.	№ докум.	Підпис	Дата		

Б_В – балансова вартість групи основних фондів на початок звітного періоду, грн.;

Н_А – норма амортизації, 0,04 %.

Оскільки для написання програми та її тестування використовується один ПК, вартістю 76 000,00 грн., то сума амортизаційних відрахувань становитиме:

$$A = \frac{76000,00 * 0,04}{150} * 160 = 3\,243 \text{ грн.}$$

4.5. Обчислення накладних витрат

Накладні витрати пов'язані з обслуговуванням виробництва, утриманням апарату управління підприємства (фірми) та створення необхідних умов праці.

Накладні витрати можуть становити 20–60 % від суми основної та додаткової заробітної плати працівників.

$$H_B = B_{\text{о.п.}} * 0,2..0,6 \quad (4.7)$$

де: H_В – накладні витрати.

$$H_B = 34\,835 * 0,4 = 13\,934 \text{ грн.}$$

4.6. Складання кошторису витрат та визначення собівартості інтернет-магазину велосипедів «Viker»

Результати проведених вище розрахунків зведемо у таблиці 4.4.

Таблиця 4.4 - Кошторис витрат інтернет-магазину велосипедів «Viker»

№	Зміст витрат	Сума, грн.	В % до загальної суми
1.	Витрати на оплату праці	42 499	69
2.	Витрати на електроенергію	2 091	3
3.	Амортизаційні відрахування	3 243	5
4.	Накладні витрати	13 934	23
5.	Собівартість	61 767	100

Собівартість (C_B) НДР розрахуємо за формулою:

$$C_B = V_{o.p} + V_{c.z} + 3e + A + H_B \quad (4.8)$$

Отже, собівартість дорівнює $C_B = 61\,767$ грн.

4.7. Розрахунок ціни інтернет-магазину велосипедів «Viker»

Розрахунок ціни науково-дослідної роботи включає в себе урахування різноманітних факторів, таких як рівень рентабельності, собівартість та податкова ставка.

Ціну робіт можна визначити за формулою:

$$Ц = C_B * (1 + P_{рен}) * (1 + ПДВ), \quad (4.9)$$

де: C_B – собівартість; $P_{рен}$ – рівень рентабельності; ПДВ – ставка податку на додану вартість.

$$Ц = 61\,767 * (1 + 0,3) * (1 + 0,2) = 96\,357 \text{ грн.}$$

4.8. Визначення економічної ефективності і терміну окупності капітальних вкладень

Ефективність виробництва – це узагальнене і повне відображення кінцевих результатів використання робочої сили, засобів та предметів праці на підприємстві за певний проміжок часу.

Для визначення ефективності продукту розраховують чисту теперішню вартість (ЧТВ) і термін окупності (Ток).

$$ЧТВ = -C_B + \sum_{i=1}^t \frac{\Gamma_{\Pi}}{(1+i)^i}, \quad (4.10)$$

де: C_B – собівартість розробки; Γ_{Π} – грошовий потік за t -ий рік; t – відповідний рік проекту; i – величина дисконтної ставки (10...15%).

$$ЧТВ = -61\,767 + \frac{35\,879}{(1+0,1)^1} + \frac{35\,879}{(1+0,1)^2} + \frac{35\,879}{(1+0,1)^3} = 27\,459 \text{ грн}$$

Якщо $ЧТВ \geq 0$, то проект може бути рекомендований до впровадження.

					2026.КВР.122.421.16.00.00 ПЗ	Арк.
						73
Зм.	Арк.	№ докум.	Підпис	Дата		

Термін окупності визначається за формулою:

$$T_{\text{ок}} = T_{\text{пв}} + \frac{N_{\text{в}}}{\Gamma_{\text{пр}}} \quad (4.11)$$

де: $T_{\text{пв}}$ – період до повного відшкодування витрат, років; $N_{\text{в}}$ – невідшкодовані витрати на початок року, грн.; $\Gamma_{\text{пр}}$ – грошовий потік на початок року, грн.

$$T_{\text{ок}} = 2 + \frac{1892}{35\,879} = 2,05 \text{ р.}$$

Всі дані внесемо в зведену таблицю 4.5.

Таблиця 4.5 – Техніко-економічні показники інтернет-магазину велосипедів «Biker»

№ п/п	Показник	Значення
1.	Собівартість, грн.	61 767
2.	Плановий прибуток або грошовий потік, грн.	35 879
3.	Ціна, грн.	96 357
4.	Чиста теперішня вартість, грн.	27 459
5.	Термін окупності, рік	2,05

Прибутковість проекту та термін окупності свідчать про його фінансову ефективність та здатність повернути капітальні вкладення протягом 2,05 року. Узагальнену таблицю техніко-економічних показників розробки наведено у графічній частині 2026.КВР.122.421.16.00.00 ТБ.

5 ОХОРОНА ПРАЦІ, ТЕХНІКА БЕЗПЕКИ ТА ЕКОЛОГІЧНІ ВИМОГИ

5.1 Система управління охороною праці

Система управління охороною праці визначає, як на підприємстві організовують безпечну роботу, хто відповідає за контроль умов праці та які дії виконуються для запобігання травмам і професійному перевантаженню. У випадку розробки інтернет-магазину велосипедів «Viker» основна увага приділяється робочому місцю розробника програмного забезпечення, роботі з персональним комп'ютером, периферійним обладнанням та електромережею.

До основних заходів належать проведення інструктажів, перевірка справності комп'ютерної техніки, правильне підключення обладнання до електромережі, контроль освітлення й параметрів мікроклімату, а також встановлення режиму праці з короткими перервами. Такі заходи зменшують зорове напруження, статичне навантаження та ризики, пов'язані з несправністю електрообладнання [18].

Робоче місце має бути організоване так, щоб розробник не працював у вимушеній позі. Монітор розміщують перед користувачем на відстані, зручній для читання тексту; верхня межа екрана має бути на рівні очей або трохи нижче. Клавіатуру та мишу розташовують на одній висоті, щоб зап'ястки не перебували в напруженому положенні. Освітлення має бути рівномірним, без відблисків на екрані. Кабелі живлення й мережевого підключення слід прокладати так, щоб вони не заважали пересуванню та не пошкоджувалися під час роботи.

Під час роботи з комп'ютерною технікою не допускається використання пошкоджених кабелів, розеток, блоків живлення або подовжувачів. Якщо обладнання перегрівається, іскрить або має запах горілої ізоляції, роботу припиняють, а пристрій від'єднують від мережі. Для пожежної безпеки важливо не перевантажувати розетки, не накривати вентиляційні отвори системного блока чи ноутбука та залишати вільний доступ до засобів пожежогасіння.

					2026.КВР.122.421.16.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		75

Режим праці має враховувати тривалу роботу з екраном. Після періодів програмування або тестування доцільно робити короткі перерви, змінювати положення тіла та переводити погляд з екрана на віддалені об'єкти. Яскравість монітора налаштовують відповідно до освітлення приміщення. Екологічні вимоги для такого робочого місця зводяться до ощадного використання електроенергії та передачі несправних електронних пристроїв, батарей і комплектуючих до спеціалізованих пунктів збору.

5.2 Мікроклімат виробничих приміщень та його параметри

Мікроклімат виробничого приміщення характеризується температурою повітря, відносною вологістю, швидкістю руху повітря та тепловим випромінюванням. Для розробника ці параметри мають практичне значення: надто висока температура спричиняє втому й сонливість, надто низька створює дискомфорт, а сухе повітря посилює втому очей під час роботи з монітором.

Норми мікроклімату встановлюють з урахуванням категорії робіт, пори року та тепловиділення в приміщенні [19]. Для робочого місця розробника програмного забезпечення прийнятними є параметри, наведені в таблиці 5.1.

Таблиця 5.1 – Параметри мікроклімату робочого місця розробника

Період року	Температура повітря, °С	Відносна вологість, %	Швидкість руху повітря, м/с
Холодний період	21–24	40–60	до 0,1
Теплий період	22–25	40–60	до 0,2
Рекомендовані умови для робочого місця розробника	22–24	40–60	до 0,2

ВИСНОВКИ

У кваліфікаційній роботі розроблено інтернет-магазин велосипедів «Viker», призначений для автоматизації процесу онлайн-продажу велосипедів. У межах роботи виконано аналіз предметної області, сформульовано технічне завдання, визначено функціональні та нефункціональні вимоги, спроектовано структуру вебзастосунку, описано базу даних, реалізовані модулі, інтерфейс користувача, адміністративну панель, порядок інсталяції, тестування та експлуатації.

У першому розділі проаналізовано існуючі підходи до створення інтернет-магазинів і визначено доцільність розробки окремого вебзастосунку. Сформульовано призначення програмного продукту, перелік користувачів, основні функції, вимоги до даних, програмної документації, етапів розробки та контролю якості.

У другому розділі обґрунтовано вибір технологій. Для реалізації застосунку використано Next.js, React і TypeScript, що забезпечують компонентну структуру, маршрутизацію та типізацію. Для роботи з даними застосовано PostgreSQL і Prisma ORM. Авторизацію реалізовано через NextAuth, оплату — через Stripe, а зберігання зображень товарів передбачено з використанням Amazon S3. Описано архітектуру системи, структуру сторінок, базу даних, серверні дії, каталог товарів, кошик, оформлення замовлення, відгуки та адміністративну панель.

У третьому розділі наведено інструкцію з інсталяції, підготовки середовища, запуску застосунку, використання тестових наборів та експлуатації програмного комплексу. Описані сценарії перевірки дозволяють контролювати працездатність головної сторінки, каталогу, фільтрів, сторінки товару, кошика, авторизації, оформлення замовлення та адміністративних функцій.

В економічному розділі підготовлено структуру розрахунків для визначення витрат на розробку та економічної ефективності впровадження. Через відсутність погоджених вихідних числових даних розділ містить формули, таблиці та маркери для подальшого заповнення після консультації з відповідальним викладачем.

У розділі охорони праці розглянуто вимоги до безпечної організації робочого

					2026.КВР.122.421.16.00.00 ПЗ	Арк.
						77
Зм.	Арк.	№ докум.	Підпис	Дата		

місця розробника, електробезпеки, режиму праці, пожежної безпеки та екологічної відповідальності під час використання комп'ютерної техніки.

Результатом роботи є вебзастосунок, який підтримує основні можливості інтернет-магазину: перегляд головної сторінки, каталог товарів, фільтрацію, сортування, сторінку детального перегляду товару, вибір кольору та кількості, додавання до кошика, авторизацію користувачів, оформлення замовлень, оплату, відгуки та адміністративне керування товарами, замовленнями і статусами.

Поставлену мету досягнуто: створено програмний продукт, що відповідає предметній області інтернет-магазину велосипедів і може бути використаний як основа для подальшого розвитку системи електронної комерції. У подальшому застосунок можна розширити за рахунок повнотекстового пошуку, розширених фільтрів, обраних товарів, історії замовлень, інтеграції зі службами доставки, email-сповіщень і покращеної аналітики продажів.

					2026.КВР.122.421.16.00.00 ПЗ	Арк.
						78
Зм.	Арк.	№ докум.	Підпис	Дата		

ПЕРЕЛІК ПОСИЛАНЬ

1. Riva M. Real-World Next.js : Build scalable, high-performance, and modern web applications using Next.js, the React framework for production. Birmingham : Packt Publishing, 2022. 366 p.
2. Banks A., Porcello E. Learning React : Modern Patterns for Developing React Apps. 2nd ed. Sebastopol : O'Reilly Media, 2020. 310 p.
3. Cherny B. Programming TypeScript : Making Your JavaScript Applications Scale. Sebastopol : O'Reilly Media, 2019. 322 p.
4. Prisma Documentation. Prisma. URL: <https://www.prisma.io/docs> (дата звернення: 21.05.2026).
5. PostgreSQL Documentation. PostgreSQL Global Development Group. URL: <https://www.postgresql.org/docs/> (дата звернення: 21.05.2026).
6. NextAuth.js Documentation. Auth.js. URL: <https://next-auth.js.org/> (дата звернення: 21.05.2026).
7. Stripe Documentation. Stripe. URL: <https://docs.stripe.com/> (дата звернення: 21.05.2026).
8. Amazon S3 Documentation. Amazon Web Services. URL: <https://docs.aws.amazon.com/s3/> (дата звернення: 21.05.2026).
9. Tailwind CSS Documentation. Tailwind CSS. URL: <https://tailwindcss.com/docs> (дата звернення: 21.05.2026).
10. Radix UI Documentation. Radix UI. URL: <https://www.radix-ui.com/primitives/docs/overview/introduction> (дата звернення: 21.05.2026).
11. Zustand Documentation. Zustand. URL: <https://zustand.docs.pmnd.rs/> (дата звернення: 21.05.2026).
12. MDN Web Docs. HTML: HyperText Markup Language. Mozilla. URL: <https://developer.mozilla.org/en-US/docs/Web/HTML> (дата звернення: 21.05.2026).
13. MDN Web Docs. CSS: Cascading Style Sheets. Mozilla. URL: <https://developer.mozilla.org/en-US/docs/Web/CSS> (дата звернення: 21.05.2026).
14. MDN Web Docs. JavaScript. Mozilla. URL: <https://developer.mozilla.org/en->

					2026.КВР.122.421.16.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		79

US/docs/Web/JavaScript (дата звернення: 21.05.2026).

15. Web Content Accessibility Guidelines (WCAG) 2.2. W3C. URL: <https://www.w3.org/TR/WCAG22/> (дата звернення: 21.05.2026).

16. ДСТУ 8302:2015. Інформація та документація. Бібліографічне посилання. Загальні положення та правила складання. Київ : ДП «УкрНДНЦ», 2016. 20 с.

17. ДСТУ 3008:2015. Інформація та документація. Звіти у сфері науки і техніки. Структура та правила оформлювання. Київ : ДП «УкрНДНЦ», 2016.

18. Про охорону праці : Закон України від 14.10.1992 № 2694-ХІІ. Верховна Рада України. URL: <https://zakon.rada.gov.ua/laws/show/2694-12> (дата звернення: 22.05.2026).

19. ДСН 3.3.6.042-99. Санітарні норми мікроклімату виробничих приміщень : постанова Головного державного санітарного лікаря України від 01.12.1999 № 42. URL: <https://zakon.rada.gov.ua/rada/show/va042282-99> (дата звернення: 22.05.2026).

20. Методичні вказівки до виконання кваліфікаційної роботи для здобувачів фахової передвищої освіти спеціальності 122 «Комп'ютерні науки» / уклад.: Г. Я. Марціяш, Р. О. Слободян. Тернопіль : ВСП «ТФК ТНТУім. І. Пулюя», 2026. 48 с.

					2026.КВР.122.421.16.00.00 ПЗ	Арк.
						80
Зм.	Арк.	№ докум.	Підпис	Дата		

ДОДАТКИ

Додаток А. Лістинг програмного коду оголошення моделей Prisma

```
generator client {
  provider = "prisma-client-js"
  output   = "../generated/prisma"
}

datasource db {
  provider = "postgresql"
  url      = env("DATABASE_URL")
}

model User {
  id          String    @id @default(cuid())
  name        String?
  email       String    @unique
  emailVerified DateTime?
  role        String    @default("USER")
  image       String?
  accounts    Account[]
  sessions    Session[]
  reviews    Review[]
  password    String?
  phone       String?
  addressLine1 String?
  addressLine2 String?
  city        String?
  state       String?
  zip         String?
  country     String?
  orders      Order[]
}
```

					2026.КВР.122.421.16.00.00 ПЗ	Арк.
						81
Зм.	Арк.	№ докум.	Підпис	Дата		

```

    createdAt      DateTime @default(now())
    updatedAt      DateTime @updatedAt
}

model Account {
  userId          String
  type            String
  provider        String
  providerAccountId String
  refresh_token   String?
  access_token    String?
  expires_at     Int?
  token_type      String?
  scope           String?
  id_token        String?
  session_state   String?

  createdAt DateTime @default(now())
  updatedAt  DateTime @updatedAt

  user User @relation(fields: [userId], references: [id], onDelete: Cascade)

  @@id([provider, providerAccountId])
}

model Session {
  sessionToken String @unique
  userId       String
  expires      DateTime
  user         User    @relation(fields: [userId], references: [id],
onDelete: Cascade)

  createdAt DateTime @default(now())
  updatedAt  DateTime @updatedAt

```

					2026.KBP.122.421.16.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		82

```
}
```

```
model VerificationToken {  
  identifier String  
  token      String  
  expires    DateTime  
  
  @@id([identifier, token])  
}
```

```
model Bicycle {  
  id          String  @id @default(uuid())  
  
  title       String  
  description String?  
  price       Float  
  images      String[]  
  
  features    String[]  
  colors      String[]  
  
  reviews     Review[]  
  orderItems  OrderItem[]  
  createdAt   DateTime @default(now())  
  updatedAt   DateTime @updatedAt  
}
```

```
model Order {  
  id          String  @id @default(uuid())  
  
  userId      String  
  statusId    String  
  stripeIntentId String
```

					2026.KBP.122.421.16.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		83

```

address      String
shippingMethod String @default("Unknown")
shippingCost Float @default(0)
total        Float @default(0)

createdAt    DateTime @default(now())

user         User      @relation(fields: [userId], references: [id])
status       Status    @relation(fields: [statusId], references: [id])
orderItems   OrderItem[]
}

model OrderItem {
  id          String    @id @default(uuid())
  orderId     String
  bicycleId   String
  quantity    Int
  color       String    @default("")

  createdAt   DateTime @default(now())
  order       Order     @relation(fields: [orderId], references: [id])
  bicycle     Bicycle   @relation(fields: [bicycleId], references: [id],
onDelete: Cascade)
}

model Review {
  id          String    @id @default(uuid())
  title       String
  comment     String
  rating      Int
  helpful     Int
  userId      String
  user        User      @relation(fields: [userId], references: [id])
  bicycleId   String

```

					2026.KBP.122.421.16.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		84

```

    bicycle      Bicycle  @relation(fields: [bicycleId], references: [id],
onDelete: Cascade)
    createdAt    DateTime @default(now())
}

model Status {
    id           String   @id @default(uuid())
    title        String   @unique
    orders       Order[]
}

```

					2026.KBP.122.421.16.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		85

Додаток Б. Лістинг програмного коду серверних дій для створення товарів

```
export async function getBicycle(id: string):
Promise<BicycleWithUserReviews> {
  const bicycle = await prisma.bicycle.findUniqueOrThrow({
    where: { id },
    include: {
      reviews: {
        include: {
          user: true,
        },
      },
    },
  });
  const rating =
    bicycle.reviews.reduce((acc, review) => acc + review.rating, 0) /
    bicycle.reviews.length;

  return {
    ...bicycle,
    rating: Number.isNaN(rating) ? '0.00' : rating.toFixed(2),
  };
}

export async function getAllBicycles(params: Params):
Promise<Array<BicycleWithReviews>> {
  const bicycles = await prisma.bicycle.findMany({
    ...params,
    include: {
      ...params.include,
      reviews: true,
    },
  });
}
```

					2026.KBP.122.421.16.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		86

```

return bicycles.map((bicycle) => {
  const rating =
    bicycle.reviews.reduce((acc, review) => acc + review.rating, 0) /
    bicycle.reviews.length;

  return {
    ...bicycle,
    rating: Number.isNaN(rating) ? '0.00' : rating.toFixed(2),
  };
});
}

export async function getBicyclesCount(params: Prisma.BicycleCountArgs) {
  const count = await prisma.bicycle.count(params);
  return count;
}

export async function getAllBicyclesColors() {
  const bicycles = await prisma.bicycle.findMany({
    select: {
      colors: true,
    },
  });

  const allColors = bicycles.flatMap((bicycle) => bicycle.colors);
  return [...new Set(allColors)];
}

export async function getBicycleCategoryCounts(): Promise<{
  total: number;
  categories: BicycleCategoryWithCount[];
  idsByCategory: Record<string, string[]>;
}> {

```

					2026.KBP.122.421.16.00.00 ПЗ	Арк.
						87
Зм.	Арк.	№ докум.	Підпис	Дата		

```

const bicycles = await prisma.bicycle.findMany({
  select: {
    id: true,
    title: true,
    description: true,
    features: true,
  },
});
const idsByCategory = Object.fromEntries(
  BICYCLE_CATEGORIES.map((category) => [category.id, [] as string[]])
);

bicycles.forEach((bicycle) => {
  const categoryId = getBicycleCategoryId(bicycle);

  if (categoryId) {
    idsByCategory[categoryId].push(bicycle.id);
  }
});
const categories = BICYCLE_CATEGORIES.map((category) => ({
  ...category,
  count: idsByCategory[category.id].length,
}));

return { total: bicycles.length, categories, idsByCategory };
}

export async function createBicycle(data: CreateBicycleData) {
  await requireAdmin();

  const images = await Promise.all(
    data.images.map(async (image) => {
      const { url } = await uploadImageToS3(image);
      return url;
    })
  );
}

```

					2026.KBP.122.421.16.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		88

```

    })
  );
  const bicycle = await prisma.bicycle.create({ data: { ...data, images }
});
  revalidatePath('/dashboard');
  return bicycle;
}

export async function deleteBicycle(id: string) {
  await requireAdmin();

  await prisma.bicycle.delete({
    where: { id }
  });
  revalidatePath('/dashboard');
}

export async function updateBicycle(id: string, data: UpdateBicycleData) {
  await requireAdmin();

  const existingImages = data.images.filter(
    (image) => typeof image === 'string'
  );
  const uploadedImages = await Promise.all(
    data.images
      .filter((image) => typeof image !== 'string')
      .map(async (image) => {
        const { url } = await uploadImageToS3(image);
        return url;
      })
  );
  const bicycle = await prisma.bicycle.update({
    where: { id },

```

					2026.KBP.122.421.16.00.00 ПЗ	Арк.
						89
Зм.	Арк.	№ докум.	Підпис	Дата		

```
data: {
  title: data.title,
  description: data.description,
  price: data.price,
  images: [...existingImages, ...uploadedImages],
  features: data.features,
  colors: data.colors,
},
});
revalidatePath('/dashboard');
return bicycle;
}
```

					2026.KBP.122.421.16.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		90

Додаток В. Лістинг програмного коду реалізації клієнтського стану

КОШИКА

```
import { create } from 'zustand';

interface CartItem {
  id: string;
  color: string;
  quantity: number;
}

interface CartState {
  isOpen: boolean;
  toggleCart: () => void;
  items: CartItem[];
  addToCart: (item: CartItem) => void;
  removeFromCart: (id: string, color: string) => void;
  removeOneItem: (id: string, color: string) => void;
  setIsOpen: (isOpen: boolean) => void;
  clearCart: () => void;
}

const useCartStore = create<CartState>((set) => ({
  isOpen: false,
  items: [],
  toggleCart: () => set((state) => ({ isOpen: !state.isOpen })),
  setIsOpen: (isOpen: boolean) => set({ isOpen }),
  addToCart: (item) => set((state) => {
    const existingItem = state.items.find((i) => i.id === item.id && i.color
=== item.color);
    if (existingItem) {
      return {
        items: state.items.map((i) =>
          i.id === item.id && i.color === item.color
```

					2026.KBP.122.421.16.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		91

```

        ? { ...i, quantity: i.quantity + item.quantity }
        : i
    ),
};
}
return { items: [...state.items, item] };
}),
removeFromCart: (id, color) =>
    set((state) => ({
        items: state.items.filter((item) => item.id !== id || item.color !==
color),
    })),
removeOneItem: (id, color) =>
    set((state) => {
        const item = state.items.find((i) => i.id === id && i.color === color);
        if (!item) return state;

        if (item.quantity > 1) {
            return {
                items: state.items.map((i) =>
                    i.id === id && i.color === color ? { ...i, quantity: i.quantity
- 1 } : i
                ),
            };
        }
        return { items: state.items.filter((i) => i.id !== id || i.color !==
color) };
    }),
    clearCart: () => set({ items: [] }),
}));

export default useCartStore;

```

					2026.KBP.122.421.16.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		92

Додаток Г. Лістинг функції створення замовлення

```
export const createOrder = async (order: CreateOrderParams) => {
  const session = await requireUser();
  const existingOrder = await prisma.order.findFirst({
    where: {
      stripeIntentId: order.stripeIntentId,
      userId: session.user.id,
    },
  });

  if (existingOrder) {
    return existingOrder;
  }

  const stripe = new Stripe(process.env.STRIPE_SECRET_KEY!, {
    apiVersion: '2025-03-31.basil',
  });
  const paymentIntent = await stripe.paymentIntents.retrieve(
    order.stripeIntentId
  );

  if (paymentIntent.status !== 'succeeded') {
    throw new Error('Payment has not succeeded');
  }

  if (paymentIntent.metadata.userId !== session.user.id) {
    throw new Error('Payment does not belong to the current user');
  }

  const cartItems = JSON.parse(
    paymentIntent.metadata.cartItems || '[]'
  ) as Array<CheckoutCartItemInput & { bicycleId?: string }>;
  const checkout = await calculateCheckout(
```

					2026.KBP.122.421.16.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		93

```

    cartItems.map((item) => ({
      id: item.id ?? item.bicycleId ?? '',
      color: item.color,
      quantity: item.quantity,
    })),
    paymentIntent.metadata.shippingOption
  );

  if (paymentIntent.amount !== checkout.amount) {
    throw new Error('Payment amount does not match the cart');
  }

  const shippingAddress = paymentIntent.shipping?.address;
  const address = [
    shippingAddress?.line1,
    shippingAddress?.line2,
    shippingAddress?.city,
    shippingAddress?.state,
    shippingAddress?.postal_code,
    shippingAddress?.country,
  ]
    .filter(Boolean)
    .join(', ');

  const createdOrder = await prisma.order.create({
    data: {
      stripeIntentId: order.stripeIntentId,
      address,
      shippingMethod: checkout.shippingOption.name,
      shippingCost: checkout.shippingCost,
      total: checkout.total,
      user: {
        connect: {
          id: session.user.id,

```

					2026.KBP.122.421.16.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		94

```

    },
  },
  status: {
    connectOrCreate: {
      where: { title: 'On hold' },
      create: { title: 'On hold' },
    },
  },
  orderItems: {
    create: checkout.cartItems.map((item) => ({
      quantity: item.quantity,
      color: item.color,
      bicycle: {
        connect: {
          id: item.bicycleId,
        },
      },
    })),
  },
});

```

```

return createdOrder;
};

```

```

export const getOrders = async (offset: number) => {
  await requireAdmin();

```

```

  const orders = await prisma.order.findMany({
    take: 5,
    skip: Math.max(0, offset - 5),
    orderBy: {
      createdAt: 'desc',
    },
  },

```

					2026.KBP.122.421.16.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		95

```

include: {
  user: {
    select: {
      name: true,
      email: true,
    },
  },
  status: {
    select: {
      title: true,
    },
  },
  orderItems: {
    include: {
      bicycle: {
        select: {
          title: true,
          price: true,
          images: true,
        },
      },
    },
  },
},
});

const total = await prisma.order.count();

return { orders, total };
};

export const updateOrderStatus = async (orderId: string, statusTitle: string)
=> {
  await requireAdmin();

```

					2026.KBP.122.421.16.00.00 ПЗ	Арк.
						96
Зм.	Арк.	№ докум.	Підпис	Дата		

```
await prisma.order.update({
  where: { id: orderId },
  data: {
    status: {
      connectOrCreate: {
        where: { title: statusTitle },
        create: { title: statusTitle },
      },
    },
  },
});
```

					2026.KBP.122.421.16.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		97

Додаток Д. Лістинг програмного коду авторизації

```
export const config = {
  callbacks: {
    async jwt({ token, user }) {
      if (user) {
        token.id = user.id;
        token.role = user.role ?? 'USER';
      }

      return token;
    },
    async session({ session, token }) {
      if (session.user && token.id) {
        session.user.id = String(token.id);
        session.user.role = typeof token.role === 'string' ? token.role :
'USER';
      }

      return session;
    },
  },
  session: {
    strategy: 'jwt',
  },
  secret: process.env.NEXTAUTH_SECRET ?? 'development-only-next-auth-
secret',
  providers: [
    GoogleProvider({
      clientId: process.env.GOOGLE_CLIENT_ID!,
      clientSecret: process.env.GOOGLE_CLIENT_SECRET!,
    }),
    CredentialsProvider({
      credentials: {
```

					2026.KBP.122.421.16.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		98

```

    email: { label: 'Email', type: 'email' },
    password: { label: 'Password', type: 'password' },
  },
  async authorize(credentials) {
    if (!credentials?.email || !credentials?.password) {
      return null;
    }

    try {
      const user = await prisma.user.findUnique({
        where: { email: credentials.email },
      });

      if (!user) {
        return null;
      }

      if (!user.password) {
        return null;
      }

      const isValid = await bcrypt.compare(
        credentials.password,
        user.password
      );

      if (!isValid) {
        return null;
      }

      return user;
    } catch (error) {
      console.error('Error during authentication:', error);
      return null;
    }
  }
}

```

					2026.KBP.122.421.16.00.00 ПЗ	Арк.
						99
Зм.	Арк.	№ докум.	Підпис	Дата		

```

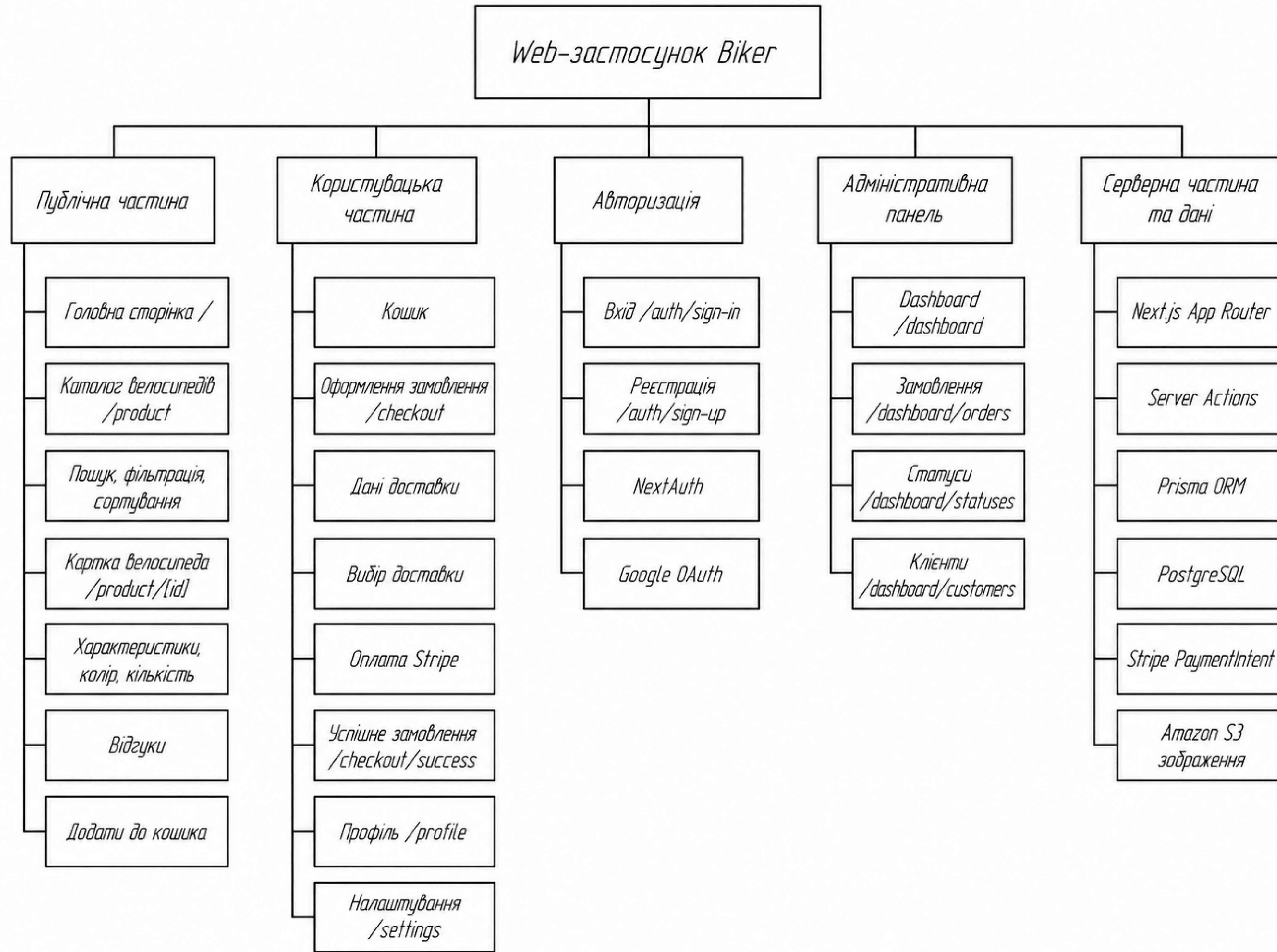
    }
  },
  }),
],
adapter: PrismaAdapter(prisma),
} satisfies NextAuthOptions;

export function auth(
  ...args:
  | [GetServerSidePropsContext['req'], GetServerSidePropsContext['res']]
  | [NextApiRequest, NextApiResponse]
  | []
) {
  return getServerSession(...args, config);
}

```

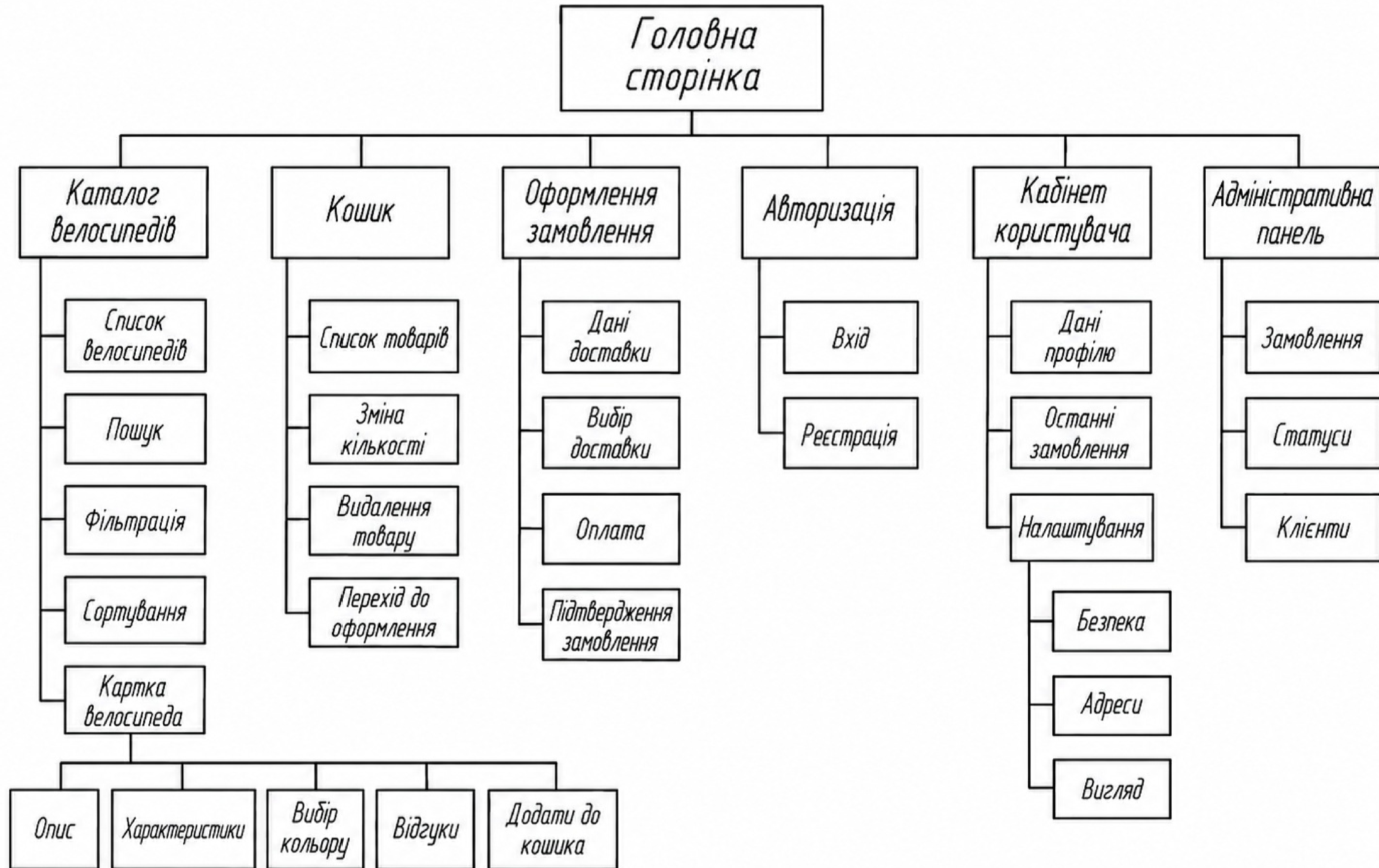
					2026.KBP.122.421.16.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		100

Схема структурна web-застосунку



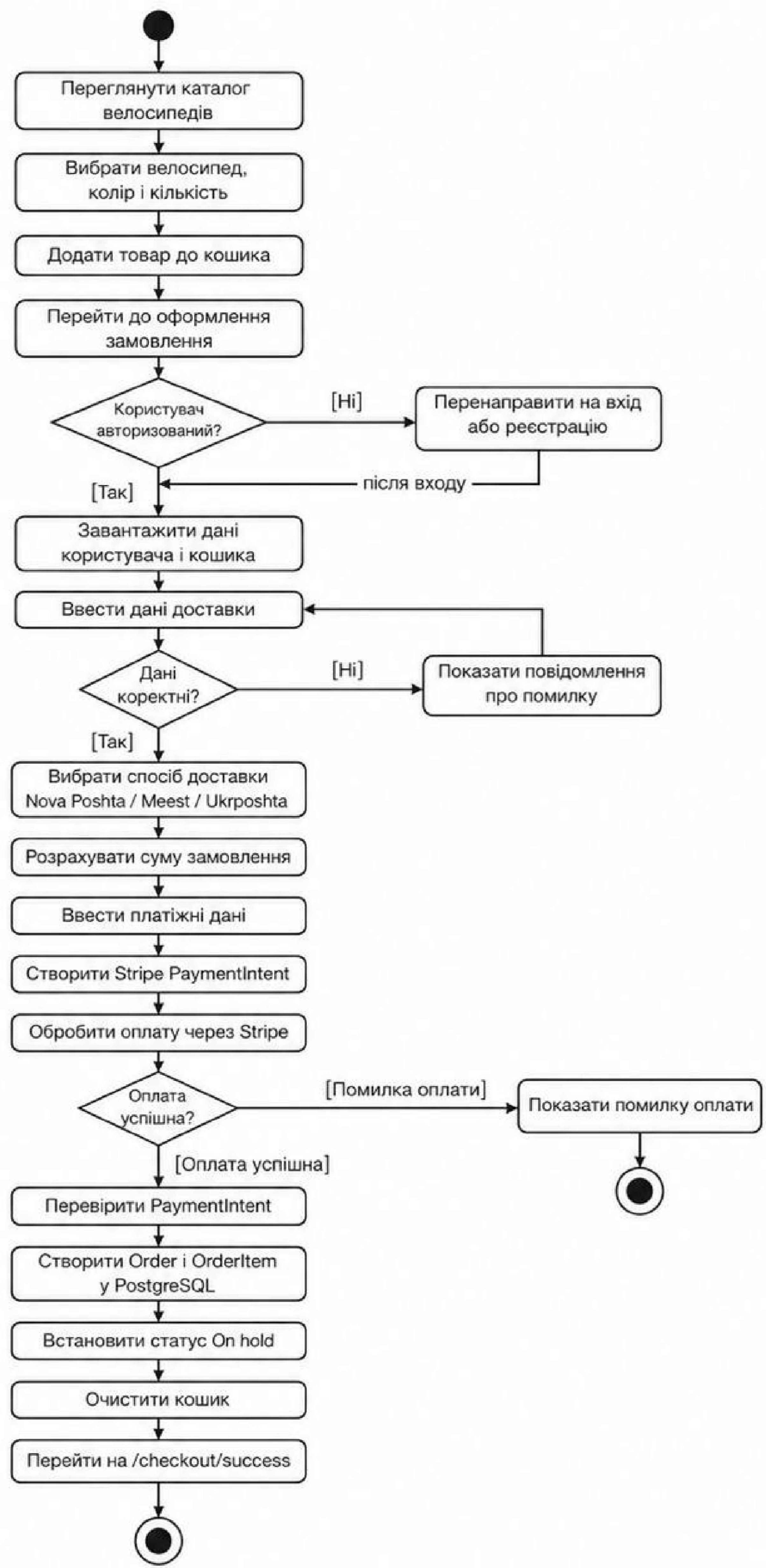
					2026. КВР.122.421.16.00.02 СС		
					Розробка інтернет-магазину велосипедів «Бікер»		
Зм.	Арк.	№ документа	Підпис	Дата	Літ.	Маса	Масшт.
Розроб.		Хромов В.В.					
Перевір.		Марціш Г.Я.					
Т. контр.					Аркуш	Аркушів 1	
Н. контр.		Гришах В. А.			ВСП ТФК ТНТУ КН. 421 м. Тернопіль		
Затв.							

Схема структурна карти сторінок web-застосунку



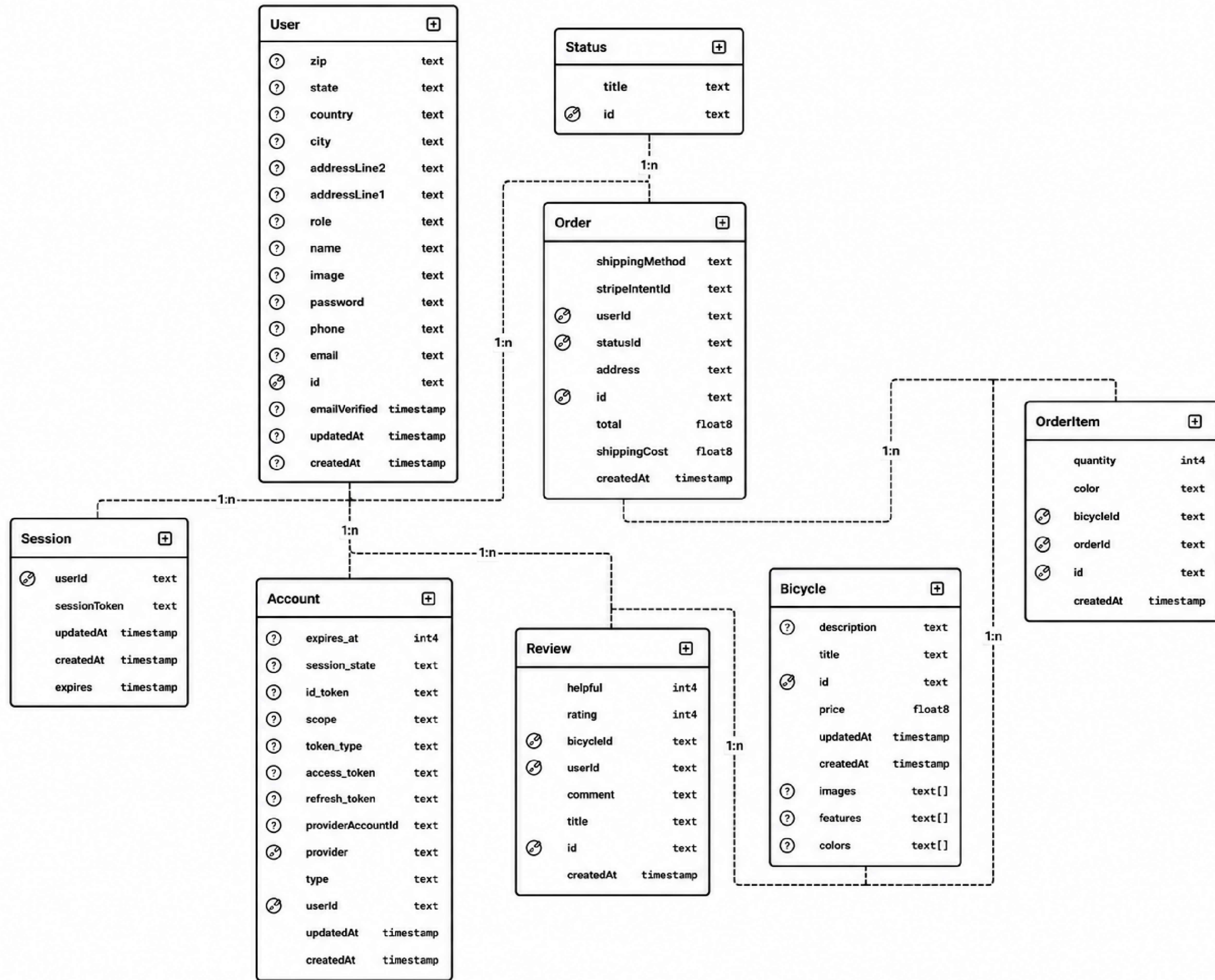
					2026 КВР.122.421.16.00.01 СС		
Зм.	Арк.	№ документа	Підпис	Дата	Разробка інтернет-магазину велосипедів «Вікес»		
Разроб.	Хромов В.В.				Літ.	Маса	Масшт.
Перевір.	Марціш Г.Я.				Схема структурна карти сторінок web-застосунку		
Т. контр.					Аркуш	Аркушів 1	
Н. контр.	Гришах В.А.				ВСП ТФК ТНТУ КН. 421 м. Тернопіль		
Затв.							

Блох-схема оформлення замовлення



				2026_KBP.122.421.16.00_00_БС				
Зм.	Арк.	№ документа	Підпис	Дата	Розробка інтернет-магазину велосипедів «Вікел»	Літ.	Маса	Масшт.
Розроб.		Хронов В.Ю.			Блох-схема оформлення замовлення			
Перевір.		Марцінів Г.Я.				Аркуш		Аркушів 1
Т контр.						ВСП ТФК ТНТУ КН-421 м. Тернопіль		
Н. контр.		Вруйман В. А.						
Затв.								

ER-діаграма бази даних



2026.KBP.122.421.16.00.00 БД				
Зм.	Арк.	№ документа	Підпис	Дата
Розроб.	Хромов В.В.			
Перевір.	Марціш Г.Я.			
Т. контр.				
Н. контр.	Гришах В. А.			
Зав.				
Розробка інтернет-магазину велосипедів «Bikes»			Літ.	Маса
ER-діаграма бази даних			Аркуш	Аркушів 1
ВСП ТФК ТНТУ КН. 421 м. Тернопіль				

Таблиця техніко-економічних показників

№	Показник	Одиниці вимірювання	Значення
1	Середовище програмування	–	Cursor
2	Інтерфейс	–	Інтернет-магазин
3	Мова програмування	–	TypeScript, JavaScript
4	Фреймворк	–	Next.js, React
5	База даних	–	PostgreSQL
6	Собівартість	грн	61 767
7	Плановий прибуток	грн	35 879
8	Ціна	грн	96 357
9	Чиста теперішня вартість	грн	27 459
10	Термін окупності	рік	2,05

2026.КВР.122.421.16.00.00 ТБ				
Зм.	Арк.	№ документа	Підпис	Дата
Разроб.		Хромов В.В.		
Перевір.		Марціш Г.Я.		
Т. контр.				
Н. контр.		Лриіах В. А.		
Затв.				

2026.КВР.122.421.16.00.00 ТБ		
Літ.	Маса	Масшт.
Аркуш	Аркушів 1	
ВСП ГОК ТНТУ КН. 421 м. Тернопіль		