

Міністерство освіти і науки України
Національний технічний університет
«Дніпровська політехніка»

Інститут електроенергетики

(інститут)

Факультет інформаційних технологій

(факультет)

Кафедра Програмного забезпечення комп'ютерних систем

(повна назва)

ПОЯСНЮВАЛЬНА ЗАПИСКА
кваліфікаційної роботи ступеня

бакалавра

(назва освітньо-кваліфікаційного рівня)

студентки

Кондрат Анни Дмитрівни

(ПІБ)

академічної групи

122-20-2

(шифр)

спеціальності

122 Комп'ютерні науки

(код і назва спеціальності)

освітньої програми

Комп'ютерні науки

(назва освітньої програми)

на тему:

Розробка веборієнтованого додатку

з продажу вінтажних велосипедів за допомогою JavaScript та PHP

Керівники	Прізвище, ініціали	Оцінка за шкалою		Підпис
		рейтинговою	інституційною	
кваліфікаційної роботи	<i>доц. Ширін А.Л.</i>			
розділів:				
спеціальний	<i>доц. Ширін А.Л.</i>			
економічний	<i>доц. Касьяненко Л.В.</i>			
Рецензент				
Нормоконтролер	<i>доц. Гуліна І.Г.</i>			

Дніпро
2024

РЕФЕРАТ

Пояснювальна записка: 111 с., 30 рис., 8 табл., 3 дод., 31 джерело.

Об'єкт розробки: веборієнтований додаток для продажу вінтажних велосипедів з використанням JavaScript та PHP.

Мета кваліфікаційної роботи: створення спеціалізованої онлайн-платформи для продажу вінтажних велосипедів, яка враховує специфіку цього ринку та задовольняє потреби як продавців, так і покупців.

У вступі розглядається актуальність розробки веборієнтованого додатку для продажу вінтажних велосипедів, конкретизується мета роботи та визначаються основні завдання. Наводиться обґрунтування вибору технологій JavaScript та PHP для реалізації проекту.

У першому розділі проаналізовано предметну галузь електронної комерції та веброзробки, визначено призначення розробки, сформульовано постановку завдання. Розглянуто інноваційні аспекти електронної комерції, характеристики галузі застосування та основні технічні виклики при розробці вебдодатків з використанням JavaScript та PHP. Проведено огляд аналогічних вебдодатків та визначено вимоги до програмної реалізації.

У другому розділі описано функціональне призначення системи, застосовані математичні методи, використані технології та мови програмування. Розглянуто структуру системи та алгоритми її функціонування, описано файлову структуру проекту та структуру бази даних. Наведено обґрунтування та організацію вхідних та вихідних даних програми, описано інтерфейс користувача та технічні засоби, використані для розробки.

В економічному розділі визначено трудомісткість розробки веборієнтованого додатку для продажу вінтажних велосипедів, проведено розрахунок витрат на створення програми та визначено очікуваний період розробки. Загальна трудомісткість становить 1371,72 людино-годин, очікуваний період створення програмного забезпечення - близько 8 місяців, а витрати на створення - 140 570,65 грн.

Практичне значення полягає у створенні функціонального вебдодатку для ефективної діяльності інтернет-магазину вінтажних велосипедів. Додаток забезпечує зручний інструмент для продажу та придбання унікальних товарів, розширюючи ринок збуту для компаній та надаючи покупцям доступ до каталогу з будь-якої точки світу.

Актуальність розробки визначається необхідністю створення спеціалізованої платформи, що враховує особливості ринку вінтажних велосипедів. Додаток забезпечує ефективну презентацію товарів, безпечні онлайн-транзакції та сприяє популяризації велосипедного транспорту.

Список ключових слів: ВЕБОРІЄНТОВАНИЙ ДОДАТОК, ВІНТАЖНІ ВЕЛОСИПЕДИ, ЕЛЕКТРОННА КОМЕРЦІЯ, JAVASCRIPT, PHP, MYSQL, BOOTSTRAP, АДАПТИВНИЙ ДИЗАЙН, КОРИСТУВАЦЬКИЙ ІНТЕРФЕЙС, БАЗА ДАНИХ, КОШИК ПОКУПОК, КАТАЛОГ ТОВАРІВ, АВТОРИЗАЦІЯ, БЕЗПЕЧНІ ТРАНЗАКЦІЇ.

ABSTRACT

Explanatory note: 111 p., 30 figures, 8 tables, 3 appendixes, 31 sources.

Development object: a web-based application for selling vintage bicycles using JavaScript and PHP.

The purpose of the qualification work: to create a specialised online platform for the sale of vintage bicycles that takes into account the specifics of this market and meets the needs of both sellers and buyers.

The introduction discusses the relevance of developing a web-based application for the sale of vintage bicycles, specifies the purpose of the work and defines the main tasks. The rationale for choosing JavaScript and PHP technologies for the project implementation is given.

The first section analyses the subject area of e-commerce and web development, defines the purpose of the development, and formulates the task. The innovative aspects of e-commerce, the characteristics of the field of application, and the main technical challenges in developing web applications using JavaScript and PHP are considered. A review of similar web applications is carried out and the requirements for software implementation are determined.

The second section describes the functional purpose of the system, the applied mathematical methods, technologies and programming languages. The structure of the system and its functioning algorithms are considered, the file structure of the project and the structure of the database are described. The rationale and organisation of the input and output data of the programme are presented, the user interface and technical means used for development are described.

The economic section defines the labour intensity of developing a web-based application for the sale of vintage bicycles, calculates the costs of creating the application and determines the expected development period. The total labour intensity is 1371.72 man-hours, the expected period of software development is about 8 months, and the cost of development is 140,570.65 UAH.

The practical significance is to create a functional web application for the effective operation of an online vintage bicycle store. The application provides a convenient tool for selling and purchasing unique goods, expanding the market for companies and giving customers access to the catalogue from anywhere in the world. The relevance of the development is determined by the need to create a specialised platform that takes into account the specifics of the vintage bicycle market. The application provides an effective presentation of goods, secure online transactions, and promotes the popularisation of cycling.

List of keywords: WEB-ORIENTED APPLICATION, VINTAGE BICYCLES, E-COMMERCE, JAVASCRIPT, PHP, MYSQL, BOOTSTRAP, RESPONSIVE DESIGN, USER INTERFACE, DATABASE, SHOPPING CART, PRODUCT CATALOGUE, AUTHORISATION, SECURE TRANSACTIONS.

ЗМІСТ

РЕФЕРАТ	3
ABSTRACT	4
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	7
ВСТУП.....	8
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАВДАННЯ	10
1.1. Загальні відомості з предметної галузі	10
1.1.1. Інноваційні аспекти електронної комерції	10
1.1.2. Характеристика галузі застосування	11
1.1.3. Основні технічні виклики при розробці вебдодатків з використанням Javascript та PHP	12
1.1.4. Огляд аналогічних вебдодатків.....	15
1.2. Призначення розробки та галузь застосування.....	21
1.3. Підстава для розробки	23
1.4. Постановка завдання.....	24
1.5. Вимоги до програми або програмного виробу.....	28
1.5.1. Вимоги до функціональних характеристик	29
1.5.2. Вимоги до інформаційної безпеки.....	30
1.5.3. Вимоги до складу та параметрів технічних засобів.....	31
1.5.4. Вимоги до інформаційної та програмної сумісності	32
РОЗДІЛ 2. ПРОЄКТУВАННЯ ТА РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ	34
2.1. Функціональне призначення системи	34
2.2. Опис застосованих математичних методів.....	35
2.3. Опис використаних технологій та мов програмування.....	36
2.3.1. Роль HTML та CSS у формуванні зовнішнього вигляду вебсторінок .	36
2.3.2. Аналіз мов програмування JavaScript та PHP у веброзробці.....	37
2.3.3. Обґрунтування доцільності обраного фреймворку та бібліотеки	39
2.3.4. Обґрунтування доцільності обраної СКБД.....	41
2.4. Опис структури системи та алгоритмів її функціонування	43

2.4.1. Алгоритми та структура проекту	43
2.4.2. Файлова структура проекту	51
2.4.3. Структура бази даних	53
2.5. Обґрунтування та організація вхідних та вихідних даних програми	60
2.6. Опис розробленої системи	64
2.6.1. Використані технічні засоби	64
2.6.2. Використані програмні засоби	64
2.6.3. Виклик та завантаження програми	65
2.6.4. Опис інтерфейсу користувача	66
РОЗДІЛ 3. ЕКОНОМІЧНИЙ РОЗДІЛ	76
3.1. Розрахунок трудомісткості та вартості розробки програмного продукту	76
3.2. Рахунок витрат на створення програми	80
ВИСНОВКИ	83
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	85
Додаток А. Лістинг програми	88
Додаток Б. Відгук керівника економічного розділу	110
Додаток В. Перелік файлів на диску	111

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

HTML – HyperText Markup Language

CSS – Cascading Style Sheets

API – Application Programming Interface

БД – База даних

ОС – Операційна система

SQL – Structured Query Language

RAM – Random-Access Memory

HTTPS - Hypertext Transfer Protocol Secure

HTTP - Hypertext Transfer Protocol

ВСТУП

В епоху стрімкого розвитку інформаційних технологій та зростання попиту на онлайн-торгівлю, розробка веборієнтованих додатків для електронної комерції набуває все більшої актуальності. Зокрема, ринок вінтажних товарів, таких як старовинні велосипеди, демонструє значний потенціал для застосування сучасних вебтехнологій з метою розширення можливостей продажу та придбання унікальних виробів.

Актуальність даної роботи полягає в необхідності створення спеціалізованої онлайн-платформи для продажу вінтажних велосипедів, яка б враховувала специфіку цього ринку та задовольняла потреби як продавців, так і покупців. Традиційні методи торгівлі часто виявляються недостатньо ефективними для реалізації таких унікальних товарів, обмежуючи можливості їх представлення та доступність для потенційних покупців.

Об'єктом дослідження є процес розробки веборієнтованого додатку для продажу вінтажних велосипедів, який включає в себе аналіз вимог, проектування архітектури, вибір технологій та реалізацію функціоналу.

Метою даної кваліфікаційної роботи є створення веборієнтованого додатку для продажу вінтажних велосипедів з використанням JavaScript та PHP. Досягнення цієї мети дозволить вирішити ряд важливих завдань:

- забезпечити ефективний канал збуту для продавців вінтажних велосипедів, розширюючи їх аудиторію та географію продажів;
- надати покупцям зручний інструмент для пошуку, вибору та придбання унікальних велосипедів;
- створити платформу для обміну інформацією між ентузіастами вінтажних велосипедів;
- сприяти збереженню та популяризації велосипедного транспорту.

Використання JavaScript та PHP як основних технологій для розробки вебдодатку забезпечить необхідну функціональність та продуктивність.

JavaScript забезпечить високу інтерактивність та зручність користувацького інтерфейсу, тоді як PHP дозволить ефективно організувати серверну частину, взаємодію з базою даних та обробку замовлень.

Практичне значення полягає у створенні функціонального вебдодатку, який забезпечить ефективну діяльність інтернет-магазину вінтажних велосипедів, надаючи зручний інструмент для продажу та придбання таких унікальних товарів. Розроблений додаток дозволить компаніям розширити ринок збуту, а покупцям - отримати доступ до каталогу вінтажних велосипедів з будь-якої точки світу.

Таким чином, розробка веборієнтованого додатку з продажу вінтажних велосипедів за допомогою JavaScript та PHP є актуальним завданням, яке відкриває нові можливості для розвитку електронної комерції у специфічній ніші колекційних товарів.

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАВДАННЯ

1.1. Загальні відомості з предметної галузі

1.1.1. Інноваційні аспекти електронної комерції

Розвиток електронної комерції тісно пов'язаний з впровадженням технологічних рішень, що покращують користувацький досвід та оптимізують бізнес-процеси.

[1] У контексті веборієнтованого додатку з продажу вінтажних велосипедів можна виділити наступні інноваційні аспекти:

- адаптивний вебдизайн: Це підхід до вебдизайну, який забезпечує оптимальне відображення сайту на різних пристроях. [2] Вебсторінки автоматично підлаштовуються під розмір екрану, будь то настільний комп'ютер, планшет чи смартфон. Це досягається за допомогою гнучких макетів, масштабованих зображень та CSS медіа-запитів. Адаптивний дизайн покращує користувацький досвід та збільшує конверсію, оскільки відвідувачі можуть зручно переглядати сайт з будь-якого пристрою;

- розширені можливості пошуку: Впровадження фільтрів за специфічними характеристиками вінтажних велосипедів (рік випуску, марка, модель, стан) дозволяє користувачам швидко знаходити потрібні товари. Це значно спрощує процес пошуку та підвищує ймовірність здійснення покупки;

- інтерактивні галереї товарів: Використання JavaScript дозволяє створювати динамічні галереї зображень, де користувачі можуть масштабувати фотографії, переглядати товар під різними кутами. Така функція підвищує довіру покупців та зменшує кількість повернень товару [3];

- безпечні онлайн-транзакції: Використання протоколів SSL/TLS забезпечує шифрування даних, що передаються між браузером користувача та сервером сайту [4]. Це захищає персональні та платіжні дані від перехоплення.

Наявність захищеного з'єднання (позначається замком у адресному рядку браузера) підвищує довіру користувачів до сайту.

Ці інноваційні аспекти спрямовані на підвищення ефективності продажів, покращення користувацького досвіду та забезпечення конкурентних переваг вебдодатку на ринку електронної комерції вінтажних велосипедів

1.1.2 Характеристика галузі застосування

Тематику кваліфікаційної роботи можна віднести до галузі електронної комерції та веброзробки. Для більш детального класифікування предметної області варто розглянути наступні підсекції:

- електронна комерція (е-комерція). Електронна комерція охоплює всі аспекти купівлі та продажу товарів і послуг через Інтернет, а також пов'язані бізнес-процеси, які забезпечують ефективну роботу онлайн-торгівлі. Це включає розміщення товарів на вебсайті, обробку замовлень, приймання платежів, організацію доставки, маркетинг і рекламу [5]. Розробка вебдодатку для продажу вінтажних велосипедів безпосередньо відноситься до сфери е-комерції. Такий додаток повинен мати зручний інтерфейс користувача, інтегровані платіжні системи та функціонал управління замовленнями;

- веброзробка. Ця підгалузь відіграє ключову роль у розробці веборієнтованого додатку для продажу вінтажних велосипедів. Вона охоплює процеси створення, проектування, розробки та підтримки вебсайтів і вебдодатків, які доступні через мережу Інтернет;

- розробка користувацького інтерфейсу (UI/UX). Ця підсекція є вкрай важливою для успішної реалізації вебдодатку з продажу вінтажних велосипедів. Вона стосується проектування та розробки зручних і ефективних інтерфейсів для взаємодії користувачів з вебдодатками та вебсайтами.

електронний маркетинг та просування: ця підгалузь відіграє ключову роль у забезпеченні успішного функціонування вебдодатку. Вона охоплює методи та стратегії просування товарів і послуг в Інтернеті.

Отже, проєкт кваліфікаційної роботи представляє собою комплексний веборієнтований додаток для продажу вінтажних велосипедів, який інтегрує різні аспекти електронної комерції, веброзробки та цифрового маркетингу. Як електронна торгова платформа, додаток включає розробку користувацького інтерфейсу, забезпечення зручності використання та позитивного досвіду користувачів. Крім того, для ефективного просування та залучення цільової аудиторії додаток має враховувати стратегії пошукової оптимізації, інтернет-реклами та маркетингу в соціальних мережах.

1.1.3. Основні технічні виклики при розробці вебдодатків з використанням Javascript та PHP

Забезпечити належну швидкодію додатку - одне з ключових завдань для комфорту користувачів при відвідуванні сайту [6]. Тому у межах виконання кваліфікаційної роботи були застосовані методи оптимізації кодових файлів:

- було мінімізовано розмір JavaScript файлів, видаливши зайві пробіли та коментарі. Це значно зменшить обсяг даних для передачі клієнту;
- було розбито великі JavaScript файли на дрібніші модулі. Наприклад, код галереї фото велосипеда спершу не підвантажується, це відбувається при переході на відповідну сторінку.

Для критично важливих компонентів, як от кошик покупок, забезпечення плавної взаємодії між клієнтською (JavaScript) та серверною (PHP) частинами є ключовим. Використовуючи AJAX, JavaScript відправлятиме дані форм на сервер, де PHP оброблятиме їх (валідація, збереження в базі даних тощо) та повертатиме оновлені дані у відповідь для рендерингу на клієнті без перезавантаження сторінки.

Отже, продуктивність та масштабованість - це не одноразові виклики, а безперервні процеси оптимізації та архітектурних змін, покликаних забезпечити

високу швидкість та стабільність роботи вашого вебдодатку незалежно від його розмірів чи кількості користувачів.

Забезпечення належної роботи вебдодатку на різних платформах є основною вимогою у веброзробці. Потенційні користувачі можуть взаємодіяти з додатком з усіх найпопулярніших операційних систем, таких як наприклад Windows або Mac OS, Linux. Більш того, додаток розроблений з використанням методик адаптивного дизайну, що дозволяє цільовій аудиторії використовувати його на мобільних пристроях Android та iOS.

Для того, щоб довести вище сказане, додаток був ретельно протестований у різних сучасних браузерах, а саме Chrome, Firefox, Safari, Edge (рис.1.1.).

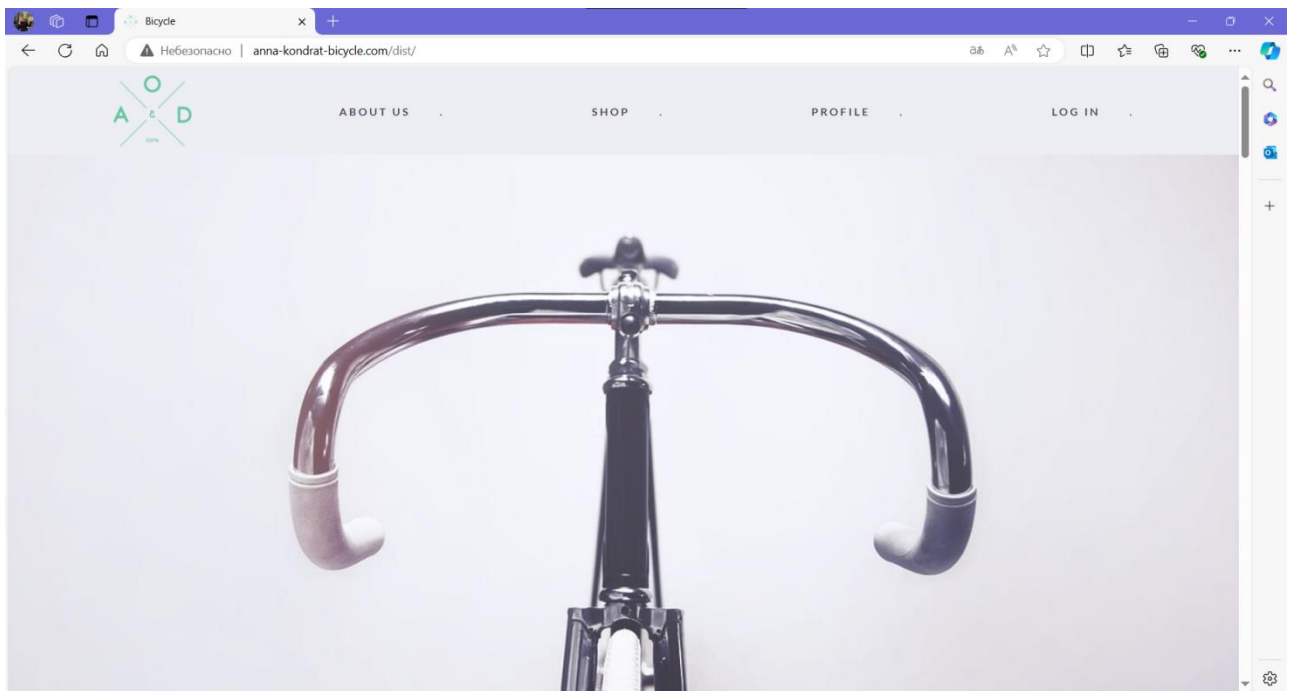


Рис. 1.1. Результат тестування головної сторінки вебдодатку «Bicycle» у браузері «Edge»

Не можна ігнорувати старіші версії браузерів, які все ще поширені серед користувачів. Для підтримки таких браузерів використовуються техніки транспіляції. Під час розробки було виконано транспіляцію за допомогою Babel, що дозволяє конвертувати сучасний JavaScript код у більш старий різновид,

сумісний з попередніми версіями браузерів. У наслідок використанні цієї техніки вебдодаток коректно відображається на усіх розширеннях екрану, починаючи з 320 px.

Крім того було використано CSS-фреймворк Bootstrap для реалізації адаптивного компоненту системи оплати. Завдяки вбудованій системі розмітки, компонент динамічно змінюється залежно від роздільної здатності екрану.

JavaScript також відіграє важливу роль у досягненні адаптивності. Наприклад, на мобільних пристроях доцільно приховувати деякі елементи, замінюючи їх компактнішими іконками. Прикладом є заміна навігаційного меню на іконку «бургер-меню» (рис.1.2.).

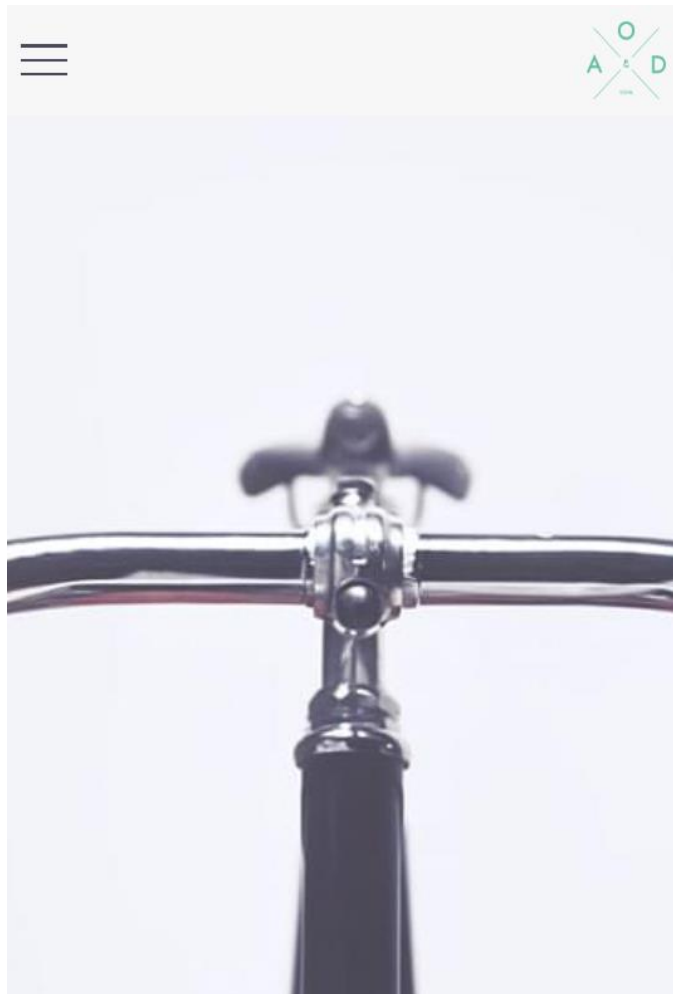


Рис. 1.2. Здійснене «Бургер-меню» для мобільних пристроїв

Отже, ретельне тестування, застосування перевірених технік та бібліотек дозволяє досягнути ідеальної сумісності з усіма пристроями та браузерами.

1.1.4. Огляд аналогічних вебдодатків

З метою дослідити сучасний стан вебсайтів з продажу вінтажних велосипедів, було проведено аналіз декількох популярних платформ. Цей аналіз включає в себе оцінку їхніх переваг, недоліків, технічних аспектів та наявних прогалин.

Одним з вебсайтів у цій сфері є "buycycle" (рис. 1.3.). Цей онлайн-платформа пропонує широкий спектр велосипедів від сотень продавців Німеччини.

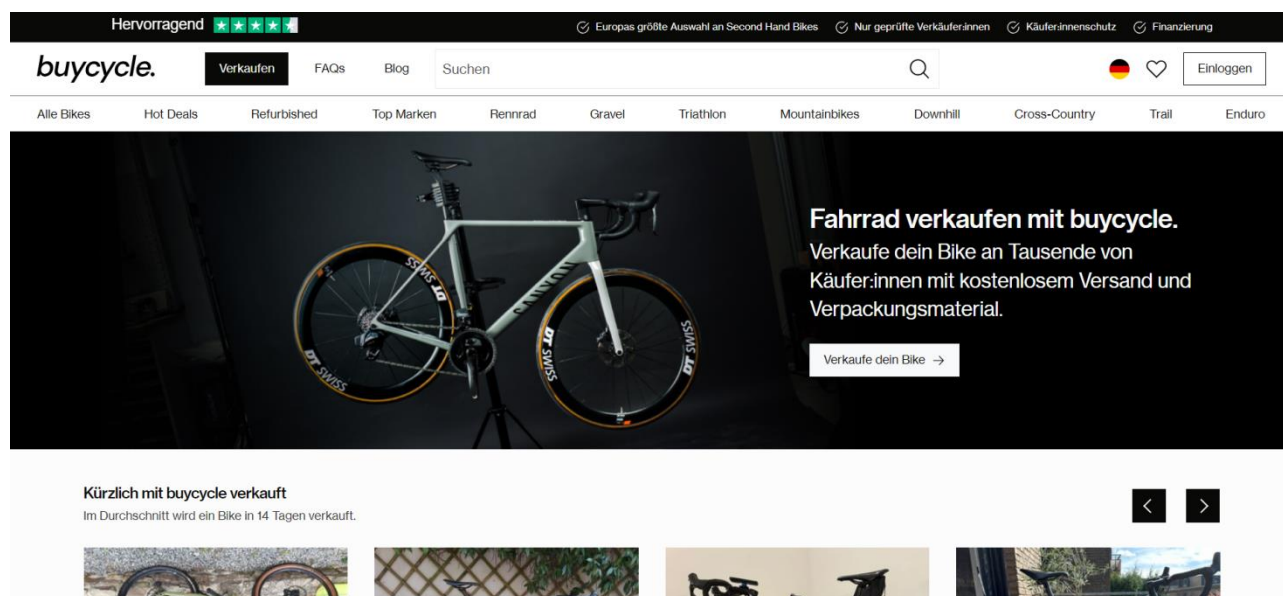


Рис. 1.3. Головна сторінка вебсайту «buycycle»

На сайті представлені детальні фотографії, технічні характеристики кожного велосипеда, що продається, а також докладна історична довідка про модель та бренд. Користувачі можуть легко знайти потрібний велосипед за допомогою розширених фільтрів пошуку. Зручна навігація за категоріями також полегшує перегляд асортименту.

Сайт доступний англійською, що робить його зручним для міжнародної аудиторії потенційних покупців вінтажних велосипедів.

З технічної точки зору, вебсайт buycycle.de має класичну структуру з верхнім меню (header) та нижнім колонтитулом (footer), що забезпечує зручну навігацію між розділами сайту. Адаптивний дизайн гарантує коректне відображення на пристроях розширенням від 320 px. Сайт складається з декількох сторінок, присвячених різним аспектам купівлі-продажу велосипедів, інформації про компанію тощо.

Одна з ключових можливостей - інтерактивна форма створення оголошення про продаж велосипеда. Сайт інтегрований з базою даних для зберігання інформації про велосипеди, їх характеристики, користувачів тощо. Також реалізовано алгоритм для рекомендації оптимального варіанту на основі моделі, року випуску, стану та інших факторів. Для потенційних покупців присутня потужна пошукова система з розширеними фільтрами пошуку велосипедів за маркою, моделлю, типом, ціною тощо. Після вибору є можливість додати товар у кошик та оформити замовлення з інтеграцією платіжних систем для приймання оплат.

З позиції інформаційної безпеки не забуто і про безпеку - реалізований надійний функціонал для керування обліковими записами користувачів, здебільшого завдяки валідації даних на клієнтській та серверній сторонах. Крім того, передача даних відбувається через захищені канали протоколу HTTPS. Також здійснюється оптимізація для пошукових систем (SEO) та аналітика поведінки користувачів. Для зручності користувачів присутні форми зворотнього зв'язку, оплати, взаємодії з продуктами тощо.

Необхідно зауважити також потенційні технічні недоліки аналога: недостатня оптимізація швидкості завантаження сторінок (рис.1.4.). Візуальний аналіз не виявляє застосування сучасних технік оптимізації продуктивності, як код-сплітінг, відкладене завантаження ресурсів.

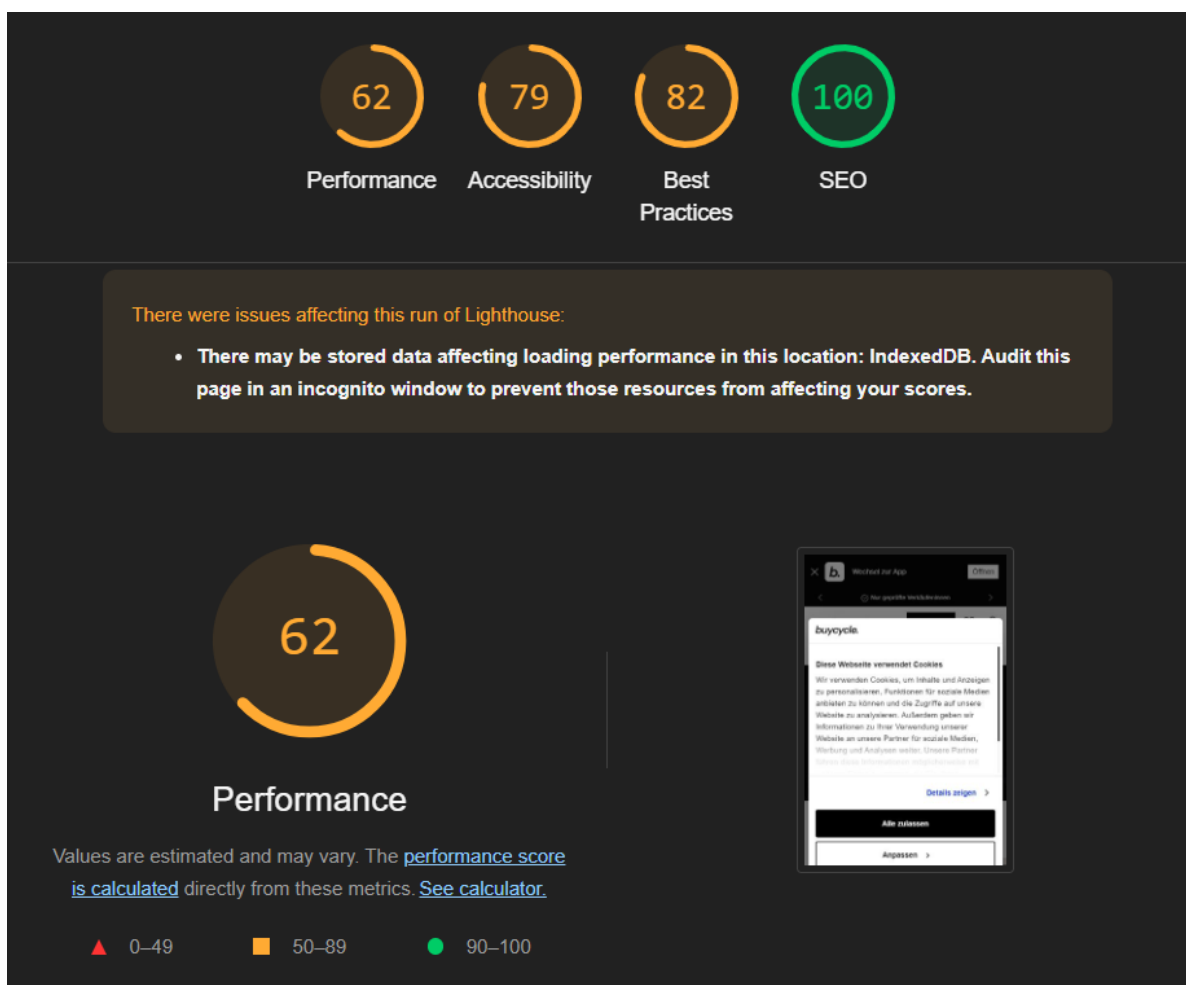


Рис. 1.4. Результати аналізу продуктивності вебсайту «buycycle.de» за допомогою інструменту Lighthouse

Проаналізувавши продуктивність вебсайту «buycycle.de», було виявлено низькі показники у таких пунктах:

- Performance (Продуктивність) - 62 бали. Цей показник оцінює швидкість завантаження та відображення сторінки. Оцінка 62 бали вказує на те, що сайт має помірні проблеми з продуктивністю [13]. На погіршення цього показника впливають:
 - великі нестиснені зображення: Зображення високої роздільної здатності без оптимізації можуть суттєво сповільнити завантаження сторінки;
 - невідоптимізований JavaScript і CSS: Великі файли скриптів та стилів, особливо якщо вони не мініфіковані або не об'єднані, збільшують час завантаження;

- повільні відповіді сервера: Якщо сервер повільно обробляє запити, це збільшує час до першого байту (TTFB);
- блокуючі ресурси: Скрипти або стилі, які завантажуються синхронно, можуть затримувати відображення контенту.
- Accessibility (Доступність) - 79 балів. Цей показник оцінює, наскільки сайт доступний для користувачів з різними обмеженнями, включаючи людей з вадами зору, слуху або моторики [14]. Що впливає на погіршення :
 - відсутність альтернативного тексту для зображень: Це ускладнює розуміння контенту для користувачів, які використовують програми читання з екрану;
 - низький контраст кольорів: Може ускладнити читання тексту для людей з вадами зору;
 - неправильна структура заголовків: Ускладнює навігацію по сторінці для користувачів програм читання з екрану;
 - неправильне використання ARIA-атрибутів: Може призвести до неправильної інтерпретації елементів сторінки допоміжними технологіями.
- Best Practices (Найкращі практики) - 82 бали. Цей показник оцінює дотримання сучасних стандартів веброзробки та безпеки [15]. Фактори, які можуть впливати на погіршення цього показнику:
 - використання застарілих API: Може призвести до проблем з сумісністю та безпекою;
 - небезпечні джерела завантаження ресурсів: Завантаження ресурсів з ненадійних джерел може становити загрозу безпеці;
 - відсутність HTTPS: Незашифровані з'єднання можуть бути вразливими до атак;
 - проблеми з безпекою JavaScript-бібліотек: Використання застарілих версій бібліотек може містити відомі вразливості;

– покращення: Оновлення до сучасних API, використання HTTPS, регулярне оновлення бібліотек та перевірка джерел ресурсів можуть покращити цей показник.

Наступним з вебсайтів у сфері продажу велосипедів є "Titanbike" (рис.1.5.). Це інтернет-магазин, який спеціалізується на реалізації продажу велосипедів та велосипедних аксесуарів в Україні.

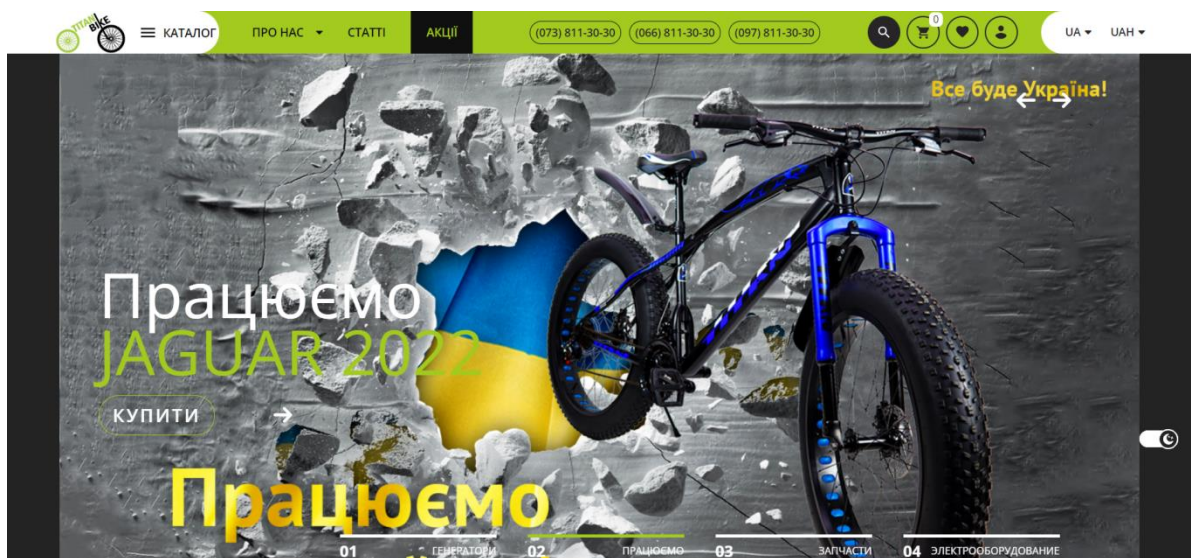


Рис. 1.5. Головна сторінка аналогу «Titanbike»

Titanbike пропонує широкий асортимент велосипедів різних категорій.. Користувачі можуть переглядати товари, фільтрувати їх за різними параметрами (ціна, бренд, діаметр коліс, тощо) та сортувати результати. Користувачі можуть додавати товари у віш-лист чи кошик для покупки та оформляти замовлення онлайн. Покупцям надається можливість вибрати спосіб доставки (кур'єрські служби, самовивіз) та оплати замовлення.

Сайт має responsive дизайн, що гарантує коректне відображення контенту та функціоналу на різних пристроях - від десктопів до мобільних гаджетів. Для зручності покупців реалізовано інтуїтивну систему взаємодії з кошиком, де можна тимчасово зберігати обрані товари перед оформленням замовлення. Зареєстровані користувачі можуть переглядати історію своїх минулих покупок.

Забезпечено належний рівень безпеки - використовується HTTPS для захисту конфіденційних даних покупців під час оформлення замовлень. Додатково на сайті присутня підтримка кількох мов (української та англійської). Загалом, застосовано різні підходи для максимальної зручності навігації, презентації товарів, керування процесом покупки та додаткового інформування відвідувачів, але аналог має недоліки та прогалини:

- відсутність можливості змінити пароль за бажанням. Мій додаток передбачає таку функцію, що спростить процес реєстрації та залучення користувачів;

- недостатня автоматизація процесів керування товарним каталогом. У моєму додатку планується впроваджена автоматизовану систему оновлення цін і товарів в режимі реального часу без необхідності ручного редагування;

- обмежені можливості персоналізації інтерфейсу та досвіду користувача. Мій додаток має функцію збереження персональних налаштувань, історії переглядів, вішлистів для кожного зареєстрованого користувача.

Аналіз вебсайтів "buucycle" та "Titanbike" виявив ключові функціональні особливості та технічні рішення, важливі для розробки власного вебдодатку з продажу вінтажних велосипедів, тож основні функції, які були реалізовані при розробці мого додатку:

- система реєстрації та авторизації з можливістю зміни паролю;
- розширена система фільтрації та пошуку товарів;
- функціонал кошика покупок та оформлення замовлень;
- персоналізація інтерфейсу (історія замовлень, віш-листи);
- адаптивний дизайн.

Для забезпечення інформаційної безпеки буде використано:

- протокол HTTPS;
- валідація даних на клієнтській та серверній сторонах;
- безпечне управління обліковими записами.

Особлива увага була приділена оптимізації швидкості завантаження сторінок та використанню сучасних технік підвищення продуктивності.

Реалізація цих елементів дозволить створити зручний та безпечний вебдодаток для купівлі-продажу вінтажних велосипедів, що враховує потреби всіх користувачів.

1.2. Призначення розробки та галузь застосування

У бакалаврській роботі розглядається веборієнтований додаток для продажу вінтажних велосипедів, з метою створення спеціалізованої платформи для продажу та купівлі унікальних велосипедів.

Фактори, що зумовлюють потребу у вебдодатку для вінтажних велосипедів:

- відсутність спеціалізованої платформи. На ринку бракує централізованого ресурсу, орієнтованого саме на продаж вінтажних велосипедів, що враховував би всі особливості цього сегменту;

- потреба в ефективній презентації товару. Вінтажні велосипеди вимагають детального опису, якісних фотографій, чого часто не вистачає на загальних платформах електронної комерції;

- забезпеченні безпечних онлайн-транзакцій між продавцями та покупцями;

- сприянні збереженню та популяризації велосипедного транспорту.

Призначення розробки веборієнтованого додатку “Bicycle” полягає в:

- створенні спеціалізованої платформи для продажу вінтажних велосипедів;

- забезпеченні зручного інструменту для пошуку та придбання унікальних моделей велосипедів;

- наданні можливості детальної презентації товару;

- забезпеченні безпечних онлайн-транзакцій між продавцями та покупцями;

- створенні бази даних вінтажних велосипедів для збереження інформації про моделі;

- сприянні збереженню та популяризації велосипедного транспорту.

Зважаючи на вищенаведені приклади, можна зробити висновок, що вебдодаток для продажу вінтажних велосипедів може бути цінним інструментом для дилерів антикварних товарів та любителів велосипедів.

Основна термінологія містить терміни з областей електронної комерції та веброзробки. Ключові слова: веборієнтований додаток, електронна комерція, вінтажні велосипеди, онлайн-платформа, користувацький інтерфейс, база даних, система управління контентом (CMS), безпечні транзакції, адаптивний дизайн, SEO-оптимізація, API, хостинг.

Терміни, використані у бакалаврській роботі, і потрібні задля розуміння елементної та технічної структури системи:

- каталог товарів – сторінка з переліком доступних вінтажних велосипедів;

- картка товару – детальний опис конкретного велосипеда з фотографіями та характеристиками;

- кошик покупок – віртуальний кошик для зберігання вибраних товарів перед оформленням замовлення;

- система фільтрації – інструмент для пошуку велосипедів за різними параметрами (рік випуску, марка, стан тощо);

- особистий кабінет покупця - частина системи, призначена для управління покупками, відстеження замовлень та збереження улюблених товарів;

- система безпечних платежів – механізм для проведення безпечних онлайн-транзакцій;

- адміністративна панель – інтерфейс для управління контентом та користувачами платформи;

- автентифікація/Авторизація - процес реєстрації, входу та перевірки облікових даних користувачів;
- хешування паролів - використання алгоритму SHA512 для безпечного зберігання паролів у зашифрованому вигляді;
- валідація форм - перевірка коректності вхідних даних з форм реєстрації/логіну перед відправкою на сервер;
- сесії - збереження даних сесії користувача, таких як дата/час останнього входу, роздільна здатність екрану, ОС;
- AJAX - асинхронні виклики для перевірки онлайн-статусу інших користувачів без перезавантаження сторінки;
- база даних - використання SQL бази даних (MySQL, MariaDB) для зберігання інформації про користувачів, товари, замовлення;
- REST API - можливо, використання REST API для отримання/змінення даних на сервері;
- локалізація - підтримка української мови, спеціальних символів у даних.

1.3. Підстава для розробки

Підставами для розробки (виконання кваліфікаційної роботи) є:

- освітня програма спеціальності 122 «Комп'ютерні науки»;
- навчальний план та графік навчального процесу;
- наказ ректора Національного технічного університету «Дніпровська політехніка» №469-с від 23.05.2024;
- завдання на кваліфікаційну роботу на тему «Розробка веборієнтованого додатку з продажу вінтажних велосипедів за допомогою JavaScript та PHP»;

1.4. Постановка завдання

У межах бакалаврської роботи розглядається створення веборієнтованого додатку з продажу вінтажних велосипедів за допомогою JavaScript та PHP «Bicycle».

Мета та призначення завдання: Розробити функціональний вебсайт для продажу вінтажних велосипедів з використанням JavaScript та PHP. Сайт повинен надавати зручний інтерфейс для перегляду каталогу товарів, здійснення покупок, відстеження замовлень тощо.

Технічна сутність завдання: створення функціонального та зручного у використанні вебсайту з комплексом можливостей для електронної комерції - каталог товарів, кошик покупок, система обробки замовлень та платежів, особистий кабінет користувача тощо. Це вимагає застосування сучасних вебтехнологій, таких як JavaScript, PHP, баз даних, платіжних шлюзів тощо.

З економічної точки зору, розроблюваний вебдодаток має на меті:

- відкрити новий канал продажів вінтажних велосипедів через інтернет, що розширює коло потенційних клієнтів і збільшує обсяги продажів;
- знизити операційні витрати на ведення бізнесу порівняно з традиційною офлайн торгівлею за рахунок відсутності необхідності в орендованих приміщеннях, персоналі тощо;
- забезпечити цілодобову доступність магазину та зручність покупок для клієнтів без прив'язки до місцезнаходження;
- покращити комунікацію з клієнтами та надавати більше інформації про товари;
- отримати інструменти для аналізу попиту.навчальний план і графік навчального процесу.

Розроблюваний вебдодаток може бути інтегрований та використовуватися різними суб'єктами, пов'язаними з велосипедною галуззю та торгівлею, зокрема:

- виробники та продавці вінтажних велосипедів. Вебдодаток стане новим каналом збуту вінтажних велосипедів через онлайн-продажі, що дозволить розширити клієнтську базу та збільшити обсяги реалізації;
- сервісні центри з ремонту та обслуговування велосипедів. Інтеграція вебдодатку може дозволити їм розміщувати інформацію про свої послуги, ціни та отримувати замовлення від клієнтів онлайн;
- магазини велозапчастин та аксесуарів. Аналогічно до продавців велосипедів, вони зможуть використовувати вебдодаток як торгову платформу;
- велоклуби та спільноти. Вебдодаток може містити розділ з інформацією про велоклуби, їхні заходи, можливість приєднання до спільноти тощо.

Таким чином, вебдодаток дозволить об'єднати всю велосипедну екосистему на одній платформі, сприяючи ефективній взаємодії між учасниками ринку.

Структура та показники об'єктів: Сайт має структуру типового інтернет-магазину з розділами каталогу товарів, кошика, оформлення замовлення, особистого кабінету користувача тощо. Ключовими показниками будуть кількість відвідувачів, конверсія продаж, найпопулярніший тип доставки.

Призначення вихідної інформації: Вихідною інформацією для користувачів буде каталог товарів, інформація про наявність, ціни, можливість створення замовлення.

Вимоги до вхідної інформації: Користувачі повинні мати можливість реєструватися, авторизуватися, вносити дані замовлення та оплати. Для забезпечення ефективного збору вхідних даних від клієнтів необхідно розробити зручні форми для реєстрації та створення облікового запису з інтуїтивним інтерфейсом. Кошик замовлень повинен мати логічну структуру для додавання, видалення чи зміни кількості товарів. Форма оформлення самого замовлення має містити чіткі та зрозумілі поля для внесення інформації про доставку, оплату тощо. Для адміністраторів передбачається можливість завантаження фото та

відео товарів до каталогу. Обов'язковою є система валідації введених даних з перевіркою на коректність та наявність обов'язкових полів. Разбить эту инфу на вимоги та призначення

Припинення автоматизованого розв'язання: Процес оформлення замовлення може бути припинений користувачем на будь-якому етапі до підтвердження оплати.

Зв'язок з іншими системами: Інтеграція з вбудованою платіжною системою Bootstrap5, яку пропоную CSS-фреймворк Bootstrap5.

Структурні об'єкти та показники, що характеризують їх стан, можна описати наступним чином:

- каталог товарів. Кількість товарних позицій, розподіл за моделями, ціновий діапазон, наявність фото та описів, оновлення асортименту;
- клієнтська база. Кількість зареєстрованих користувачів, активних користувачів, нових реєстрацій за період, дані про покупки клієнтів, історія замовлень;
- кошик покупок. Кількість активних кошиків, середня вартість кошика, конверсія в замовлення, залишені кошики;
- замовлення. Кількість оформлених замовлень, сума продажів, спосіб доставки, статуси замовлень;
- платежі. Розподіл платежів за способами оплати;
- пошук і навігація. Популярні пошукові запити, шляхи переходів на сайті, джерела трафіку;
- взаємодія з клієнтами. Повідомлення на пошту щодо потреби користувачів у зміні важливих інформаційних даних а також підтверджень їїніх замовлень;
- технічна інфраструктура. Швидкість завантаження сторінок, інформаційна безпека;
- якість контенту. Релевантність назв, описів товарів, візуального контенту, інтуїтивний інтерфейс.

Ці показники допомагають відстежувати ефективність роботи вебдодатку, виявляти проблемні області та можливості для покращення.

Розподіл функцій між користувачами та технічними засобами для вебдодатку з продажу вінтажних велосипедів може виглядати наступним чином (рис.1.6.):

- для незареєстрованих відвідувачів:
 - перегляд каталогу товарів без можливості оформлення замовлення;
 - реєстрація облікового запису шляхом надання емейл, пароллю, імені/прізвища.
- для зареєстрованих користувачів:
 - Авторизація за допомогою емейл/пароллю.
 - Перегляд та редагування особистого профілю (ім'я, контактні дані тощо).
 - Додавання товарів до кошика з можливістю змінювати кількість.
 - Оформлення замовлення: введення адреси доставки, вибір способу доставки та оплати.
 - Перегляд історії замовлень та можливість повторити попереднє в один клік.
 - Відстеження статусу оформлених замовлень.
 - Отримання підтверджуючого емейлу після оформлення замовлення.
- технічні засоби:
 - Отримання підтверджуючого емейлу після оформлення замовлення;
 - валідація вхідних даних при реєстрації, оформленні замовлень;
 - шифрування паролів користувачів (SHA512);
 - інтеграція з платіжними системами для приймання оплати;

- автоматична генерація номеру замовлення;
- відправка підтверджуючих емейлів після оформлення замовлення.

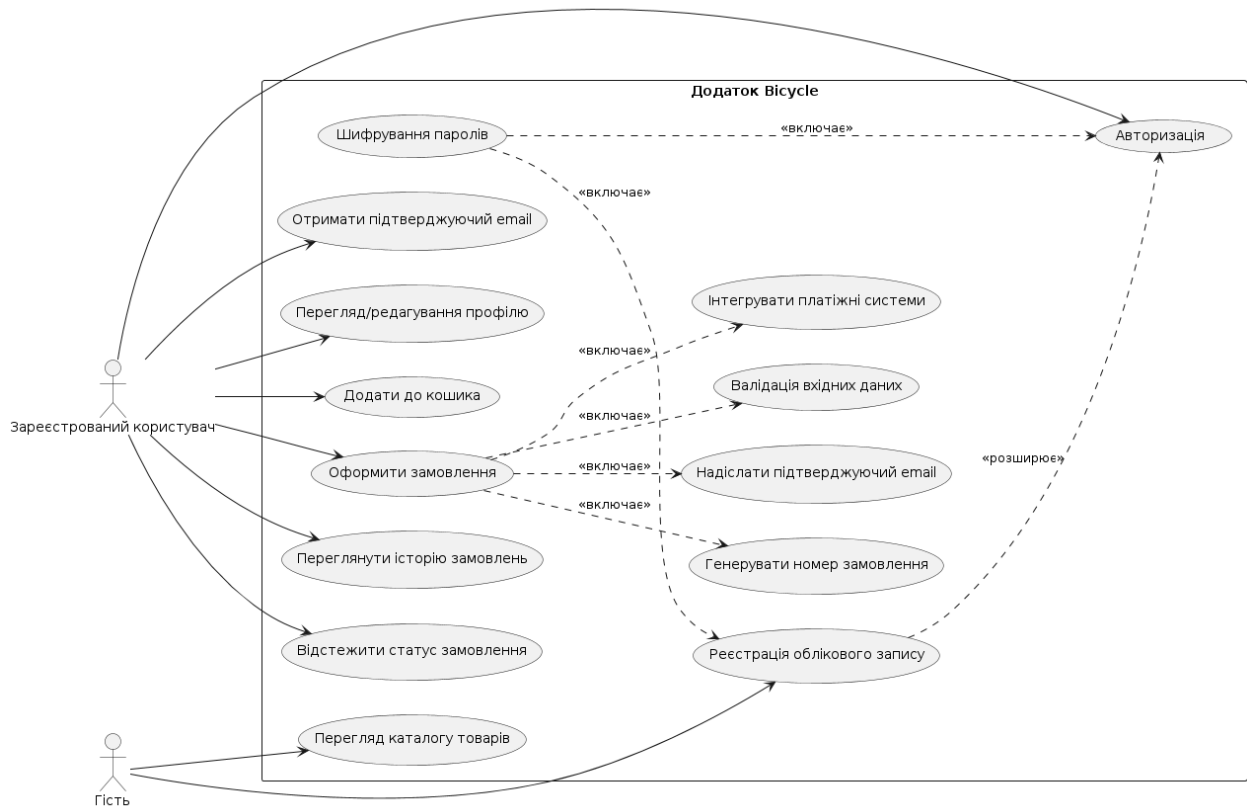


Рис. 1.6. Діаграма прецедентів Вісцле

Такий розподіл забезпечує зручне виконання всіх необхідних функцій для наявних типів користувачів.

1.5. Вимоги до програми або програмного виробу

Вебдодаток повинен ефективно забезпечувати збір, зберігання, пошук, представлення, обробку та передачу інформації користувачам, запобігаючи виникненню збоїв та помилок. Основна функція додатку – надання клієнтам можливості переглядати каталог вінтажних велосипедів, здійснювати пошук за параметрами, формувати замовлення та оформляти покупки. Це має

реалізовуватися з дотриманням вимог надійності, продуктивності, високої швидкодії та гнучкої модульної архітектури.

Програмний продукт повинен бути орієнтований на користувача, забезпечуючи максимальну зручність в електронній комерції: легке сприйняття інформації, інтуїтивний та простий інтерфейс, зрозумілість функцій та процесів. Додаток має забезпечувати роботу з базами даних клієнтів, товарів та замовлень, надійно зберігаючи й обробляючи відповідну інформацію та бути економічно ефективним, забезпечуючи прибутковість бізнесу з продажу вінтажних велосипедів. Архітектура повинна бути гнучкою для внесення змін, доповнень чи оновлень функціоналу без впливу на інші компоненти.

1.5.1. Вимоги до функціональних характеристик

Щоб досягти мети, поставленої в роботі, вебдодаток має забезпечувати:

- Віддалений доступ через веббраузер для всіх користувачів, незалежно від їх місцезнаходження та типу пристрою (ПК, планшет, смартфон).
- Високу швидкість та оперативність у роботі основних функцій, таких як перегляд каталогу товарів, оформлення замовлень, авторизація тощо. Час відгуку системи не повинен перевищувати 100 мілісекунд.
- Гнучку та модульну архітектуру, що дозволить легко вносити зміни, додавати або оновлювати функціональні компоненти без негативного впливу на інші частини системи.
- Надійність та стабільність у роботі, мінімізуючи ризики збоїв або відмов. База даних повинна бути стійкою до втрати або пошкодження інформації, її блокування або несанкціонованого редагування.
- Зберігання всіх даних додатку (інформація про товари, користувачів, замовлення тощо) у реляційній базі даних із забезпеченням цілісності та узгодженості інформації.

1.5.2. Вимоги до інформаційної безпеки

Інформаційна безпека – це стан захищеності інформаційних систем, даних та інформаційних ресурсів від несанкціонованого доступу, використання, розголошення, порушення цілісності, знищення або інших загроз [16].

За аспектом інформаційної безпеки, на який спрямовані загрози, виділяють наступні різновиди загроз [17]:

- загрози конфіденційності, а саме несанкціонований доступ до інформації: Коли конфіденційна інформація стає доступною тим, хто не має на це права. Це може статися через кібератаки, витік даних або людську помилку. Приклади: Взлом комп'ютерної системи, крадіжка мобільного телефону, несанкціоноване стеження.

- загрози цілісності, а саме неправомірна зміна даних: Зміна або руйнування інформації зловмисним або ненавмисним способом. Це може призвести до неточних або оманливих даних. Приклади: Кібератаки, комп'ютерні віруси, людська помилка.

- Загрози доступності, а саме блокування або ускладнення доступу до ресурсів: Коли уповноважені користувачі не можуть отримати доступ до необхідних їм даних або систем. Це може бути спричинено DoS/DDoS-атаками, збоями в роботі обладнання або програмними помилками. Приклади: Атаки відмови в обслуговуванні (DoS/DDoS), перебої в електропостачанні, стихійні лиха.

Ці різновиди загроз інформаційної безпеки необхідно враховувати та впроваджувати відповідні контрзаходи для захисту систем від можливих порушень конфіденційності, цілісності та доступності даних.

Що стосується загрози конфіденційності (неправомірний доступ до інформації), у рамках мого вебдодатку з продажу вінтажних велосипедів було зроблено наступне: використано хешування SHA-512 для паролів користувачів, що захищає їх від витіку в незашифрованому вигляді, а також обмежено доступ

до вебдодатку лише для авторизованих користувачів, щоб запобігти несанкціонованому доступу до конфіденційних даних.

Щодо загрози цілісності даних (неправомірна зміна даних), були впроваджені такі заходи: перевірка унікальності імені користувача (електронної адреси) під час реєстрації для запобігання створенню дублікатів облікових записів, а також вимога до валідації вхідних даних на клієнтській стороні перед відправкою на сервер, що захищає від ін'єкційних атак і пошкодження даних.

Крім того, як додатковий захід безпеки, було впроваджено вимогу до складності паролів (мінімальна довжина, використання літер різних регістрів та цифр), що підвищує стійкість до зламу та сприяє кращій конфіденційності даних користувачів.

1.5.3. Вимоги до складу та параметрів технічних засобів

Мінімальні вимоги для роботи вебдодатку на персональних комп'ютерах:

- двоядерний процесор з тактовою частотою від 2 ГГц;
 - 4 ГБ оперативної пам'яті;
 - операційна система Windows 10, macOS 10.13 або новіша версія, або сучасний дистрибутив Linux;
 - сучасний веббраузер з підтримкою HTML5, CSS3 та ECMAScript 2015 (ES6) (наприклад, Google Chrome, Mozilla Firefox, Microsoft Edge, Safari);
 - роздільна здатність екрану не менше 1366x768 пікселів;
 - стабільне широкосмугове з'єднання з Інтернетом.
- Мінімальні вимоги для роботи вебдодатку на мобільних пристроях:
- чотириядерний процесор з тактовою частотою від 1,5 ГГц;
 - 3 ГБ оперативної пам'яті;
 - операційна система Android 8.0 або новіша версія, або iOS 12 або новіша версія;
 - сучасний мобільний веббраузер з підтримкою HTML5, CSS3 та ECMAScript 2015 (ES6);

- роздільна здатність екрану не менше 720p (1280x720 пікселів);
- підтримка 3G або більш швидкісного мобільного інтернет-з'єднання.

Наведені технічні характеристики є мінімальними вимогами для забезпечення стабільної роботи вебдодатку. Перевищення цих вимог дозволить користувачам отримати кращий досвід використання програмного продукту.

1.5.4. Вимоги до інформаційної та програмної сумісності

Для нормального функціонування веборієнтованого додатку з продажу вінтажних велосипедів необхідно, щоб обчислювальна машина, на якій буде працювати система, відповідала наступним вимогам:

- операційна система: Unix, Linux, Microsoft Windows XP/7/8/10;
- веббраузер: Microsoft Internet Explorer, Mozilla Firefox, Opera, Google Chrome;

Веборієнтована інформаційна система має бути реалізована з використанням:

- мови програмування: PHP;
- мови розмітки: HTML5, CSS3;
- бібліотеки JavaScript: jQuery;
- вебсервер: Apache HTTP Server;
- система управління базами даних: MySQL;

Додаткові вимоги:

- підтримка актуальних версій використаних технологій та можливість оновлення до нових версій;
- забезпечення безпечного з'єднання з Інтернетом для коректної роботи вебдодатку;
- наявність достатніх обчислювальних ресурсів (процесор, оперативна пам'ять, дисковий простір) для забезпечення належної продуктивності системи.

Дані вимоги забезпечать сумісність веборієнтованого додатку з різними платформами та програмним забезпеченням, а також гарантуватимуть коректну роботу системи під час її експлуатації.

РОЗДІЛ 2

ПРОЄКТУВАННЯ ТА РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1. Функціональне призначення системи

Функціональні можливості вебдодатку “Bicycle” визначаються наступними ключовими аспектами:

- ведення бази даних товарів (вінтажних велосипедів) із детальною інформацією про кожен товар: назва, опис, зображення, ціна тощо;
- створення системи обліку та аутентифікації користувачів із захищеним зберіганням персональних даних, історії замовлень;
- впровадження онлайн-кошика для вибору товарів, встановлення кількості, застосування знижок, розрахунку підсумкової вартості замовлення;
- інтеграція різних варіантів оплати та доставки замовлень із автоматичним розрахунком додаткових витрат;
- генерація та надсилання електронних підтверджень разом із деталями кожного оформленого замовлення;
- підтримка комунікації з клієнтами через форму підписки на розсилку новин та акційних пропозицій.

Експлуатаційне призначення:

- створення повноцінної онлайн-платформи для продажу вінтажних велосипедів без необхідності утримання фізичного магазину;
- розширення географії продажів та залучення нових клієнтів за рахунок цілодобової онлайн-присутності;
- автоматизація процесів приймання, обробки та виконання замовлень для підвищення ефективності та скорочення помилок;
- забезпечення зручного пошуку, перегляду товарів та оформлення замовлень для покращення користувацького досвіду;

- можливість аналізу та виявлення найпопулярніших моделей, географії замовників тощо для оптимізації асортименту та маркетингової стратегії;
- формування лояльної клієнтської бази та підвищення впізнаваності бренду через якісний дизайн, контент та комунікацію.

2.2. Опис застосованих математичних методів

Під час розробки веборієнтованого додатку для продажу вінтажних велосипедів використовувались арифметичні операції та логічні перетворення для обчислення вартості замовлення та розрахунку вартості доставки.

Обчислення загальної вартості замовлення:

Для розрахунку загальної вартості замовлення використовується формула (2.1):

$$T = \sum(P_i \times Q_i) - D + S \quad (2.1)$$

де T – загальна вартість замовлення,

P_i – ціна i -го товару,

Q_i – кількість i -го товару,

D – сума всіх знижок,

S – вартість доставки.

Розрахунок вартості доставки:

Вартість доставки обчислюється за формулою (2.2):

$$S = B + A \quad (2.2)$$

де S – кінцева вартість доставки,

B – базова вартість доставки,

A – додаткова вартість, яка залежить від обраного методу доставки.

При цьому A визначається наступним чином:

Якщо метод = DPD, то $A = -15 \text{ €}$

Якщо метод = DHL, то A = 0 €

Якщо метод = DHL Express, то A = 67 €

2.3. Опис використаних технологій та мов програмування

2.3.1. Роль HTML та CSS у формуванні зовнішнього вигляду вебсторінок

У сфері веброзробки HTML (HyperText Markup Language) є основною технологією, що використовується для створення структури вебсторінок у переважній більшості випадків [19]. HTML задає базову структуру контенту, визначаючи семантичну організацію документа. Проте важливо розуміти деякі специфічні аспекти цієї мови [20].

HTML є декларативною мовою, що означає, що вона описує структуру документа без визначення послідовності його побудови [11]. Це може створювати певні обмеження у динамічному управлінні контентом. На рисунку 2.1. показано, як елементи HTML формують ієрархічну структуру, де кожен елемент може містити інші елементи або текстовий вміст.

```
<div>  
  <h1>Заголовок</h1>  
  <p>Параграф з <strong>виділеним</strong> текстом.</p>  
</div>
```

Рис. 2.1. Приклад ієрархічної структури HTML

HTML не забезпечує вбудованих засобів для стилізації та оформлення контенту, що обмежує можливості візуального представлення інформації (рис. 2.2).

```
<table>
  <tr>
    <td>Клітинка 1</td>
    <td>Клітинка 2</td>
  </tr>
</table>
```

Рис. 2.2. Приклад таблиці HTML без стилізації

Для вирішення питань візуального оформлення та створення адаптивних інтерфейсів використовується CSS (Cascading Style Sheets). CSS є мовою стилів, що дозволяє визначати зовнішній вигляд HTML-елементів [21]. Згідно з даними W3Techs, CSS застосовується на більшості вебсайтів.

CSS надає потужні інструменти для стилізації, але також має свої особливості [12]. Наприклад, каскадність CSS може призводити до непередбачуваних результатів при неправильному використанні специфічності селекторів (рис. 2.8).

```
.button {
  background-color: blue;
}

#submit-button {
  background-color: green;
}
```

Рис. 2.5. Приклад конфлікту специфічності в CSS

2.3.2. Аналіз мов програмування JavaScript та PHP у веброзробці

Для розробки backend-частини вебдодатку було використано мови програмування JavaScript та PHP. Ці мови часто використовуються разом для створення повноцінних вебдодатків, де JavaScript відповідає за інтерактивність

та динаміку на стороні клієнта, а PHP обробляє дані та виконує складні операції на стороні сервера.

JavaScript є невід'ємною частиною сучасної веброзробки, надаючи вебсторінкам інтерактивності та динамічності [7]. Однією з ключових особливостей JavaScript є його асинхронна природа, що дозволяє створювати більш плавні та чуйні інтерфейси. Для структурування асинхронного коду JavaScript пропонує різні механізми, такі як callback-функції, проміси (promises) та async/await.

У розробленому вебдодатку JavaScript виконує ряд важливих функцій:

- на головній сторінці реалізує слайдери з фотографіями велосипедів за допомогою бібліотеки Bootstrap Carousel та фільтри глобального пошуку;
- при переході до каталогу товарів забезпечує завантаження лише частини сторінки без повного перезавантаження;
- на сторінці детального перегляду велосипеда забезпечує інтерактивну галерею фото та можливість додавати товари у кошик;
- у кошику дозволяє оновлювати кількість чи видаляти товари;
- на сторінці входу в систему виконує валідацію даних форми та генерує хеш пароля;
- під час реєстрації нового користувача валідує поля форми та перевіряє унікальність електронної пошти в режимі реального часу.

PHP, у свою чергу, виконується на стороні сервера і має повний доступ до його ресурсів. Области застосування PHP у веброзробці включають системи керування контентом (CMS), інтернет-магазини, соціальні мережі, форуми та блоги [8].

У розробленому вебдодатку PHP виконує наступні функції:

- отримує дані через AJAX або REST API запити та записує їх у базу даних;
- при оформленні замовлення збирає інформацію, вносить замовлення в базу та генерує підтвердження;

- зберігає дані входу користувача, включаючи дату, час, роздільну здатність екрану та операційну систему;
- обробляє дані реєстрації, зберігає їх у базі даних та відправляє електронний лист з тимчасовим паролем;
- оновлює стан кошика в базі даних і відправляє відповідь на клієнтську сторону;
- при оформленні замовлення зберігає деталі в базі, розраховує знижки та вартість доставки, генерує номер замовлення.

Поєднання JavaScript для клієнтської взаємодії та PHP для серверної обробки дає значні переваги, створюючи повнофункціональний, безпечний вебдодаток для успішної електронної комерції вінтажними велосипедами. Такий підхід використовується багатьма популярними вебсайтами, забезпечуючи потужну функціональність та простоту у використанні для мільйонів користувачів щодня.

2.3.3. Обґрунтування доцільності обраного фреймворку та бібліотеки

Щодо фреймворків та бібліотек у сфері веброзробки, вони відіграють ключову роль, значно прискорюючи процес створення вебдодатків та підвищуючи їхню якість. Згідно з дослідженням Stack Overflow за 2023 рік, 77,7% професійних розробників використовують фреймворки або бібліотеки у своїй роботі [9]. Однак, важливо розуміти особливості та обмеження кожного інструменту.

На рисунку 2.7 представлено топ 6 вебфреймворків і технологій за популярністю серед розробників. Лідером є Node.js, який використовують 42,65% опитаних, за ним слідує React з 40,58%. Ці дані підкреслюють важливість вибору правильного інструменту для розробки, враховуючи як популярність технології, так і її відповідність конкретним вимогам проекту. Цікаво відзначити, що незважаючи на свій вік, jQuery все ще займає третє місце з 21,98% користувачів, що свідчить про його стійку позицію в екосистемі веброзробки.

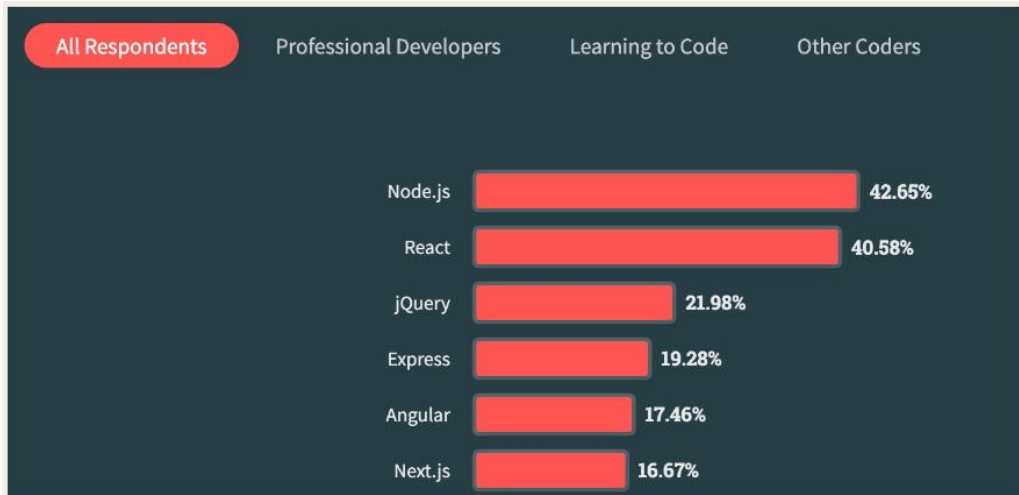


Рис. 2.7. Топ 6 вебфреймворків і технологій

Bootstrap, найпоширеніший фреймворк CSS, використовується майже третиною всіх сайтів в інтернеті. Його популярність зумовлена великою кількістю готових компонентів та інструментів, що значно спрощують та пришвидшують створення адаптивних дизайнів. Однак, надмірне використання стандартних елементів Bootstrap може призвести до того, що сайти виглядатимуть однотипно [10].

Як показано на рисунку 2.8, Bootstrap дійсно домінує серед JavaScript бібліотек з показником абсолютного використання 19,9% та часткою ринку 24,5%. Це підтверджує його статус як найпоширенішого CSS фреймворку. У порівнянні з ним, React, який є популярною бібліотекою для створення користувацьких інтерфейсів, має значно менші показники: 4,2% абсолютного використання та 5,2% частки ринку. Ці дані підкреслюють, наскільки широко Bootstrap застосовується в веброботці, що пояснює його вплив на загальний вигляд багатьох сайтів в інтернеті. Проте, розробникам слід пам'ятати про баланс між зручністю використання готових рішень та створенням унікального дизайну для своїх проєктів.

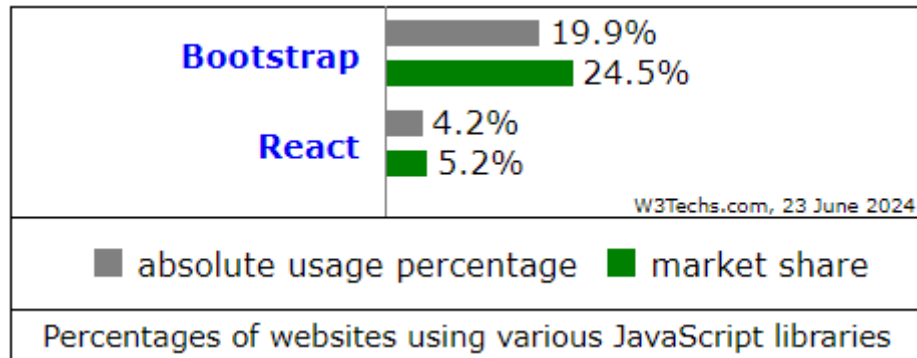


Рис. 2.8. Порівняння використання JavaScript бібліотек Bootstrap та React

jQuery, хоча й втратила частину своєї популярності в останні роки, залишається широко використовуваною бібліотекою JavaScript, яка присутня на більшості вебсайтів. Її основні переваги полягають у спрощенні роботи з DOM (Document Object Model) та AJAX-запитами, що робить розробку інтерактивних вебсторінок швидшою та зручнішою [22]. Проте, надмірне використання jQuery може призвести до надмірного застосування імперативного стилю програмування, коли код описує кожен крок виконання завдання, замість того, щоб зосередитися на кінцевому результаті. Це може ускладнити читабельність та підтримку коду, особливо у великих проєктах [23].

2.3.4. Обґрунтування доцільності обраної СКБД

В якості потужної системи керування реляційними базами даних (СКБД) баз даних було використано MySQL, яка є основою майже половини всіх вебсайтів, що використовують бази даних. У контексті розробки вебдодатку для продажу вінтажних велосипедів, MySQL може забезпечити надійне та ефективне зберігання та управління всіма необхідними даними [24]:

- можливість аналізу та виявлення найпопулярніших моделей, географії замовників тощо для оптимізації асортименту та маркетингової стратегії;
- інформація про користувачів, як наприклад особисті дані, контактна інформація, історія покупок, відгуки тощо;

- каталог товарів, а саме детальна інформація про кожен велосипед (модель фотографії, ціна), наявність на складі;
- історія замовлень, як наприклад дані про оформлені замовлення, їх статус, інформація про доставку та оплату.

Також MySQL пропонує широкий спектр можливостей для роботи з даними, включаючи:

- історія замовлень. Дані про оформлені замовлення, їх статус, інформація про доставку та оплату;
- структуроване зберігання. Дані організовані у вигляді таблиць, що дозволяє легко їх шукати, фільтрувати та сортувати;
- підтримка транзакцій. Забезпечує цілісність даних, гарантуючи, що всі зміни будуть виконані або скасовані повністю;
- мова запитів SQL. Потужний інструмент для вибірки, оновлення та видалення даних;
- висока продуктивність. MySQL оптимізована для роботи з великими обсягами даних та забезпечує швидкий доступ до них.

Для взаємодії з MySQL та управління нею розробники часто використовують phpMyAdmin – популярний вебінтерфейс, який використовують відомі інструменти адміністрування баз даних. phpMyAdmin надає зручний візуальний інтерфейс для створення та редагування таблиць бази даних, виконання SQL-запитів та інших операцій [13].

Вибір phpMyAdmin для управління базою даних у проєкті веборієнтованого додатку обумовлений низкою ключових факторів. Інтуїтивно зрозумілий вебінтерфейс phpMyAdmin значно спрощує процес управління базою даних MySQL, що є критичним для швидкого редагування даних, замовлення та користувачів. Широка функціональність інструменту дозволяє ефективно виконувати весь спектр необхідних операцій з базою даних. Наприклад, за допомогою phpMyAdmin ми можемо легко створювати складні запити для аналізу продажів, такі як виявлення найпопулярніших моделей велосипедів за

певний період часу. Крім того, інструмент дозволяє нам швидко вносити зміни в структуру бази даних, наприклад, додавати нові поля для опису велосипедів, без необхідності написання складних SQL-скриптів.

Враховуючи специфіку мого проєкту, де ключову роль відіграє гнучкість у керуванні даними про унікальні товари та можливість швидко адаптувати структуру бази даних, phpMyAdmin виявився оптимальним вибором. Цей інструмент забезпечує необхідну гнучкість та зручність використання, що сприяє створенню надійного та ефективного вебдодатку.

2.4. Опис структури системи та алгоритмів її функціонування

2.4.1. Алгоритми та структура проєкту

Веборієнтований додаток "Bicycle" для онлайн-магазину з продажу вінтажних велосипедів розроблено з використанням архітектури Model-View-Controller (MVC) [25]. Ця архітектура забезпечує чітке розділення логіки, представлення даних та управління, що полегшує розробку, тестування та подальшу підтримку системи. Для кращого розуміння структури веборієнтованого додатку "Bicycle" було застосовано метод декомпозиції. Цей метод дозволяє розбити систему на менші, більш керовані компоненти та візуалізувати їх взаємозв'язки. Нижче наведено таблицю декомпозиції системи:

Декомпозиція проекту

Інтернет - магазин	Про магазин			
	Продукти	Каталог	Сторінка продукту та його опис	фото
				назва
				ціна
				купити
				Видалити з кошику
				Пошук продуктів
	Профіль користувача	Історія замовлень	Номер замовлення	
			Опис продуктів	
			Замовити знову	
			Ціна замовлення	
	Кошик	Список товарів	Опис товару	Фото
				Назва
				Кількість
				Ціна
Видалити				
Добавити замовлення				

	Адреса рахунку	Ім'я			
		Прізвище			
		Логін			
		Пошта			
		Адреса			
	Доставка				
	Країна				
	Місто				
	Поштовий індекс				
	Продукти	Назва			
		Ціна			
	Загальна ціна				
Оплата	Ім'я				
	Номер картки				
	Дата				
	CV-код				
Замовити	Підтвердження через електронну пошту				
Авторизація	Логін / Пошта				
	Пароль				
Вхід в систему	Реєстрація	Клієнт	Логін	Валідація даних	
			Ім'я		
			Прізвище		
Сервер	Приєм запиту	Валідація даних	Зберігання даних користувача		

					дата і час	
				Створення користувача	роздільна здатність екрану (1280×768 або 3840×2500...)	
					операційна система Mac OS, Windows, Android, Linux	
				Відповідь	Результат створення	Надсилання повідомлення на пошту
Пошук	Система пошуку та відображення					

Вебдодаток "Bicycle" представляє собою вебсайт для продажу вінтажних велосипедів, що має наступні основні сторінки:

- головна сторінка;
- реєстрація;
- авторизація;
- особистий кабінет;
- каталог товарів;
- кошик покупок;
- сторінка оформлення замовлення.

Всі ці сторінки мають власні компоненти, які пов'язані між собою та утворюють функціонал додатку.

З головної сторінки є можливість перейти до авторизації чи реєстрації користувача. Після введення даних та успішної авторизації користувач отримує доступ до повного функціоналу сайту, включаючи перегляд каталогу та доступ до особистого кабінету.

каталозі товарів користувач може переглядати доступні вінтажні велосипеди, додавати їх до кошика та оформлювати замовлення. Система передбачає функціонал фільтрації та пошуку товарів для зручності користувачів.

Особистий кабінет містить інформацію про користувача, історію замовлень та можливість редагування особистих даних. Тут користувач може переглядати статус своїх поточних замовлень та повторювати попередні покупки в один клік.

Кошик покупок дозволяє користувачу переглядати обрані товари, змінювати їх кількість або видаляти перед оформленням замовлення.

На сторінці оформлення замовлення користувач вводить необхідну інформацію для доставки, обирає спосіб оплати та підтверджує замовлення.

Система автоматично розраховує вартість доставки.

Під час виконання транзакцій деякі операції можуть затримуватись на час обробки платежу та підтвердження замовлення. Ці дії виконуються для забезпечення безпечних покупок та надійної роботи сервісу.

Система також передбачає відправку електронних листів для підтвердження реєстрації, оформлення замовлення та інших важливих подій.

Логічна структура вебдодатку забезпечує зручну навігацію між різними сторінками та ефективну взаємодію користувача з системою на всіх етапах процесу покупки вінтажних велосипедів.

Логічна структура представлена на рис. 2.2.

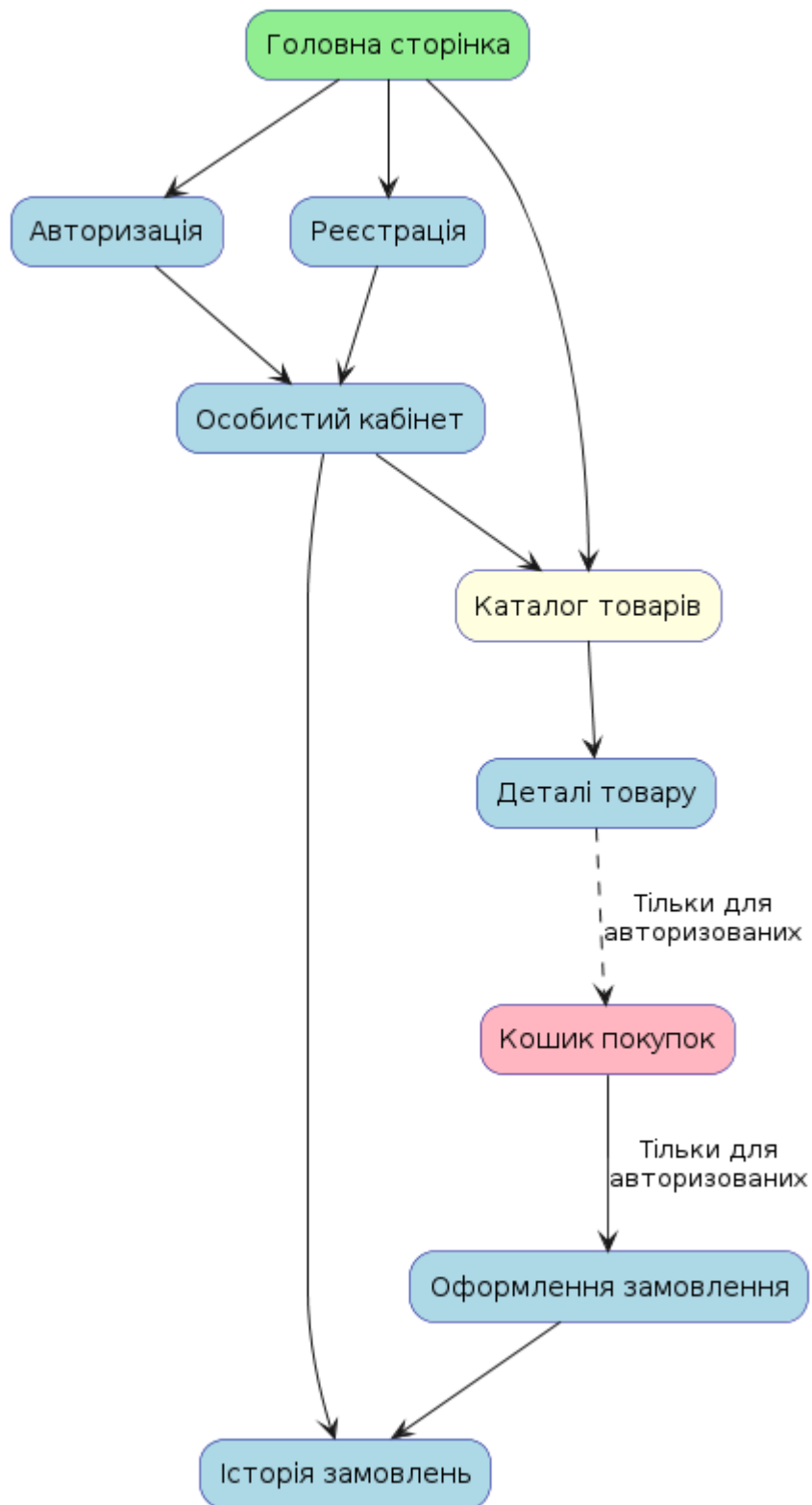


Рис. 2.9. Діаграма логічної структури додатку

У вебдодатку 'Bicycle' реалізовано чітке розмежування ролей між відвідувачами та зареєстрованими користувачами. Відвідувачі мають можливість переглядати каталог товарів, але функціонал покупки доступний

лише авторизованим користувачам. Це забезпечує безпеку транзакцій та дозволяє підтримувати якість обслуговування.

Зареєстровані користувачі отримують доступ до розширеного функціоналу, включаючи можливість додавати товари до кошика, оформлювати замовлення та переглядати історію своїх покупок. Така структура дозволяє ефективно керувати процесом продажу, забезпечуючи при цьому зручність для всіх категорій відвідувачів сайту.

Діаграма діяльності (рис. 2.10) демонструє основні можливості взаємодії користувачів з вебдодатком, включаючи процеси реєстрації, авторизації, перегляду каталогу, додавання товарів до кошика та оформлення замовлення. Ця діаграма наочно показує, як різні компоненти системи взаємодіють між собою, забезпечуючи цілісний та ефективний процес покупки вінтажних велосипедів.



Рис. 2.10. Діаграма діяльності користувачів

2.4.2. Файлова структура проекту

Файлова ієрархія:

1. assets/

- Містить статичні ресурси проекту:
- fonts/: Шрифти, що використовуються в додатку.
- images/: Зображення для вебсторінок.
- json/: JSON файли для зберігання структурованих даних.
- php/: PHP скрипти для серверної логіки.
- js/: JavaScript файли, розділені на модулі та сторінки.

2. main/

- Основні файли сторінок додатку:
- index.html: Головна сторінка сайту.
- script.mjs: JavaScript код для головної сторінки.
- style.scss: Стили для головної сторінки.
- Підкаталоги для різних секцій сайту (overview, reassign, teststatus).

3. scss/modules/

- Модульні SCSS файли для різних компонентів:
- catalog/: Стили для каталогу товарів.
- login/: Стили для сторінки входу.
- main/: Основні стилі сайту.
- Загальні SCSS файли (_fonts.scss, _functions.scss, _mixins.scss, _reset.scss).

4. js/modules/

JavaScript модулі:

- functions/: Допоміжні функції.
- login/: Логіка авторизації.
- webpack/: Конфігурація Webpack.

5. pages/

- головна сторінка;
- Компоненти окремих сторінок:
- `catalog/`: Сторінка каталогу товарів.
- `checkout/`: Сторінка оформлення замовлення.
- `login/`: Сторінка входу в систему.

6. Кореневі файли

- `.eslintrc`: Конфігурація ESLint для забезпечення якості коду.
- `.gitignore`: Список файлів та директорій, які Git повинен ігнорувати.

Така структура проекту забезпечує чітку організацію коду та ресурсів, що полегшує розробку та підтримку вебдодатку. Розділення на компоненти (сторінки, модулі) та типи файлів (HTML, JavaScript, SCSS) відповідає сучасним практикам веброзробки.

Загальна структура поточного проекту візуально представлена на рис. 2.11.

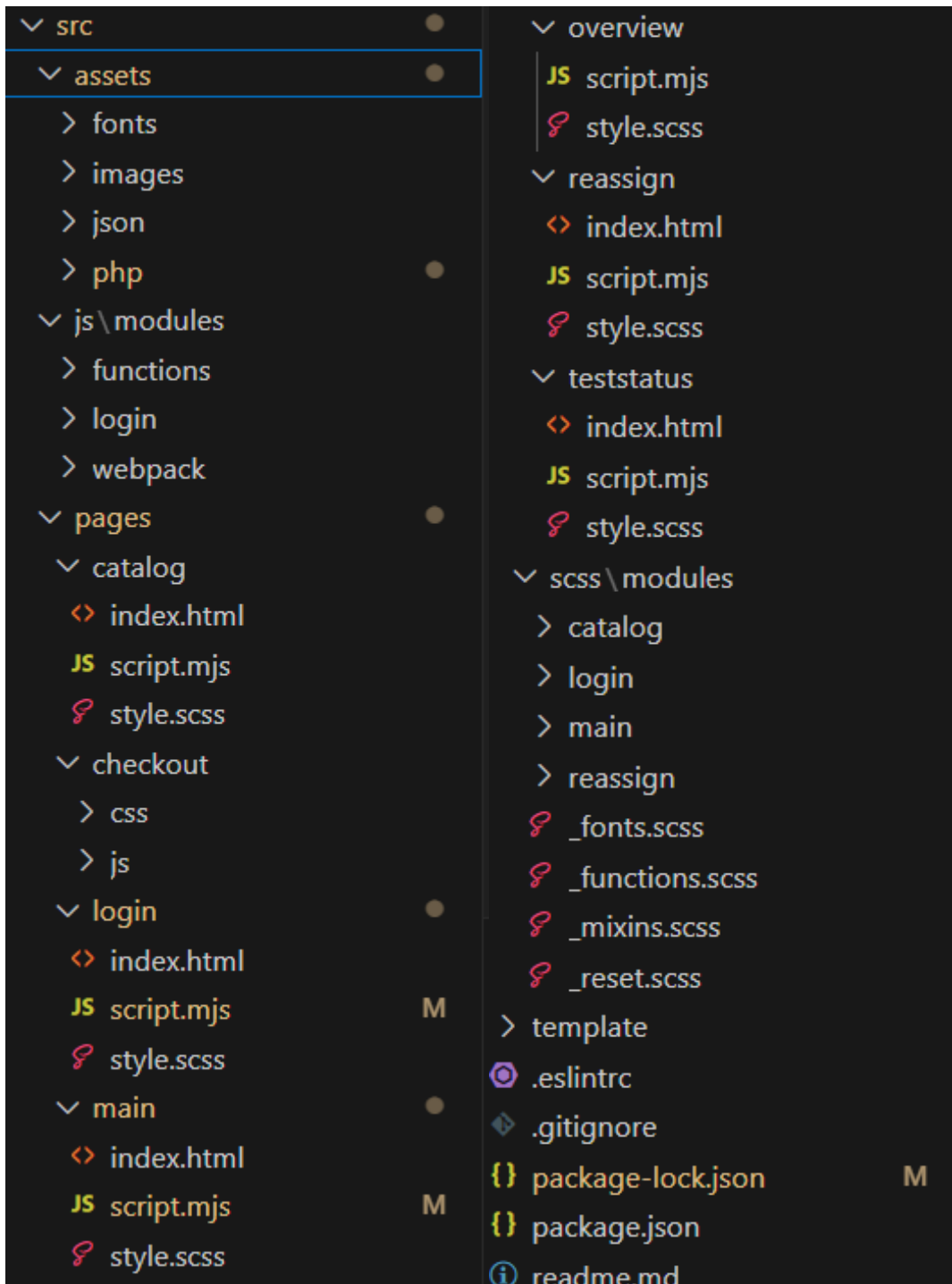


Рис. 2.11. Файлова ієрархія програми

2.4.3. Структура бази даних

База даних, що визначена для цієї кваліфікаційної роботи, є реляційною. Було визначено основні логічні сутності та побудовано фізичні структури, а

також контекст бази даних для взаємодії з таблицями в застосунку. Основними сутностями є:

- користувачі. Дана сутність представляє зареєстрованих користувачів системи. Ця інформація використовується для авторизації та персоналізації досвіду користувача на сайті;
- товари. Містить інформацію про усі наявні вінтажні велосипеди, їх назву, опис, ціну, кількість на складі та зображення. Ця інформація відображується на сторінці каталогу товарів;
- замовлення. Зберігає усю інформацію про виконані замовлення, включаючи обраний метод доставки, загальну суму та статус замовлення;
- кошик покупок. Являє собою тимчасове сховище товарів, які користувач додав для потенційної покупки;
- історія входів. Представляє дані про кожен вхід користувача в систему, включаючи дату, час, роздільну здатність екрану та операційну систему.

Таким чином, логічна структура бази даних інтернет-магазину вінтажних велосипедів буде складатися та мати зв'язки, що вказані на рис. 2.12.

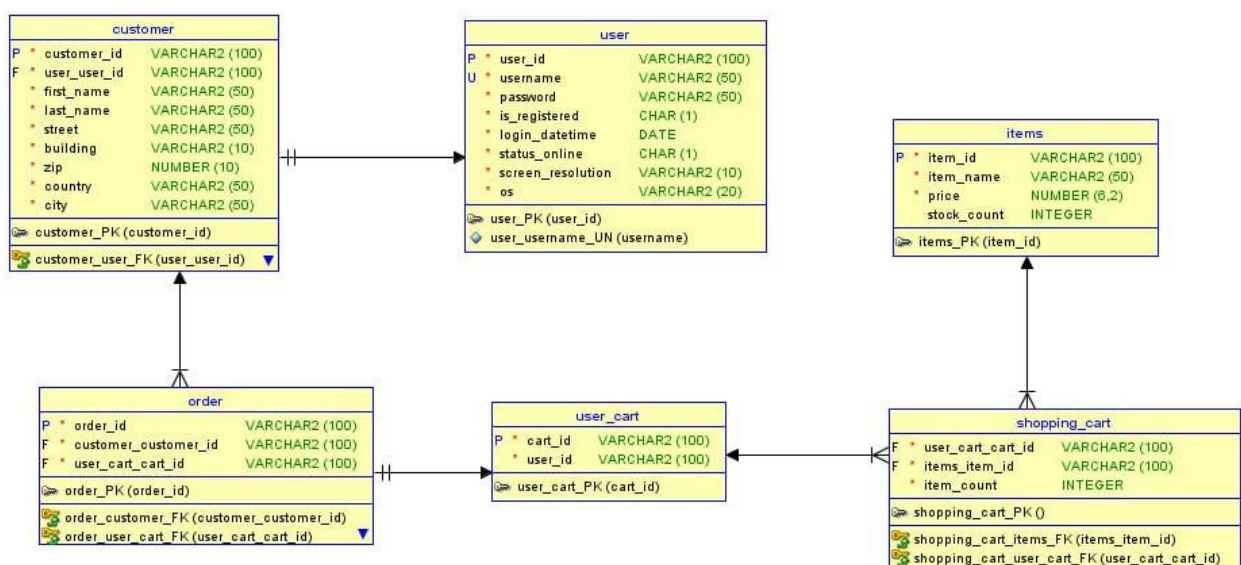


Рис. 2.12. Логічна модель даних UML

Детальний опис призначення атрибутів вказано в таблиці 2.2.

Таблиця 2.2

Атрибути і первинні ключі сутностей

СУТНІСТЬ	ПЕРВИННИЙ КЛЮЧ	АТРИБУТИ
Order	Унікальний ключ номер замовлення	Унікальний ключ номер замовлення Унікальний ключ номер клієнта Унікальний ключ номер кошика
Customer	Унікальний ключ номер клієнта	Унікальний ключ номер клієнта Унікальний ключ номер користувача Ім'я Прізвище Вулиця Будинок Поштовий індекс Країна Місто
Items	Унікальний ключ номер товару	Унікальний ключ номер товару Назва товару Ціна Кількість на складі

User_cart	Унікальний ключ номер кошика	Унікальний ключ номер кошика Унікальний ключ номер користувача
Shopping_cart	Унікальний ключ номер кошика, унікальний ключ номер товару	Унікальний ключ номер кошика Унікальний ключ номер товару Кількість товару
User	Унікальний ключ номер користувача	Унікальний ключ номер користувача Ім'я користувача Пароль Статус реєстрації Дата та час входу Статус онлайн Розширення екрану Операційна система

Таким чином, опис зв'язків таблиць буде наступним:

- order та customer. Кожне замовлення пов'язане з одним клієнтом, але один клієнт може мати багато замовлень. Тип даного зв'язку – багато-до-одного;
- user_cart та user. Кожен кошик користувача пов'язаний з одним користувачем, але один користувач може мати лише один кошик. Зв'язок один-до-одного;

- customer та user. Кожен клієнт пов'язаний з одним користувачем, і кожен користувач може бути пов'язаний лише з одним клієнтом. Отже, зв'язок один-до-одного;
- shopping_cart та items. Кожен запис у shopping_cart пов'язаний з одним товаром, але один товар може бути присутнім у багатьох записах shopping_cart. Зв'язок багато-до-одного.
- order та user_cart. Кожне замовлення пов'язане з одним кошиком користувача, і кожен кошик може бути пов'язаний лише з одним замовленням. Зв'язок один-до-одного;
- shopping_cart та user_cart. Кожен запис у shopping_cart пов'язаний з одним кошиком користувача, але один кошик може містити багато записів у shopping_cart. Зв'язок багато-до-одного;

Для детального представлення опису фізичної моделі використано таблиці (табл. 2.3 – 2.8).

Таблиця 2.3

Таблиця «Замовлення»

Orders (Замовлення)				1
№ п/п	Найменування стовбців	Примітка	Тип	Розмір
1.	OrderId	Унікальний ключ номер замовлення	integer	
2.	CustomerId	Унікальний ключ номер клієнта	integer	
3.	UserCartId	Унікальний ключ номер кошика	integer	

Таблиця «Користувач»

User (Користувач)					2
№ п/п	Найменування стовбців	Примітка	Тип	Розмір	
1.	UserId	Унікальний ключ номер користувача	integer		
2.	Username	Ім'я користувача	varchar	50	
3.	Password	Пароль	varchar	255	
4.	IsRegistered	Статус реєстрації	boolean		
5.	LoginDatetime	Дата та час входу	datetime		
6.	StatusOnline	Статус онлайн	boolean		
7.	ScreenResolution	Розширення екрану	varchar	20	
8.	OS	Операційна система	varchar	50	

Таблиця «Клієнт»

Customer (Клієнт)					3
№ п/п	Найменування стовбців	Примітка	Тип	Розмір	
1.	CustomerId	Унікальний ключ номер ролі	integer		
2.	UserUserId	Унікальний ключ номер користувача	integer		
3.	FirstName	Ім'я	varchar	50	
4.	LastName	Прізвище	varchar	50	
5.	Street	Вулиця	varchar	100	

Продовж. табл. 2.5

6.	Building	Будинок	varchar	10
7.	Zip	Поштовий індекс	varchar	10
8.	Country	Країна	varchar	50
9.	City	Місто	varchar	50

Таблиця.2.6

Таблиця «Товари»

Products (Продукти)				4
№ п/п	Найменування стовбців	Примітка	Тип	Розмір
1.	ItemId	Унікальний ключ номер продукту	integer	
2.	ItemName	Назва продукту	varchar	100
3.	StockCount	Кількість на складі	decimal	(10,2)
4.	Price	Ціна продукту	real	

Таблиця 2.7

Таблиця «Кошик користувача»

User_cart (Кошик користувача)				4
№ п/п	Найменування стовбців	Примітка	Тип	Розмір
1.	CartId	Унікальний ключ номер кошика	integer	
2.	UserId	Унікальний ключ номер користувача	integer	

Таблиця 2.8

Таблиця «Список покупок»

Shopping_cart (Список покупок)				4
№ п/п	Найменування стовбців	Примітка	Тип	Розмір
1.	UserCartCartId	Унікальний ключ номер кошика	integer	
2.	ItemsItemId	Унікальний ключ номер товару	integer	
3.	ItemCount	Кількість товару	integer	

Фізичну структуру бази даних вебдодатку "Bicycle" зображено на рис. 2.13. Ця схема детально відображає організацію даних на рівні таблиць, включаючи назви стовпців та типи даних. На діаграмі також представлені первинні та зовнішні ключі, що демонструють зв'язки між таблицями. Такий рівень деталізації дозволяє отримати повне уявлення про структуру бази даних та її реалізацію на фізичному рівні. Ця схема є ключовим елементом для розуміння того, як дані про користувачів, товари, замовлення та кошики покупок зберігаються та взаємодіють між собою в системі онлайн-магазину вінтажних велосипедів.

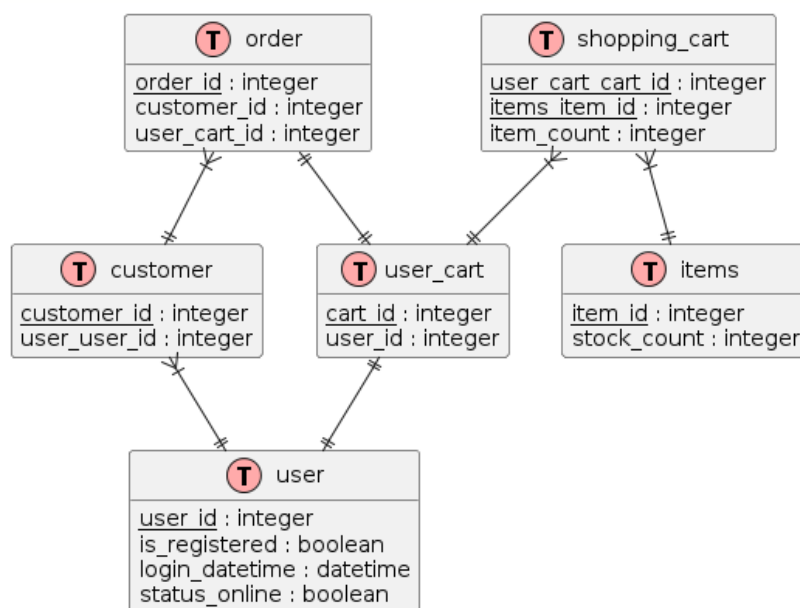


Рис. 2.12. Фізична структура бази даних

2.5. Обґрунтування та організація вхідних та вихідних даних програми

Вебдодаток "Bicycle" обробляє різноманітні вхідні та вихідні дані відповідно до вимог, описаних у розділі 1.5. Взаємодія користувачів з інтерфейсом ініціює обробку даних на сервері. Після валідації (рис. 2.13) дані зберігаються в базі даних (рис. 2.14), а інтерфейс оновлюється. При виникненні помилок користувачі отримують відповідні повідомлення.

WELCOME TO OUR SHOP!

Log in to your account

Email
Kondrat.A.D@nmu.one

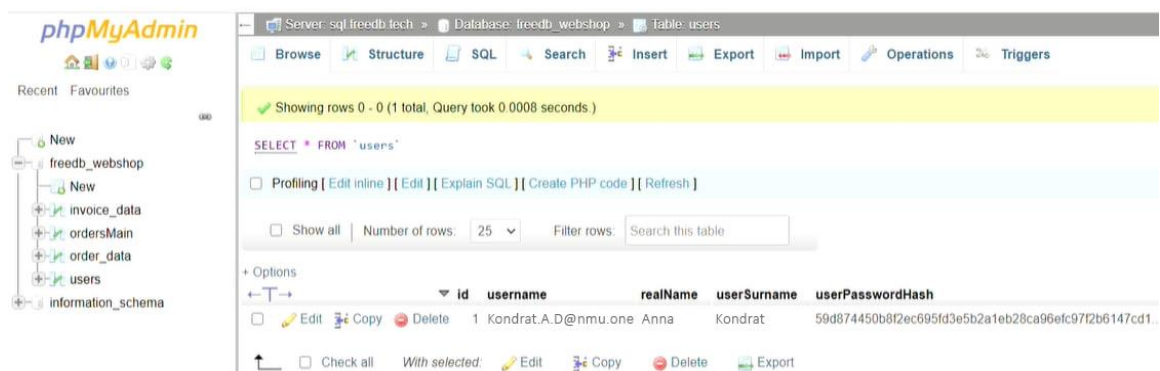
Password
.....

Password should contain at least 9 characters

Continue with Email

Don't have an account? [Register](#)

Рис. 2.13. Форма входу користувача в вебдодатку "Bicycle" з повідомленням про помилку валідації пароля



Server: sql.freedb.tech > Database: freedb_webshop > Table: users

Showing rows 0 - 0 (1 total, Query took 0.0008 seconds)

```
SELECT * FROM `users`
```

id	username	realName	userSurname	userPasswordHash
1	Kondrat.A.D@nmu.one	Anna	Kondrat	59d874450b8f2ec695fd3e5b2a1eb28ca96efc97f2b6147cd1..

Рис. 2.14. Таблиця "users" в базі даних MySQL вебдодатку "Bicycle"

Вхідні дані включають:

- реєстраційні дані користувачів (email як username, пароль). Ці дані збираються при реєстрації нового користувача та використовуються для створення облікового запису в системі;

- інформацію для входу (логін, пароль). Ці дані використовуються для автентифікації користувача при вході в систему;

- дані про товари (вінтажні велосипеди). Ці дані включають інформацію про кожен товар (назву, опис, ціну, зображення тощо) та зберігаються в базі даних;

- інформацію для оформлення замовлень. Ці дані включають адресу доставки, спосіб оплати та інші деталі, необхідні для обробки замовлення.

У системі реалізовано форми для оптимізації процесів реєстрації, додавання товарів до кошика та оформлення замовлень. Ці форми містять функції редагування та вибору опцій для зменшення ризику помилок. Додаток працює з текстовими, числовими даними та датами.

Рівень доступу залежить від типу користувача. Незареєстровані можуть переглядати каталог, а зареєстровані - здійснювати покупки та керувати профілем.

Вихідні дані формуються у вигляді HTML-сторінок і включають:

- каталог товарів (мінімум 9 товарів). Цей розділ відображає інформацію про товари (вінтажні велосипеди), отриману з бази даних MySQL/ Кожен товар представлений зображенням, назвою, описом, ціною та кнопкою "Додати до кошика";

- кошик покупок з можливістю редагування кількості. Ця сторінка дозволяє користувачеві переглянути вибрані товари, змінити їх кількість або видалити. Дані про товари в кошику зберігаються в сесії користувача;

- особистий кабінет користувача. Цей розділ дозволяє зареєстрованим користувачам переглядати та редагувати свої дані, історію замовлень та керувати підписками. Дані профілю користувача зберігаються в базі даних;

- підтвердження замовлень по email. Після успішного оформлення замовлення користувачеві надсилається електронний лист з підтвердженням та деталями замовлення. Цей процес відбувається на сервері після обробки даних замовлення.

Розглянемо основні процеси взаємодії користувача з додатком.

Процес реєстрації користувача:

- користувач заповнює форму реєстрації, вводячи свої дані (email, пароль);
- дані проходять валідацію на клієнті (за допомогою JavaScript) та сервері (за допомогою PHP);
- при успішній валідації дані зберігаються в базі даних MySQL;
- створюється профіль користувача, і він автоматично авторизується в системі;

Процес додавання товару до кошика:

- користувач вибирає товар з каталогу та натискає кнопку "Додати до кошика";
- дані про товар (ідентифікатор, кількість) відправляються на сервер;
- перевіряється наявність товару на складі (за допомогою запиту до бази даних);
- якщо товар є в наявності, він додається до кошика покупок користувача (дані зберігаються в сесії);
- оновлюється вміст кошика на сторінці.

Така організація даних забезпечує цілісність інформації, зручність користування вебдодатком "Bicycle" та ефективну взаємодію між користувачем і системою.

2.6. Опис розробленої системи

2.6.1. Використані технічні засоби

Для розробки та проведення тестування додатку використовувалися технічні засоби з наступними характеристиками:

- процесор: Intel(R) Core(TM) i7-10750H CPU;
- оперативна пам'ять: 16 ГБ;
- диск SSD: Kingston A2000 NVMe PCIe SSD 1TB;
- відеокарта: NVIDIA GeForce RTX 3060 6 ГБ;
- операційна система: 64-розрядна Windows 11;
- комп'ютерна миша;
- клавіатура.

2.6.2. Використані програмні засоби

- Visual Studio Code: Основне середовище розробки для написання коду.
- Figma: Для створення дизайну інтерфейсу та прототипування.
- XAMPP: Для створення локального середовища розробки з Apache та MySQL.
- phpMyAdmin: Для управління базою даних через вебінтерфейс.

Для забезпечення кросбраузерної сумісності, вебдодаток тестувався у різних браузерах, включаючи Google Chrome, Mozilla Firefox, Microsoft Edge.

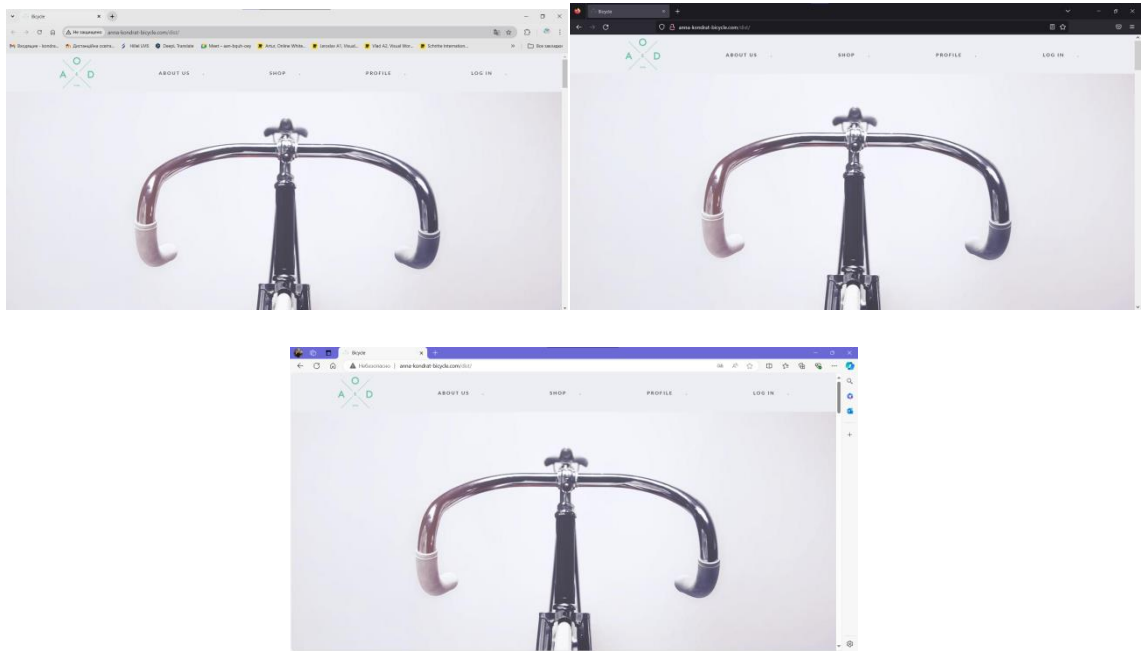


Рис. 2.13. Запуск «Vicycle» в найпопулярніших браузерях

2.6.3. Виклик та завантаження програми

Для успішного запуску вебдодатку "Vicycle" та підключення до бази даних необхідно виконати наступні кроки:

Підготовка сервера:

- Переконайтеся в наявності вебсервера (наприклад, Apache) для хостингу PHP-додатку.
- Забезпечити достатньо ресурсів для обробки очікуваного навантаження.

Встановлення необхідного програмного забезпечення:

- Встановити PHP (версія 7.0 або вище) на сервер.
- Встановити MySQL для управління базою даних.

Налаштування бази даних:

- Створити нову базу даних MySQL для проекту "Vicycle".
- Імпортувати структуру бази даних, використовуючи надані SQL-скрипти.

Розгортання вебдодатку:

- Завантажити файли проекту в кореневу директорію вебсервера.
- Налаштувати файл конфігурації (наприклад, `config.php`) для встановлення параметрів підключення до MySQL.

Запуск і тестування:

- Запустити вебсервер Apache.
- Відкрити веббраузер та перейти за адресою `http://localhost/bicycle` (або відповідною IP-адресою сервера).
- Перевірити функціональність вебдодатку та коректність підключення до бази даних.

Налаштування безпеки:

- Забезпечити HTTPS-з'єднання для захисту даних користувачів.
- Налаштувати файрвол для обмеження доступу до серверу.

Ця послідовність дій забезпечить правильне розгортання та функціонування вебдодатку "Bicycle".

2.6.4. Опис інтерфейсу користувача

Після переходу на веборієнтований додаток "Bicycle" для продажу вінтажних велосипедів, користувач бачить інтуїтивно зрозумілий та адаптивний інтерфейс. Майже всі сторінки системи успадковують єдиний макет, який містить головні компоненти системи, розташовані у шапці сайту.

Шапка сайту (Header) (рис. 2.14) забезпечує користувачів постійним доступом до ключових функцій і навігації протягом усього часу взаємодії з додатком, що позитивно впливає на користувацький досвід.

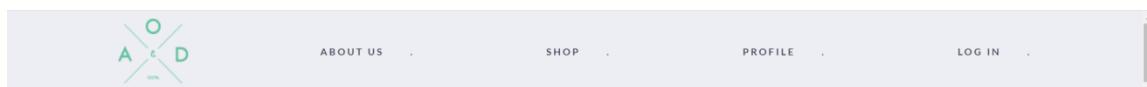


Рис. 2.14. Шапка вебдодатку «Bicycle»

Розглянемо її компоненти детальніше:

– логотип "Bicycle" розташований у лівому верхньому куті шапки. Він не лише забезпечує впізнаваність і формує імідж магазину, але й слугує посиланням на домашню сторінку, надаючи користувачам швидкий доступ до початку сайту.

– основне навігаційне меню розташоване поруч із логотипом та допомагає орієнтуватися і швидко знаходити потрібну інформацію. Воно забезпечує доступ до наступних сторінок: "Головна", "Каталог", "Про нас", "Контакти". Меню адаптується до розміру екрану, змінюючи свій вигляд на мобільних пристроях (рис. 2.15).

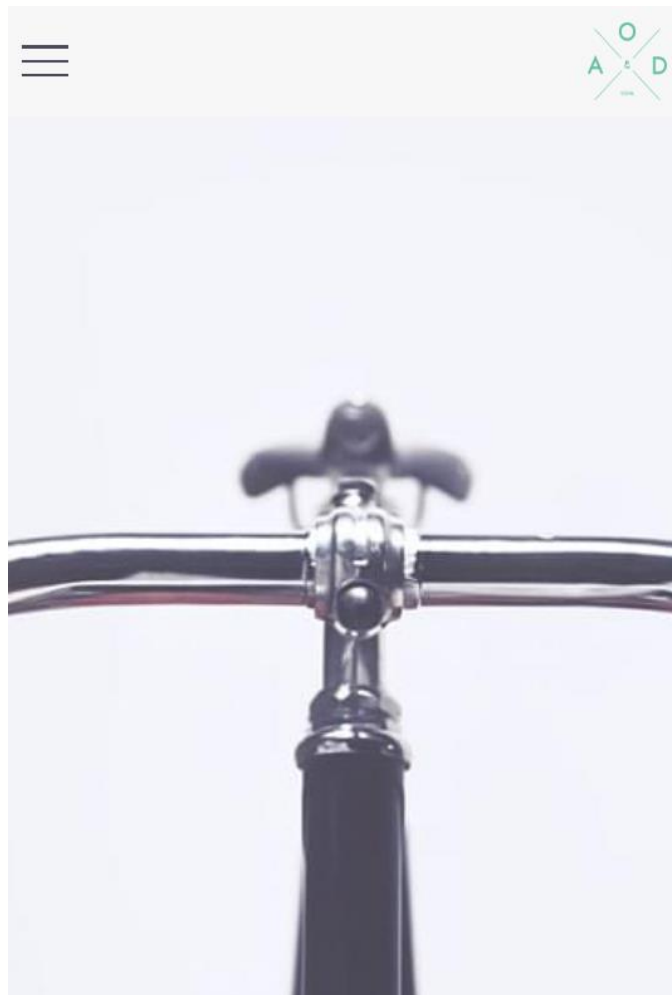
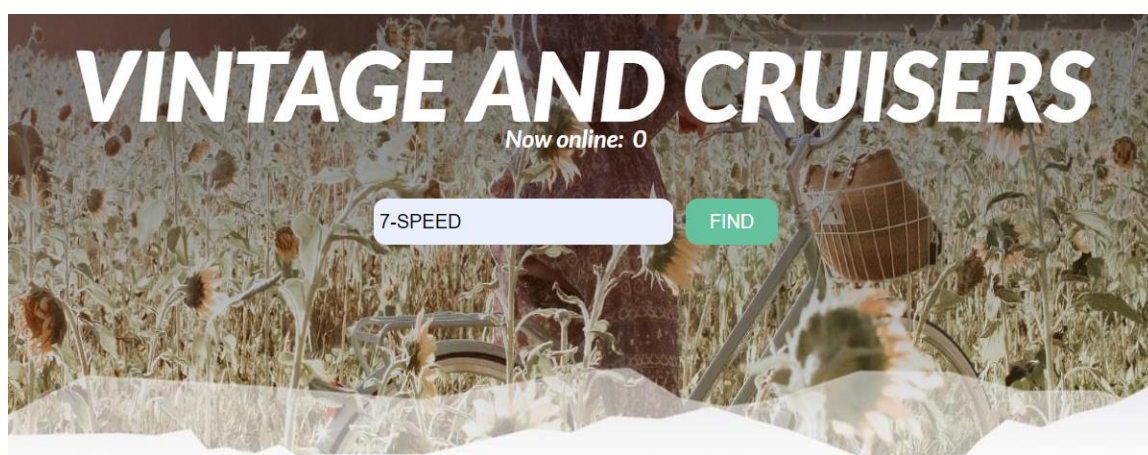


Рис. 2.15. Вигляд головної сторінки вебдодатку "Bicycle" на мобільному пристрої

– іконки кошика та особистого кабінету знаходяться у правому верхньому куті шапки. Вони надають швидкий доступ до функцій покупок та

управління обліковим записом. Для незареєстрованих користувачів іконка особистого кабінету замінюється на "Увійти/Зареєструватися", а іконка кошика недоступна.

– форма глобального пошуку розташована поруч з іконками та дозволяє здійснювати пошук по всьому асортименту велосипедів, запчастин та аксесуарів. При введенні запиту з'являються підказки з популярними або відповідними товарами (рис. 2.16).



LADIES CLASSIC 7-SPEED BIKE



LADIES CLASSIC LITE 7-SPEED BIKE

Рис. 2.16. Приклад використання функції пошуку

Така організація шапки сайту забезпечує користувачам зручну навігацію та швидкий доступ до основних функцій вебдодатку "Bicycle", створюючи комфортне середовище для вибору та придбання вінтажних велосипедів.

Головна сторінка вебдодатку "Bicycle" є візитною карткою магазину та створена для привернення уваги відвідувачів, а також для швидкого доступу до

ключової інформації про асортимент вінтажних велосипедів та переваги магазину (рис. 2.17).

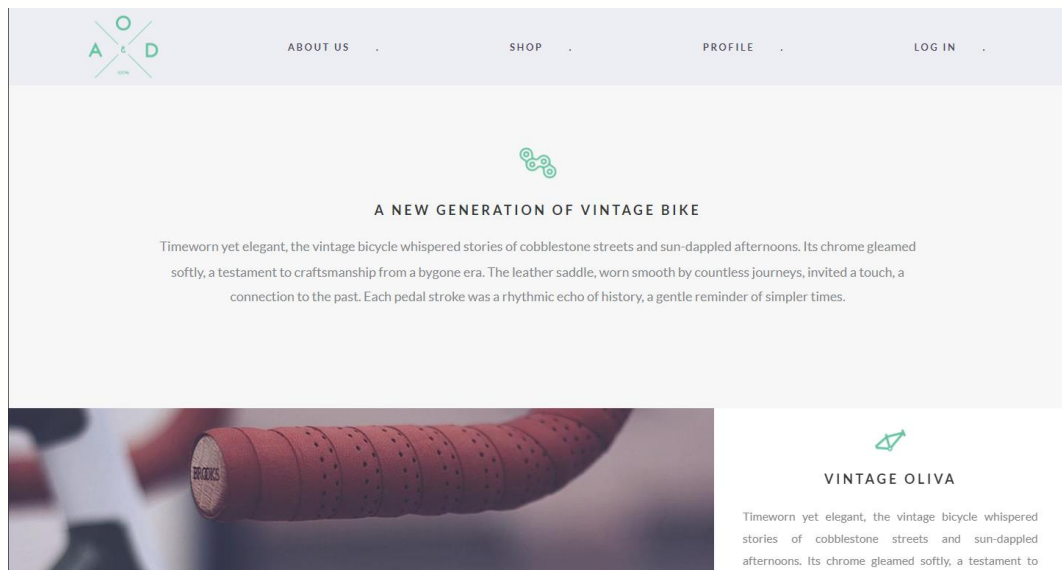


Рис. 2.17. Головна сторінка вебдодатку

Основні елементи головної сторінки включають:

1. Слайдер з фотографіями популярних моделей велосипедів розташований у верхній частині сторінки, одразу під шапкою сайту (рис. 2.18).

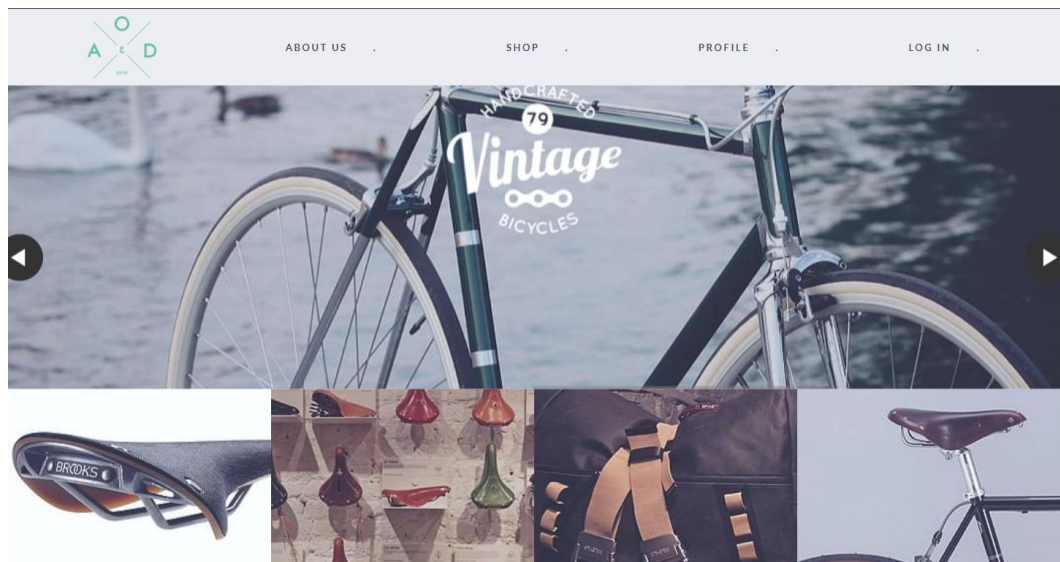


Рис. 2.18. Реалізація слайдерів

Цей елемент виконує кілька важливих функцій:

- Привертає увагу користувачів яскравими, високоякісними зображеннями вінтажних велосипедів.
- Демонструє різноманітність асортименту магазину.
- Дозволяє швидко перейти до сторінки конкретного велосипеда при кліку на зображення.

Слайдер адаптується до розміру екрану, забезпечуючи оптимальне відображення на різних пристроях (рис. 2.19).

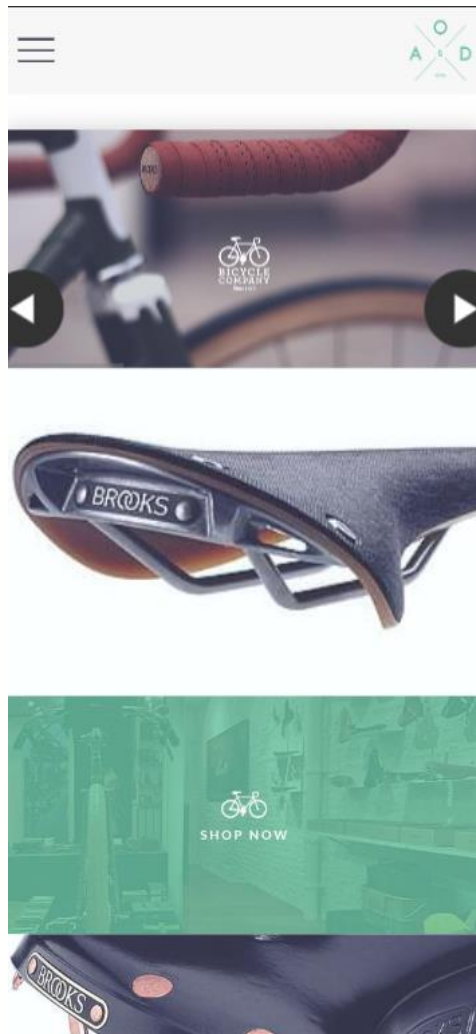



Рис. 2.19. Приклад адаптації слайдера на мобільному устройстві

2. Форма підписки: Центральним елементом сторінки є форма для введення email-адреси. Вона складається з текстового поля з підказкою "enter

your email..." та кнопки "GO" зеленого кольору (рис. 2.20). Ця форма дозволяє відвідувачам підписатися на новини та оновлення магазину.



Stay on the saddle!

Рис. 2.20. Форма для введення email-адреси

3. Контактна інформація: У нижній частині сторінки розміщена адреса магазину "AOD Dnipro" з повними контактними даними, включаючи вулицю, місто та email для зв'язку (рис.2.21).

4. Соціальні мережі: Праворуч від контактної інформації розташовані іконки популярних соціальних мереж (Twitter, Facebook, Pinterest), що дозволяють користувачам швидко перейти на відповідні сторінки магазину (рис.2.21).

5. Інформація про розробника: У правому нижньому куті вказано, що сайт створено за допомогою платформи Yebo (рис.2.21).



Рис. 2.21. Футер вебсайту «Bicycle»

Усі елементи головної сторінки розроблені з урахуванням адаптивного дизайну, забезпечуючи оптимальне відображення на десктопних комп'ютерах, планшетах та смартфонах.

Каталог товарів є ключовим розділом вебдодатку "Bicycle", де представлена вся колекція вінтажних велосипедів (рис. 2.22). Цей розділ розроблений з урахуванням зручності перегляду та пошуку, забезпечуючи користувачам ефективний доступ до асортименту магазину.

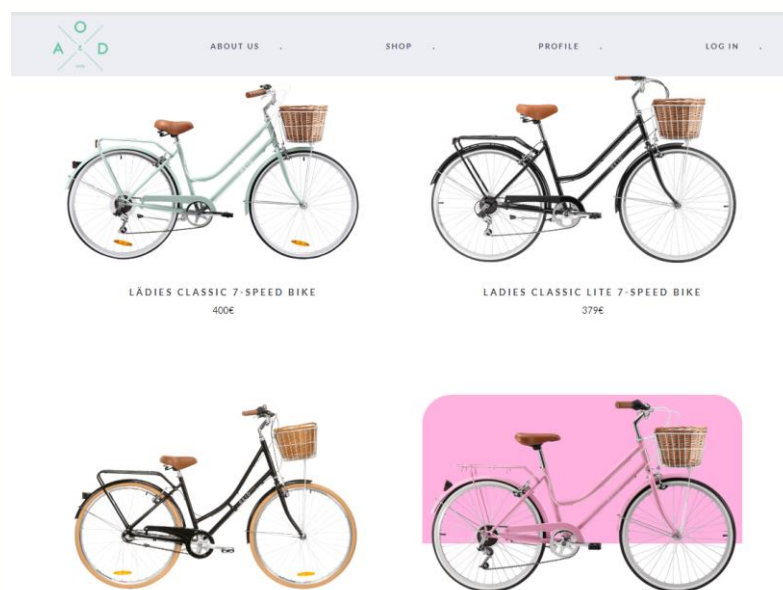


Рис. 2.22. Каталог товарів вебсайту «Bicycle»

Основні елементи каталогу товарів включають:

1. Картки товарів: Кожен вінтажний велосипед представлений у вигляді окремої картки, яка містить:

- високоякісну фотографію велосипеда, що займає більшу частину картки;
- назву моделі, розташовану під фотографією;
- короткий опис, який включає ключові характеристики;
- ціну, виділену контрастним кольором для легкого сприйняття.

Також, як і на інших сторінках, реалізовано форма глобального пошуку.

Вона допомагають користувачам швидко знаходити велосипеди, що відповідають їхнім конкретним вимогам.

2. Кнопка "Додати до кошика" для авторизованих користувачів. Кожна картка товару містить кнопку "Додати до кошика", що дозволяє користувачам швидко здійснити покупку або зберегти велосипед для подальшого розгляду.

Каталог товарів оптимізований для швидкого завантаження, що забезпечує плавний досвід перегляду навіть при повільному інтернет-з'єднанні.

Така організація каталогу товарів забезпечує користувачам зручний та приємний процес вибору вінтажних велосипедів, стимулюючи інтерес до товарів та збільшуючи ймовірність здійснення покупки.

Кошик покупок у вебдодатку "Bicycle" є ключовим елементом процесу здійснення покупки. Він відображає всі обрані користувачем товари у вигляді зручного списку (рис. 2.23), де кожен велосипед представлений фотографією, назвою, ціною та обраною кількістю. Користувачі можуть легко змінювати кількість кожного товару за допомогою інтуїтивних кнопок "+" та "-", при цьому загальна вартість замовлення автоматично оновлюється. Для зручності покупців реалізовано функцію видалення товарів з кошика, а також автоматичний розрахунок вартості доставки. У нижній частині кошика розміщена помітна кнопка "Оформити замовлення", яка веде користувача до наступного етапу покупки.

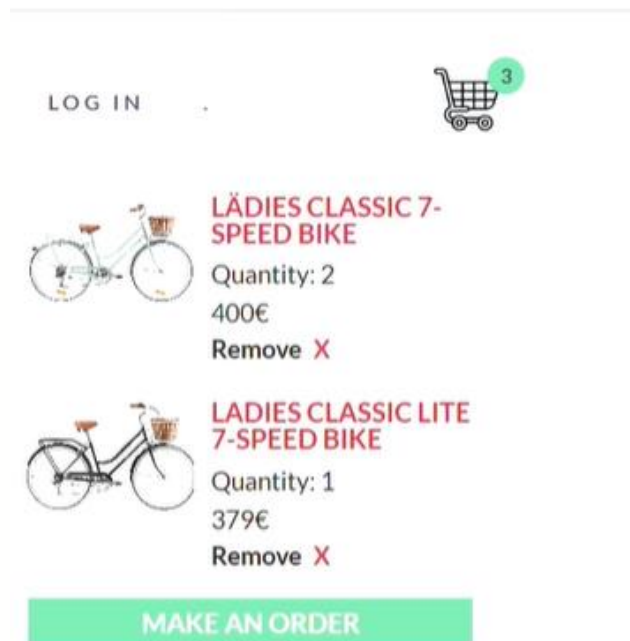


Рис. 2.23. Реалізація кошика покупок

Форми реєстрації та входу розроблені з урахуванням зручності та безпеки користувачів. Вони містять стандартні поля для введення email та паролю, а також додаткові поля для імені та прізвища при реєстрації (рис. 2.13). Реалізована валідація введених даних в реальному часі, що допомагає уникнути помилок при заповненні форм. Користувачі також мають доступ до функції відновлення паролю та можуть легко переключатися між формами реєстрації та входу.

Особистий кабінет у вебдодатку "Bicycle" - це приватний простір користувача, де він може керувати своїм обліковим записом та переглядати інформацію про покупки.

У розділі "Профіль" відображаються основні особисті дані: ім'я, прізвище, email та контактний телефон. Користувач має можливість редагувати ці дані, натиснувши кнопку "Редагувати профіль". Тут же можна змінити пароль облікового запису, забезпечуючи додатковий рівень безпеки.

Розділ "Історія замовлень" (рис. 2.24) представляє собою хронологічний список всіх покупок користувача. Кожне замовлення відображається у вигляді картки з основною інформацією: номер замовлення, дата, загальна сума та статус

виконання. При натисканні на конкретне замовлення відкривається детальна інформація, яка включає перелік придбаних велосипедів, їх кількість, ціну, обрані опції доставки та оплати. Ця функція дозволяє користувачам легко відстежувати свої покупки та за необхідності звертатися до історії замовлень.

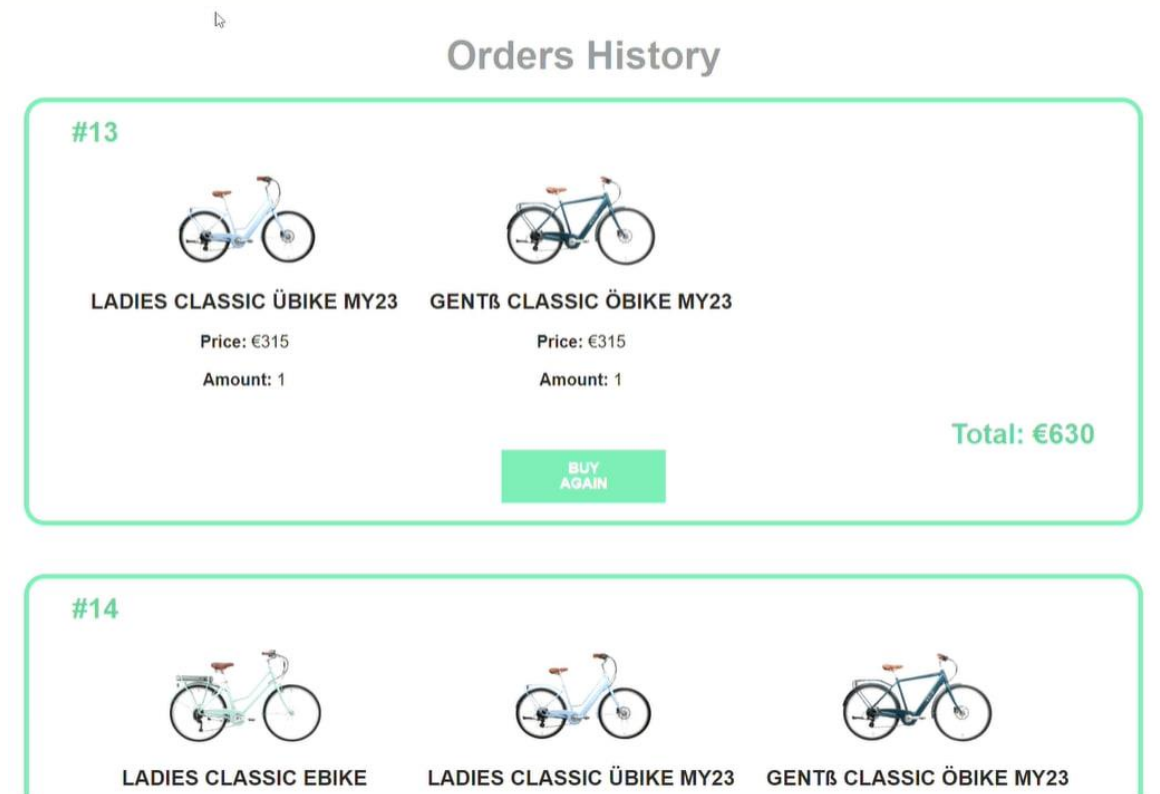


Рис. 2.24. Реалізації історії замовлень

РОЗДІЛ 3

ЕКОНОМІЧНИЙ РОЗДІЛ

3.1. Розрахунок трудомісткості та вартості розробки програмного продукту

Початкові дані:

1. передбачуване число операторів програми – 1311;
2. коефіцієнт складності програми – 1,3;
3. коефіцієнт корекції програми в ході її розробки – 0,07;
4. годинна заробітна плата програміста – 191,49 грн/год;

Згідно з даними, отриманими з вебпорталу «Українська спільнота програмістів (DOU)» [26], середня місячна заробітна плата Junior Software Engineer з досвідом роботи 1-2 роки у період з червня по грудень 2023 року залишалася незмінною та становила 900 \$/місяць. Враховуючи офіційний курс валют НБУ [27] за період розробки проєкту (з 01.09.2023 по 30.04.2024), де середня ціна 1 долара США в середньому дорівнювала 37,447 гривень, що еквівалентно 33 702,3 гривень на місяць. Значення стандартного графіку роботи береться як 176 год/місяць. Тоді годинна заробітна плата програміста становитиме 191,49 грн/год.

5. коефіцієнт збільшення витрат праці внаслідок недостатнього опису задачі – 1,3;
6. коефіцієнт кваліфікації програміста, обумовлений від стажу роботи з даної спеціальності – 0,8;
7. вартість машино-години ЕОМ – 0,65 грн/год.

Розрахунок показника ЕОМ було здійснено з урахуванням актуальних умов на період розробки вебдодатку «Vicusle», яка тривала 8 місяців, з 01.09.2023 по 30.04.2024. У цей період тариф на споживання електроенергії становив 2,64 грн за 1 кВт·год [28]. Оскільки модель ноутбука «VivoBook_ASUSLaptop X571LI_X571LI», використана під час розробки, в середньому споживає 80

Вт·год [29], вартість витрат на електроенергію під час роботи складає: $0,08 \times 2,64 = 0,2$ грн/год. Крім того, щомісячна абонентна плата за інтернет-пакет «Дім» від провайдера «Київстар» у м. Дніпро становить 330 грн/місяць [30]. Враховуючи, що в період з 01.09.2023 по 30.04.2024 середня тривалість місяця складала 30,5 дні, тобто 732 годин, вартість користування інтернетом дорівнює: $330 \text{ грн} / 732 \text{ год} = 0,45$ грн/год. Таким чином, загальна вартість машино-години ЕОМ становить: $0,2 \text{ грн/год} + 0,45 \text{ грн/год} = 0,65 \text{ грн/год}$.

Нормування праці в процесі створення ПЗ істотно ускладнено в силу творчого характеру праці програміста. Тому трудомісткість розробки ПЗ може бути розрахована на основі системи моделей з різною точністю оцінки.

Трудомісткість розробки ПЗ можна розрахувати за формулою (3.1) [31]:

$$t = t_o + t_u + t_a + t_n + t_{отл} + t_d, \text{ людино-годин.} \quad (3.1)$$

де t_o – витрати праці на підготовку й опис поставленої задачі (приймається 50 людино-годин);

t_u – витрати праці на дослідження алгоритму рішення задачі;

t_a – витрати праці на розробку блок-схеми алгоритму;

t_n – витрати праці на програмування по готовій блок-схемі;

$t_{отл}$ – витрати праці на налагодження програми на ЕОМ;

t_d – витрати праці на підготовку документації.

Складові витрати праці визначаються через умовне число операторів у ПЗ, яке розробляється.

Умовне число операторів (підпрограм) визначається за формулою (3.2) [31]:

$$Q = q \cdot C \cdot (1 + p) \quad (3.2)$$

де q – передбачуване число операторів (1311);

C – коефіцієнт складності програми (1,3);

p – коефіцієнт корекції програми в ході її розробки (0,07).

Розрахуємо, умовне число операторів в програмі за формулою (3.2):

$$Q = 1311 \cdot 1,3 \cdot (1 + 0,07) = 1823,6$$

Витрати праці на вивчення опису задачі t_u обчислюються за формулою (3.3) з урахуванням уточнення опису і кваліфікації програміста [31]:

$$t_u = \frac{Q \cdot B}{(75 \cdot 85) \cdot k}, \text{ людино-годин.} \quad (3.3)$$

де B – коефіцієнт збільшення витрат праці внаслідок недостатнього опису задачі (1,4);

k – коефіцієнт кваліфікації програміста, обумовлений від стажу роботи з даної спеціальності, і у разі менше двох років (0,8).

$$t_u = \frac{1823,6 \cdot 1,3}{79 \cdot 0,8} = 37,51 \text{ людино-годин.}$$

Витрати праці на розробку алгоритму рішення задачі встановлюються за формулою (3.4) [31]:

$$t_a = \frac{Q}{(20 \dots 25) \cdot k}, \text{ людино-годин.} \quad (3.4)$$

де Q – умовне число операторів (1391,78);

k – коефіцієнт кваліфікації програміста, обумовлений від стажу роботи з даної спеціальності (0,8).

Розрахунок:

$$t_a = \frac{1823,6}{23 \cdot 0,8} = 99,12 \text{ людино-годин}$$

Витрати на складання програми за готовою блок-схемою, також обчислюються за формулою (3.4):

$$t_n = \frac{1823,6}{21 \cdot 0,8} = 108,67 \text{ людино-годин}$$

Витрати праці на налагодження програми на ЕОМ:

– за умови автономного налагодження одного завдання визначаються за формулою (3.5) [31]:

$$t_{\text{отл}} = \frac{Q}{(4..5) \cdot k}, \text{ людино-годин.} \quad (3.5)$$

Розрахунок:

$$t_{\text{отл}} = \frac{1823,6}{4 \cdot 0,8} = 569,875 \text{ людино-годин.}$$

– за умови комплексного налагодження завдання обчислюються за формулою (3.6) [31]:

$$t_{\text{отл}}^k = 1,5 \cdot t_{\text{отл}}, \text{ людино-годин.} \quad (3.6)$$

Розрахунок:

$$t_{\text{отл}}^k = 1,5 \cdot 569,875 = 854,81 \text{ людино-годин}$$

Витрати праці на підготовку документації, розраховуються за формулою (3.7) [31]:

$$t_d = t_{\text{др}} + t_{\text{до}}, \text{ людино-годин.} \quad (3.7)$$

де $t_{\text{др}}$ – трудомісткість підготовки матеріалів і рукопису;

$t_{\text{до}}$ – трудомісткість редагування, печатки й оформлення документації.

Трудомісткість підготовки матеріалів і рукопису потрібно визначити за формулою (3.8) [31]:

$$t_{др} = \frac{Q}{(15..20) \cdot k}, \text{ люДИНО-ГОДИН.} \quad (3.8)$$

Розрахунок:

$$t_{др} = \frac{1823,6}{18 \cdot 0,8} = 126,64 \text{ люДИНО-ГОДИН}$$

Трудомісткість редагування, печатки й оформлення документації можна обчислити за формулою (3.9) [31]:

$$t_{до} = 0,75 \cdot t_{др}, \text{ люДИНО-ГОДИН.} \quad (3.9)$$

Розрахунок:

$$t_{до} = 0,75 \cdot 126,64 = 94,98 \text{ люДИНО-ГОДИН.}$$

Розрахунок праці на підготовку документації, за формулою (3.7):

$$t_{д} = 126,64 + 94,98 = 221,62 \text{ люДИНО-ГОДИН.}$$

Тож розрахунок трuдомісткості розробки програмного забезпечення за формулою (3.1):

$$t = 50 + 37,5 + 99,12 + 108,67 + 854,81 + 221,62 = 1371,72 \text{ люДИНО-ГОДИН.}$$

3.2. Рахунок витрат на створення програми

Витрати на розробку ПЗ складаються із заробітної плати розробника та вартості машинного часу, який використовується для налагодження програми на ЕОМ та розраховується за формулою (3.10) [31]:

$$K_{ПО} = Z_{ЗП} + Z_{МВ}, \text{ грн.} \quad (3.10)$$

де $Z_{ЗП}$ – витрати на заробітну плату програміста;

Z_{MB} – витрати машинного часу.

Заробітна плата виконавця визначається за формулою (3.11) [31]:

$$Z_{зп} = t \cdot C_{пп}, \text{ грн.} \quad (3.11)$$

де t – загальна трудомісткість (732,17 людино-годин);

$C_{пп}$ – середня годинна заробітна плата програміста (191,49 грн/год).

Розрахунок:

$$Z_{зп} = 732,17 \cdot 191,49 = 140,200.23 \text{ грн}$$

Вартість машинного часу, необхідного для налагодження програми на ЕОМ, обчислюється за формулою (3.12) [31]:

$$Z_{MB} = t_{отл} \cdot C_{мч}, \text{ грн.} \quad (3.12)$$

де $t_{отл}$ – трудомісткість налагодження програми на ЕОМ (521,93 год);

$C_{мч}$ – вартість машино-години ЕОМ (0,74 грн/год).

Розрахунок:

$$Z_{MB} = 569,875 \cdot 0,65 = 370,42 \text{ грн.}$$

Розрахунок витрат на створення ПЗ, що обчислюється за формулою (3.10) [31]:

$$K_{пз} = 140\,200,23 + 370,42 = 140\,570,65 \text{ грн.}$$

Очікуваний період розробки ПЗ розраховується за формулою (3.13) [31]:

$$T = \frac{t}{B_k \cdot F_p}, \text{ міс.} \quad (3.13)$$

де t – загальна трудомісткість (1371.72 людино-годин);

B_k – число виконавців (1);

F_p – місячний фонд робочого часу (при 40 годинному робочому тижні
 $F_p = 176$ годин).

Розрахунок:

$$T = \frac{1371,72}{1 \cdot 176} = 7,8 \text{ міс}$$

Висновок: для розробки веборієнтованого додатку з продажу вінтажних велосипедів за допомогою JavaScript та PHP загальна трудомісткість становить 1 371,72 людино-годин, згідно чого очікуваний період створення програмного забезпечення дорівнює майже 8 місяців, а витрати на створення – 140 570,65 грн.

ВИСНОВКИ

Згідно з поставленим завданням та метою кваліфікаційної роботи, було розроблено веборієнтований додаток "Bicycle" для продажу вінтажних велосипедів з використанням JavaScript та PHP.

Після аналізу існуючих рішень у сфері онлайн-продажу велосипедів було виявлено потребу в спеціалізованій платформі для вінтажних моделей. Створений додаток враховує особливості цього ринку та забезпечує ефективну презентацію унікальних товарів.

Практичне значення розробленого додатку полягає у наданні зручного інструменту для продажу та придбання вінтажних велосипедів. «Bicycle» забезпечує:

- ефективну діяльність інтернет-магазину вінтажних велосипедів;
- розширення ринку збуту для продавців;
- доступ до каталогу вінтажних велосипедів для покупців з будь-якої точки світу;
- безпечні онлайн-транзакції між продавцями та покупцями.

Користувачам надається наступний функціонал:

- реєстрація та авторизація;
- перегляд каталогу товарів з інформацією про кожен велосипед;
- додавання товарів до кошика та оформлення замовлень;
- управління особистим кабінетом та історією замовлень.

Додаток розроблено з використанням JavaScript для клієнтської частини та PHP для серверної. Для зберігання даних використано MySQL, а для стилізації - Bootstrap. Реалізовано адаптивний дизайн для коректного відображення на різних пристроях.

Згідно з розрахунками в економічному розділі, на розробку додатку було витрачено 1371,72 людино-годин, очікуваний період розробки склав близько 8 місяців, а витрати на створення - 140 570,65 грн.

Подальший розвиток проекту "Bicycle" може включати розширення функціоналу, такого як впровадження системи відгуків та рейтингів, а також оптимізація для ще більшого покращення швидкодії та зручності використання.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. McKinsey & Company: "E-commerce: The Next Chapter". URL: <https://www.mckinsey.com/business-functions/marketing-and-sales/our-insights/e-commerce-the-next-chapter> (дата звернення: 12.05.2024).
2. Nielsen Norman Group: "Mobile E-commerce: State of the Industry". URL: <https://www.nngroup.com/articles/mobile-ecommerce-usability/> (дата звернення: 18.05.2024).
3. HubSpot: "E-commerce Marketing: The Ultimate Guide". URL: <https://blog.hubspot.com/marketing/ecommerce-marketing> (дата звернення: 25.05.2024).
4. TechCrunch: "The Role of Technology in E-commerce". URL: <https://techcrunch.com/> (дата звернення: 01.06.2024).
5. Shopify Blog: "The Future of E-commerce: Trends to Watch in 2024". URL: <https://www.shopify.com/blog> (дата звернення: 07.06.2024).
6. IBM: "Application Performance Monitoring". URL: <https://www.ibm.com/de-de/products/instana/application-performance-monitoring> (дата звернення: 13.06.2024).
7. Wired: "The Evolution of E-commerce Technology". URL: <https://www.wired.com/> (дата звернення: 19.06.2024).
8. Business Insider: "E-commerce Trends to Know". URL: <https://www.businessinsider.com/> (дата звернення: 24.06.2024).
9. Digital Commerce 360: "E-commerce Trends to Watch". URL: <https://www.digitalcommerce360.com/> (дата звернення: 05.05.2024).
10. Sommerville, I. (2016). Software Engineering (10th ed.). Pearson.
11. Pressman, R. S., & Maxim, B. R. (2020). Software Engineering: A Practitioner's Approach (9th ed.). McGraw-Hill Education.

12. Semrush: "Google PageSpeed Insights". URL: <https://www.semrush.com/blog/google-pagespeed-insights/> (дата звернення: 10.05.2024).
13. Eye-Able: "Barrierefreiheits-Service". URL: <https://eye-able.com/barrierefreiheits-service-eye-able/> (дата звернення: 16.05.2024).
14. Fastly: "DevOps Roadmap". URL: <https://learn.fastly.com/devops-roadmap.html> (дата звернення: 22.05.2024).
15. Wikipedia: "Інформаційна безпека". URL: https://uk.wikipedia.org/wiki/Інформаційна_безпека (дата звернення: 28.05.2024).
16. Wikipedia: "Загрози інформаційної безпеки". URL: https://uk.wikipedia.org/wiki/Загрози_інформаційної_безпеки (дата звернення: 03.06.2024).
17. Stripe: "API Documentation". URL: <https://stripe.com/docs/api> (дата звернення: 09.06.2024).
18. Друкарня: "HTML теги та атрибути". URL: <https://drukarnia.com.ua/articles/html-tegi-ta-atributi-yak-voni-pracyuyut-i-sho-potribno-znati-9FSYK> (дата звернення: 15.06.2024).
19. Документація API платіжної системи Stripe / URL: <https://stripe.com/docs/api>. (дата звернення: 21.06.2024).
20. SheCodes: "Basic Structure of an HTML Document". URL: <https://www.shecodes.io/athena/33449-what-is-the-basic-structure-of-an-html-document> (дата звернення: 06.05.2024).
21. Asabix: "What is Website Layout in Simple Words". URL: <https://asabix.com.ua/what-is-website-layout-in-simple-words/> (дата звернення: 11.05.2024).
22. Medium: "Best 3 jQuery Alternatives of Modern JavaScript World". URL: <https://medium.com/@sofiullahc/best-3-jquery-alternatives-of-modern-javascript-world-b2d5adf539a6> (дата звернення: 17.05.2024).

23. BSSCommerce: "JavaScript, jQuery, AJAX: Are They the Same or Different?". URL: <https://bsscommerce.com/blog/javascript-jquery-ajax-are-they-the-same-or-different/> (дата звернення: 23.05.2024).

24. DOU: "MySQL Forum Topic". URL: <https://dou.ua/forums/topic/45890/> (дата звернення: 29.05.2024).

25. Wikipedia: "Модель-вид-контролер". URL: <https://uk.wikipedia.org/wiki/Модель-вид-контролер> (дата звернення: 04.06.2024).

26. Статистика зарплат програмістів, тестувальників і РМ в Україні: DOU. URL: <https://jobs.dou.ua/salaries/?period=2023-12&position=3D%20Artist&experience=0-2> (дата звернення: 10.06.2024).

27. Офіційний курс гривні щодо іноземних валют: Національний банк України. URL: <https://bank.gov.ua/ua/markets/exchangerates> (дата звернення: 16.06.2024).

28. Тарифи на електроенергію з 1.06.2023 по 31.05.2024: Мінфін. URL: <https://index.minfin.com.ua/ua/tariff/electric/2023-06-01/> (дата звернення: 22.06.2024).

29. VivoBook_ASUSLaptop X571LI_X571LI. URL: <https://www.asus.com/de/> (дата звернення: 07.05.2024).

30. Тарифи Інтернет та Інтерактивного ТБ: Київстар. URL: <https://kyivstar.ua/home-internet> (дата звернення: 13.05.2024).

31. Вагонова О.Г., Романюк Н.Н. Факультет менеджменту, кафедра прикладної економіки: методичні вказівки з виконання економічного розділу в дипломних роботах.

ЛІСТИНГ ПРОГРАМИ

index.html

```
<!DOCTYPE html>
<html lang="en">
  <!-- Head-->
  <head>
    <!-- Title -->
    <title>Bicycle</title>

    <!-- SEO -->
    <meta name="description" content="" />

    <!-- Favicons -->
    <link
      rel="apple-touch-icon"
      sizes="180x180"
      href="../../assets/images/main/favicon/apple-touch-icon.png"
    />
    <link
      rel="icon"
      type="image/png"
      sizes="32x32"
      href="../../assets/images/main/favicon/favicon-32x32.png"
    />
    <link
      rel="icon"
      type="image/png"
      sizes="16x16"
      href="../../assets/images/main/favicon/favicon-16x16.png"
    />
    <link
      rel="manifest"
      href="../../assets/images/main/favicon/site.webmanifest"
    />
    <link
      rel="mask-icon"
      href="../../assets/images/main/favicon/safari-pinned-tab.svg"
      color="#3f91a9"
    />
    <link
      rel="shortcut icon"
      href="../../assets/images/main/favicon/favicon.ico"
    />
    <meta name="msapplication-TileColor" content="#0f0b0a" />
    <meta
      name="msapplication-config"
      content="../../assets/images/main/favicon/browserconfig.xml"
    />
```



```

<meta name="theme-color" content="#d3d3d3" />

<!-- Charset -->
<meta charset="UTF-8" />

<!-- Adaptive meta tags -->
<meta http-equiv="X-UA-Compatible" content="IE=edge" />
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
</head>
<!-- Body-->
<body>
  <div class="wrapper">
    <!-- Header -->
    <header class="header">
      <div class="header__container _container">
        <div class="header__row">
          <div class="header__burger">
            <span></span>
          </div>

          <div class="header__logo">
            <a href="./index.html" class="header__logo-link logo-link">
              
            </a>
          </div>

          <nav class="header__nav">
            <ul class="header__list">
              <li class="header__item">
                <a href="index.html" class="header__logo-link logo-link">
                  
                </a>
              </li>

              <li class="header__item">
                <a href="#about" class="header__link header__link_after">
                  about us
                </a>
              </li>

              <li class="header__item">
                <a
                  href="./catalog.html"

```

```

        class="header__link header__link_after"
    >
        shop
    </a>
</li>

<li class="header__item">
    <a href="#contact" class="header__link header__link_after">
        profile
    </a>
</li>

<li class="header__item">
    <a
        href="/login.html "
        class="header__link header__link_after"
    >
        log in
    </a>
</li>
</ul>
</nav>
</div>
</div>
</header>
<!-- Header -->

<!-- Main content -->
<main class="content">
    <!-- Intro start -->
    <div class="intro">
        <div class="intro__bg ibg">
            
        </div>
    </div>
    <!-- Intro end -->

    <!-- About start -->
    <div class="about" id="about">
        <div class="about__container">
            <div class="about__body">
                <div class="about__icon">
                    
                </div>
            </div>
        </div>
    </div>

```

```
</div>
```

```
<div class="about__title title">  
  <h1>A NEW GENERATION OF VINTAGE BIKE</h1>  
</div>
```

```
<p class="about__text">  
  Lorem Ipsum is simply dummy text of the printing and typesetting  
  industry. Lorem Ipsum has been the industry's standard dummy  
  text ever since the 1500s, when an unknown printer took a galley  
  of type and scrambled it to make a type specimen book. It has  
  survived not only five centuries, but also the leap into  
  electronic typesetting, remaining essentially unchanged.
```

```
</p>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
<!-- About end -->
```

```
<!-- Work section start -->
```

```
<section class="work" id="work">
```

```
<div class="work__container container-full">
```

```
<div class="work__cards">
```

```
<div class="work__card_first lg-image">
```

```

```

```
</div>
```

```
<div class="work__card_second card-content">
```

```
<div class="card-content__body">
```

```
<div class="card-content__icon">
```

```

```

```
</div>
```

```
<h2 class="card-content__title title">vintage Oliva</h2>
```

```
<p class="card-content__text">
```

```
  Lorem Ipsum is simply dummy text of the printing and  
  typesetting industry. Lorem Ipsum has been the industry's  
  standard dummy text ever since the 1500s, when an unknown  
  printer took a galley of type and scrambled it to make a  
  type specimen book. It has survived.
```

```
</p>
```

```
</div>
```

```
</div>
```

```
<div class="work__card_third sm-image">  
    
</div>
```

```
<div class="work__card_fourth card-content">  
  <div class="card-content__body">  
    <div class="card-content__icon">  
        
    </div>
```

```
<h2 class="card-content__title title">La Boriosa</h2>
```

```
<p class="card-content__text">  
  Lorem Ipsum is simply dummy text of the printing and  
  typesetting industry. Lorem Ipsum has been the industry's  
  standard dummy text ever since the 1500s, when an unknown  
  printer took a galley of type and scrambled it to make a  
  type specimen book. It has survived.  
</p>  
</div>  
</div>
```

```
<div class="work__card_fifth sm-image">  
    
</div>
```

```
<div class="work__card_six lg-image">  
    
</div>
```

```
<div class="work__card_seventh medium-image">  
  
</div>

<div class="work__card_eighth card-content">
    <div class="card-content__body">
        <div class="card-content__icon">
            
        </div>

        <h2 class="card-content__title title">
            retrò Bike - M. Hulot
        </h2>

        <p class="card-content__text">
            Lorem Ipsum is simply dummy text of the printing and
            typesetting industry. Lorem Ipsum has been the industry's
            standard dummy text ever since the 1500s, when an unknown
            printer took a galley of type and scrambled it to make a
            type specimen book. It has survived.
        </p>
    </div>
</div>

<div class="work__card_ninth sm-image">
    
</div>

<div class="work__card_tenth medium-image">
    
</div>
</div>
</div>
</section>
<!-- Work section end -->

<!-- Quotes section start -->
<div class="slider">

```

```

<div class="slider__container">
  <div class="slider__row">
    <div class="slider__pic">
      
    </div>

    <div class="slider__items">
      <input type="radio" name="s" id="s1" />
      <input type="radio" name="s" id="s2" />
      <input type="radio" name="s" id="s3" />
      <input type="radio" name="s" id="s4" />

      <div class="slider__items-item s1">
        "Normality is a paved road: it's comfortable to walk on, but
        flowers don't grow on it"
      </div>
      <div class="slider__items-item">
        "What is drawing? This is the ability to break through the
        iron the wall that stands between what you feel and what you
        know how"
      </div>
      <div class="slider__items-item">
        "In the evening I walked along the deserted seashore. It
        wasn't fun and not sad - it was great"
      </div>
      <div class="slider__items-item">
        "My love is not made of moonlight and roses, but sometimes
        prosaic as Monday morning"
      </div>
    </div>
    <div class="slider-nav">
      <label for="s1" class="slider-nav-item"></label>
      <label for="s2" class="slider-nav-item"></label>
      <label for="s3" class="slider-nav-item"></label>
      <label for="s4" class="slider-nav-item"></label>
    </div>
  </div>
</div>
</div>
</div>
</div>
<!-- Quotes section end -->

<!-- new -->
<section class="carousel" aria-label="Gallery">
  <ol class="carousel__viewport">
    <li id="carousel__slide1" tabindex="0" class="carousel__slide">
      <div class="carousel__snapper">
        <a href="#carousel__slide4" class="carousel__prev">
          Go to last slide

```

```

    </a>
    <a href="#carousel__slide2" class="carousel__next">
      Go to next slide
    </a>
  </div>
</li>
<li id="carousel__slide2" tabindex="0" class="carousel__slide">
  <div class="carousel__snapper"></div>
  <a href="#carousel__slide1" class="carousel__prev">
    Go to previous slide
  </a>
  <a href="#carousel__slide3" class="carousel__next">
    Go to next slide
  </a>
</li>
<li id="carousel__slide3" tabindex="0" class="carousel__slide">
  <div class="carousel__snapper"></div>
  <a href="#carousel__slide2" class="carousel__prev">
    Go to previous slide
  </a>
  <a href="#carousel__slide4" class="carousel__next">
    Go to next slide
  </a>
</li>
<li id="carousel__slide4" tabindex="0" class="carousel__slide">
  <div class="carousel__snapper"></div>
  <a href="#carousel__slide3" class="carousel__prev">
    Go to previous slide
  </a>
  <a href="#carousel__slide1" class="carousel__next">
    Go to first slide
  </a>
</li>
</ol>
</section>
<!-- new -->

<!-- Shop start -->
<div class="shop" id="shop">
  <div class="shop__container container-full">
    <div class="shop__cards">
      <div class="shop__card shop__card_square">
        
      </div>

      <div class="shop__card shop__card_square">
        
</div>

<div class="shop__card shop__card_ver">
  
</div>

<div class="shop__card shop__card_ver">
  
</div>

<div class="shop__card shop__card_hor">
  
</div>

<div class="shop__card shop__card_hor">
  
</div>

<div class="shop__card shop__card_square">
  
</div>

<div class="shop__card shop__card_square">
  
</div>

```



```

    />
  </div>
</div>

<div class="shop__footer">
  <a href="/catalog.html" class="shop__button">
    DISCOVER THE SHOP
  </a>
</div>
</div>
</div>
</div>
<!-- Shop end -->

<!-- Contact start -->
<div class="contact" id="contact">
  <div class="contact__container">
    <div class="contact__body">
      <a href="index.html" class="contact__logo-link logo-link">
        
      </a>

      <p class="contact__text">Stay on the saddle!</p>

      <form class="contact__form">
        <fieldset class="contact__form_wrap">
          <p class="contact__form_items">
            <input
              type="email"
              required
              name="usermail"
              class="contact__form_field"
              placeholder="enter your email..."
            />
            <button type="submit" class="contact__form_submit">
              GO
            </button>
          </p>
        </fieldset>
      </form>
    </div>
  </div>
</div>
</div>
<!-- Contact end -->
</main>
<!-- Main content -->

<!-- Footer -->

```

```

<footer class="footer">
  <div class="footer__container">
    <div class="footer__row">
      <address class="footer__contact">
        AOD New York
        <br />
        <mark>
          2250 Lexington Avenue
          <br />
          New York,br NY 10035&ensp;- info@domain.com
        </mark>
      </address>

      <div class="footer__social">
        <a
          href="https://twitter.com/?lang=ru"
          class="footer__social-link"
        >
          
        </a>

        <a href="https://www.facebook.com/" class="footer__social-link">
          
        </a>

        <a href="https://www.pinterest.ru/" class="footer__social-link">
          
        </a>
      </div>

      <div class="footer__copyright">
        Handcrafted by
        <a href="#" class="footer__copyright-link">
          
        </a>
      </div>

```

```

    </div>
  </div>
</div>
</footer>
<!-- Footer -->
</div>
</body>
</html>

```

script.mjs main

```

// IMPORT
import './index.html';
import './style.scss';

// Function for inverting custom image to background image
function ibg() {
  const imgBackground = document.querySelectorAll('.ibg');
  for (let i = 0; i < imgBackground.length; i++) {
    if (imgBackground[i].querySelector('img')) {
      imgBackground[i].style.backgroundImage = `url(${imgBackground[i]
        .querySelector('img')
        .getAttribute('src')})`;
    }
  }
}

ibg();

let slideIndex = 1;
showSlides(slideIndex);

function plusSlides(n) {
  showSlides((slideIndex += n));
}

function currentSlide(n) {
  showSlides((slideIndex = n));
}

function showSlides(n) {
  let i;
  const slides = document.getElementsByClassName('mySlides');
  const dots = document.getElementsByClassName('dot');

  if (n > slides.length) {
    slideIndex = 1;
  }
  if (n < 1) {
    slideIndex = slides.length;
  }
}

```

```

for (i = 0; i < slides.length; i++) {
  slides[i].style.display = 'none';
}
for (i = 0; i < dots.length; i++) {
  dots[i].className = dots[i].className.replace(' active', '');
}
slides[slideIndex - 1].style.display = 'block';
dots[slideIndex - 1].className += ' active';
}

// Scroll to anchors

// eslint-disable-next-line func-names
(function () {
  // eslint-disable-next-line func-names
  const smoothScroll = function (targetEl, duration) {
    const headerElHeight = document.querySelector('.header').clientHeight;
    const target = document.querySelector(targetEl);
    const targetPosition = target.getBoundingClientRect().top - headerElHeight;
    const startPosition = window.pageYOffset;
    let startTime = null;

    // eslint-disable-next-line func-names
    const ease = function (t, b, c, d) {
      // eslint-disable-next-line no-param-reassign
      t /= d / 2;
      if (t < 1) return (c / 2) * t * t + b;
      // eslint-disable-next-line no-plusplus, no-param-reassign
      t--;
      return (-c / 2) * (t * (t - 2) - 1) + b;
    };

    // eslint-disable-next-line func-names
    const animation = function (currentTime) {
      if (startTime === null) startTime = currentTime;
      const timeElapsed = currentTime - startTime;
      const run = ease(timeElapsed, startPosition, targetPosition, duration);
      window.scrollTo(0, run);
      if (timeElapsed < duration) requestAnimationFrame(animation);
    };
    requestAnimationFrame(animation);
  };

  // eslint-disable-next-line func-names
  const scrollTo = function () {
    const links = document.querySelectorAll('.js-scroll');
    links.forEach(each => {
      // eslint-disable-next-line func-names
      each.addEventListener('click', function () {
        const currentTarget = this.getAttribute('href');

```

```

        smoothScroll(currentTarget, 1000);
    });
});
};
scrollTo();
})();

```

script.mjs catalog

```

import $ from 'jquery';

import './index.html';

import json from '../assets/json/json.json';

// eslint-disable-next-line import/extensions, no-unused-vars

import './style.scss';

// search
let lastResFind = "";
let copy_page = "";

const textFindBtn = document.querySelector('.text-find');
function TrimStr(s) {
    s = s.replace(/^\s+/g, "");
    return s.replace(/\s+$/g, "");
}
textFindBtn.addEventListener('click', () => FindOnPage('text-to-find'));

function FindOnPage(inputId) {
    console.log('work');
    // шукає текст на сторінці, у параметр передається ID поля для введення
    let obj = window.document.getElementById(inputId);
    let textToFind;

    if (obj) {
        textToFind = TrimStr(obj.value);
    } else {
        alert('The entered phrase was not found');
        return;
    }
    if (textToFind == "") {
        alert('You did not enter anything');
        return;
    }

    if (document.body.innerHTML.indexOf(textToFind) == '-1')
        alert('Nothing found, please check your input!');

    if (copy_page.length > 0) document.body.innerHTML = copy_page;

```

```

else copy_page = document.body.innerHTML;

document.body.innerHTML = document.body.innerHTML.replace(
    eval('/name=' + lastResFind + '/gi'),
    ''
);
document.body.innerHTML = document.body.innerHTML.replace(
    eval('/' + textToFind + '/gi'),
    '<a name=' +
    textToFind +
    " style='background: #64c29e'" +
    textToFind +
    '</a>'
); // Замінюємо знайдений текст посиланнями з якорем;
lastResFind = textToFind; // зберігаємо фразу для пошуку, щоб надалі за нею стерти всі
посилання
window.location = '#' + textToFind; // переміщуємо скрол до останнього знайденого збігу

return false;
}

function getUserSurname(welcomeUserSurname) {
    let userSurname;
    $.ajax({
        url: './assets/php/catalog/welcome.php',
        method: 'post',
        async: false,
        data: {
            welcomeUserSurname,
        },
        success: (response) => {
            try {
                if (response) {
                    const responseFromServer = JSON.parse(response);

                    if (responseFromServer.status === 'success') {
                        userSurname = responseFromServer.userSurname;
                    }
                }
            } catch (error) {
                console.log(error);
            }
        },
    });

    return userSurname;
}

function updateOnlineUsers(amountOfOnline, nowOnlineField) {
    $.ajax({
        url: './assets/php/teststatus/online-amount.php',

```

```

method: 'post',
data: { amountOfOnline },
success: (response) => {
  try {
    if (response) {
      const responseFromServer = JSON.parse(response);

      if (responseFromServer.status === 'success') {
        nowOnlineField.textContent = responseFromServer.online;
      }
    }
  } catch (error) {
    console.log(error);
  }
},
});
}

```

```
let orderIdCount = 1;
```

```

window.onload = function () {
  const welcomeUser = document.querySelector('.welcome__user');
  const nowOnlineField = document.querySelector('.welcome__amount-online span');

```

```
const userSurname = getUserSurname('userSurname');
```

```

if (userSurname) {
  welcomeUser.textContent = `Hello Mr/Mrs ${userSurname}`;

```

```

  setInterval(() => {
    const amountOfOnline = Number(nowOnlineField.textContent);
    updateOnlineUsers(amountOfOnline, nowOnlineField);
  }, 3000);
}

```

```

// eslint-disable-next-line no-use-before-define
document.addEventListener('click', documentActions);

```

```
getUserSurname('value');
```

```
// Actions (click)
```

```

function documentActions(e) {
  const targetElement = e.target;

```

```

  if (targetElement.classList.contains('products__btn-add')) {
    const productId = targetElement.closest('.products__item').dataset.pid;
    // eslint-disable-next-line no-use-before-define
    addToCart(targetElement, productId);
    // eslint-disable-next-line no-unused-expressions
    e.preventDefault();
  }
}

```

```

// Shopping cart
if (targetElement.classList.contains('shopping-cart__icon')) {
  if (document.querySelector('.cart-list').children.length > 0) {
    document
      .querySelector('.cart-header__body')
      .classList.toggle('_active'); // toggle
  }
  e.preventDefault();
} else if (
  !targetElement.closest('.header__shopping-cart') &&
  !targetElement.classList.contains('products__btn-add') &&
  !targetElement.classList.contains('products__btn-remove') &&
  !targetElement.classList.contains('cart-list__remove')
) {
  document.querySelector('.cart-header__body').classList.remove('_active');
} // Remove from cart
if (targetElement.classList.contains('products__btn-remove')) {
  // eslint-disable-next-line no-shadow
  const targetElement = document.querySelector('.cart-list__remove');
  const productId =
    targetElement.closest('.cart-list__item').dataset.cartPid;
  // eslint-disable-next-line no-use-before-define
  updateCart(targetElement, productId, false);
  e.preventDefault();
}

if (targetElement.classList.contains('cart-list__remove')) {
  const productId =
    targetElement.closest('.cart-list__item').dataset.cartPid;

  // eslint-disable-next-line no-use-before-define
  updateCart(targetElement, productId, false);
  e.preventDefault();
}

// Create an object
if (targetElement.classList.contains('cart-list__btn-active')) {
  console.log(targetElement);
  // targetElement.setAttribute('type', 'button');

  // eslint-disable-next-line no-new-object
  let username = getUsername('New user name');
  console.log(username);

  targetElement.setAttribute('type', 'submit');

  const orderObj = {};
  orderObj.id = orderIdCount++;
  orderObj.username = username;
  orderObj.isPaid = 'not';
}

```



```

let orderObjId = orderObj.id;
const orderObjIsPaid = orderObj.isPaid;
// eslint-disable-next-line no-new-object

const products = document.querySelectorAll('.cart-list__item');

products.forEach((element) => {
  const productObjId = element.dataset.cartPid;

  const productObjImage = element.querySelector(
    '.products__image-img'
  ).src;

  const productObjTitle = element
    .querySelector('.cart-list__title')
    .textContent.trim();

  const productObjPrice = element
    .querySelector('.cart-list__price')
    .innerHTML.trim();

  const productObjQuantity = element
    .querySelector('.cart-list__quantity span')
    .innerHTML.trim();

  const cartObj = new Object();

  orderObj.order = cartObj;

  cartObj.id = productObjId;
  cartObj.image = productObjImage;
  cartObj.title = productObjTitle.trim();
  cartObj.price = productObjPrice.slice(0, -1);
  cartObj.quantity = productObjQuantity;

  const order = orderObj;

  const orderId = orderObjId;

  const itemId = productObjId;

  const itemPrice = cartObj.price;

  $.ajax({
    url: './assets/php/catalog/cart.php',
    method: 'post',
    // async: false,
    data: {
      order,
      itemId,

```

```

    orderId,
    username,
    productObjImage,
    productObjTitle,
    itemPrice,
    productObjQuantity,
    orderObjIsPaid,
  },
  success: (response) => {
    if (response) console.log(response);
  },
});
});
}
}

```

// AddToCart

```

function addToCart(productButton, productId) {
  if (!productButton.classList.contains('_hold')) {
    productButton.classList.add('_hold');
    productButton.classList.add('_fly');

    const cart = document.querySelector('.header__shopping-cart');
    const product = document.querySelector(`[data-pid="${productId}"]`);
    const productImage = product.querySelector('.products__image-img');

    const productImageFly = productImage.cloneNode(true);

    const productImageFlyWidth = productImage.offsetWidth;
    const productImageFlyHeight = productImage.offsetHeight;

    const productImageFlyTop = productImage.getBoundingClientRect().top;
    const productImageFlyLeft = productImage.getBoundingClientRect().left;

    productImageFly.setAttribute('class', '_flyImage _ibg');
    productImageFly.style.cssText = `
left: ${productImageFlyLeft}px;
top: ${productImageFlyTop}px;
width: ${productImageFlyWidth}px;
height: ${productImageFlyHeight}px;
`;

    document.body.append(productImageFly);

    const cartFlyLeft = cart.getBoundingClientRect().right;
    const cartFlyTop = cart.getBoundingClientRect().top;

    productImageFly.style.cssText = `
left: ${cartFlyLeft}px;
top: ${cartFlyTop}px;
width: 0px;

```

```

height: 0px;
opacity: 1;
`;

productImageFly.addEventListener('transitionend', () => {
  if (productButton.classList.contains('_fly')) {
    productImageFly.remove();
    // eslint-disable-next-line no-use-before-define
    updateCart(productButton, productId);
    productButton.classList.remove('_fly');
  }
});
}
}

// UpdateCart
function updateCart(productButton, productId, productAdd = true) {
  const cart = document.querySelector('.header__shopping-cart');
  const cartForm = document.querySelector('.cart-header__body');
  const cartIcon = cart.querySelector('.shopping-cart');
  const cartQuantity = cart.querySelector('span');

  const cartProduct = document.querySelector(
    `[data-cart-pid="${productId}"]`
  );
  const cartList = document.querySelector('.cart-list');

  // Add
  if (productAdd) {
    if (cartQuantity) {
      // eslint-disable-next-line no-plusplus
      cartQuantity.innerHTML = ++cartQuantity.innerHTML; // number at the cart
      cartQuantity.classList.add('shopping-cart__body');
    } else {
      cartIcon.insertAdjacentHTML(
        'beforeend',
        `<span class="shopping-cart__body" >1</span>`
      );
    }
  }
  if (!cartProduct) {
    const product = document.querySelector(`[data-pid="${productId}"]`);
    const cartBtn = document.querySelector('.cart-list__btn');

    const cartProductImage =
      product.querySelector('.products__image').innerHTML;

    const cartProductTitle =
      product.querySelector('.products__name').innerHTML;

    const cartProductPrice =
      product.querySelector('.products__price').innerHTML;

```

```

const cartProductContent = `
  <a href="#" class="cart-list__image">${cartProductImage}</a>
  <div class="cart-list__body">
    <a href="#" class="cart-list__title" name="title">${cartProductTitle}</a>
    <div class="cart-list__quantity">
      Quantity:
      <span name="quantity">1</span>
    </div>
    <div class="cart-list__price" name="price">${cartProductPrice}</div>
    <button class="cart-list__remove">Remove <span> X </span></button>
  </div>
`;
cartBtn.classList.add('cart-list__btn-active');
cartList.insertAdjacentHTML(
  'beforeend',
  `
  <li data-cart-pid="${productId}" class="cart-list__item">${cartProductContent}</li>`
);
} else {
const cartProductQuantity = cartProduct.querySelector(
  '.cart-list__quantity span'
);

// eslint-disable-next-line no-plusplus
cartProductQuantity.innerHTML = ++cartProductQuantity.innerHTML;
}
// after all
productButton.classList.remove('_hold');
} else {
const cartProductQuantity = cartProduct.querySelector(
  '.cart-list__quantity span'
);
// eslint-disable-next-line no-plusplus
cartProductQuantity.innerHTML = --cartProductQuantity.innerHTML;

// eslint-disable-next-line radix
if (!parseInt(cartProductQuantity.innerHTML)) {
  cartProduct.remove();
}

// eslint-disable-next-line no-plusplus
const cartQuantityValue = --cartQuantity.innerHTML;

if (cartQuantityValue) {
  cartQuantity.innerHTML = cartQuantityValue;
} else {
  cartQuantity.remove();
  cartForm.classList.remove('_active');
}
}
}

```

```
}  
};
```

```
function getUsername(userName) {  
  let username;  
  $.ajax({  
    url: './assets/php/catalog/get-user.php',  
    method: 'post',  
    async: false,  
    data: { userName },  
    success: (response) => {  
      try {  
        if (response) {  
          const responseFromServer = JSON.parse(response);  
  
          if (responseFromServer.status === 'success') {  
            username = responseFromServer.username;  
          }  
          if (responseFromServer.status === 'error') {  
            const cart = document.querySelector('.header__shopping-cart');  
            console.log('nobody');  
            cart.style.display = 'none';  
          }  
        }  
      } catch (error) {  
        const cart = document.querySelector('.header__shopping-cart');  
        const add = document.querySelectorAll('.products__btn-add');  
        const remove = document.querySelectorAll('.products__btn-remove');  
  
        cart.style.display = 'none';  
  
        add.forEach((btn) => {  
          btn.style.display = 'none';  
        });  
  
        remove.forEach((btn) => {  
          btn.style.display = 'none';  
        });  
  
        remove.style.display = 'none';  
      }  
    },  
  });  
  return username;  
}
```

ВІДГУК

Керівника економічного розділу

на кваліфікаційну роботу бакалавра на тему:

«Розробка веборієнтованого додатку з продажу вінтажних велосипедів

за допомогою JavaScript та PHP»

Студентів групи 122-20-2 Кондрат Анни Дмитрівни

Керівник економічного розділу доц. каф. ПЕП та ПУ, к.е.н	Л.В. Касьяненко
--	------------------------

ПЕРЕЛІК ФАЙЛІВ НА ДИСКУ

Ім'я файлу	Опис
Пояснювальні документи	
Кондрат.docx	Пояснювальна записка до кваліфікаційної роботи. Документ Word.
Кондрат.pdf	Пояснювальна записка до кваліфікаційної роботи у форматі PDF
Програма	
Кондрат.zip	Архів. Містить коди програми і скомпільовану програму
Презентація	
Кондрат.ppt	Презентація кваліфікаційної роботи