

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ**

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Розробка web-застосунку для кастомізації і продажу
тенісних ракеток»

на здобуття освітнього ступеня бакалавра
зі спеціальності 121 Інженерія програмного забезпечення
освітньо-професійної програми «Інженерія програмного забезпечення»

*Кваліфікаційна робота містить результати власних досліджень. Використання
ідей, результатів і текстів інших авторів мають посилання
на відповідне джерело*

_____ Аліна МЕЛЬНИЧУК
(підпис)

Виконала: здобувачка вищої освіти групи ПД-41

_____ Аліна МЕЛЬНИЧУК

Керівник: _____ Ірина ЗАМРІЙ
д.т.н., професор

Рецензент: _____

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ**

Навчально-науковий інститут інформаційних технологій

Кафедра Інженерії програмного забезпечення

Ступінь вищої освіти Бакалавр

Спеціальність 121 Інженерія програмного забезпечення

Освітньо-професійна програма «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри

Інженерії програмного забезпечення

_____ Ірина ЗАМРІЙ

« ____ » _____ 2025 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

_____ Мельничук Аліні Юріївні

1. Тема кваліфікаційної роботи: «Розробка web-застосунку для кастомізації і продажу тенісних ракеток»

керівник кваліфікаційної роботи д.т.н., професор Ірина ЗАМРІЙ,
затверджені наказом Державного університету інформаційно-комунікаційних технологій від «24» лютого 2025 р. № 56.

2. Строк подання кваліфікаційної роботи «02» червня 2025 р.

3. Вихідні дані до кваліфікаційної роботи: теоретичні відомості про принципи електронної комерції та персоналізації товарів, аналіз сучасних підходів до кастомізації спортивного інвентарю, технічна документація щодо використання інструментів для розробки web-застосунків.

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

1. Аналіз аналогів та підходів до онлайн-кастомізації товарів у сфері електронної комерції.
2. Визначення вимог та проектування web-застосунку для кастомізації та продажу тенісних ракеток.
3. Реалізація клієнтської частини застосунку з використанням HTML, CSS, JavaScript та Tailwind CSS.

4. Використання MockAPI для збереження кастомізованих конфігурацій і оформлення замовлень.
5. Тестування функціональності та адаптивності застосунку.
6. Розробка демонстраційних матеріалів та оформлення проєкту.

5. Перелік графічного матеріалу: *презентація*

1. Аналіз аналогів.
2. Вимоги до програмного забезпечення.
3. Програмні засоби реалізації.
4. Архітектура web-застосунку
5. Діаграма варіантів використання.
6. Карта web-застосунку
7. Екранні форми.
8. Апробація результатів дослідження

6. Дата видачі завдання «25» лютого 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Підбір та аналіз науково-технічної літератури	25.02.-04.03.2025	
2	Аналіз та дослідження існуючих аналогів	05.03-13.03.2025	
3	Огляд методик реалізації web-додатків для електронної комерції та кастомізації товарів.	14.03–25.03.2025	
4	Проектування застосунку для кастомізації та продажу тенісних ракеток	26.03–02.04.2025	
5	Програмна реалізація застосунку	03.04-19.04.2025	
6	Тестування застосунку	20.04–28.04.2025	
7	Оформлення роботи: вступ, висновки, реферат	29.04-11.05.2025	
8	Розробка демонстраційних матеріалів	12.05-18.05.2025	
9	Попередній захист роботи	19.05-30.05.2025	

Здобувачка вищої освіти

_____ (підпис)

Аліна МЕЛЬНИЧУК

Керівник
кваліфікаційної роботи

_____ (підпис)

Ірина ЗАМРІЙ

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня бакалавра: 53 стор., 2 табл., 29 рис., 25 джерел.

Мета роботи – спрощення процедури персоналізації та замовлення тенісних ракеток.

Об'єкт дослідження – процес онлайн-кастомізації та продажу спортивного інвентарю, зокрема тенісних ракеток.

Предмет дослідження – web-застосунок для кастомізації і продажу тенісних ракеток з інтерактивним інтерфейсом.

Короткий зміст роботи: В роботі проаналізовано предметну область електронної комерції у сфері спортивного інвентарю. Проведено огляд існуючих web-сервісів для кастомізації ракеток, визначено їх недоліки. Обґрунтовано вибір технологій для розробки: HTML, CSS, JavaScript, Tailwind, MockAPI, GitHub, Figma. Спроектовано і реалізовано web-застосунок для кастомізації тенісних ракеток з візуалізацією змін у реальному часі та можливістю оформлення замовлення. Проведено тестування розробленого додатку та оформлено демонстраційні матеріали.

Сферою використання web-застосунку є електронна комерція у спортивній галузі.

КЛЮЧОВІ СЛОВА: WEB-ЗАСТОСУНОК, КАСТОМІЗАЦІЯ, ТЕНІСНА РАКЕТКА, ЕЛЕКТРОННА КОМЕРЦІЯ, UX/UI, JAVASCRIPT, TAILWIND CSS, АДАПТИВНИЙ ДИЗАЙН, МОСКАPI

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	9
ВСТУП.....	10
1 АНАЛІЗ ЗАСТОСУНКУ ДЛЯ КАСТОМІЗАЦІЇ ТА ПРОДАЖУ ТЕНІСНИХ РАКЕТОК.....	12
1.1 Web-застосунок для кастомізації тенісних ракеток	12
1.2 Специфіка кастомізації спортивного інвентарю.....	14
1.3 Цільова аудиторія.....	15
1.4 Дослідження існуючих аналогів.....	15
1.4.1 Wilson Custom Tennis.....	16
1.4.2 RacketTier.....	17
1.4.3 Racquet Design	18
1.4.4 RAGE Custom	20
1.4.5 Висновки.....	22
1.5 Визначення вимог до застосунку	24
2 ЗАСОБИ РЕАЛІЗАЦІЇ	26
2.1 Середовище розробки.....	26
2.1.1 Visual Studio Code	26
2.2 Мови програмування	28
2.2.1 JavaScript.....	28
2.2.2 HTML та CSS.....	29
2.3 CSS-фреймворк Tailwind.....	30
2.4 База даних і API.....	30
2.5 Система контролю версій.....	31
2.6 Інструменти проектування інтерфейсу.....	31
3 РЕАЛІЗАЦІЯ WEB-ЗАСТОСУНКУ	32
3.1 Проектування архітектури застосунку	32
3.2. Основні сторінки	35
3.3 База даних	37
3.3.1 Структура ресурсу basket	37
3.3.2 Структура ресурсу tennis.....	39

3.3.3 Переваги використання MockAPI	41
3.3.4 Перспективи розширення.....	41
3.4 Структура проекту	41
3.5 Програмна реалізація web-застосунку	44
3.5.1 Кастомізація ракеток	44
3.5.2 Каталог ракеток.....	45
3.5.3 Оформлення замовлень	47
4 ДЕМОНСТРАЦІЯ РЕЗУЛЬТАТІВ ТА ТЕСТУВАННЯ	49
4.1 Демонстрація web-застосунку	49
4.2 Тестування	54
4.2.1 Тест-кейси заповнення форми	54
4.2.2 Тестування адаптивності під різні пристрої	56
4.2.3 Тестування збереження замовлень у MockAPI.....	59
ВИСНОВКИ.....	61
ПЕРЕЛІК ПОСИЛАНЬ	63
ДОДАТОК А. ДЕМОНСТРАЦІЙНІ МАТЕРІАЛ	66
ДОДАТОК Б. ЛІСТИНГИ ОСНОВНИХ МОДУЛІВ.....	73

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

API – Application Programming Interface

CSS – Cascading Style Sheets

HTML – HyperText Markup Language

JS – JavaScript

SPA – Single Page Application

UI – User Interface

UX – User Experience

ВСТУП

Актуальність: розвиток електронної комерції та зростання попиту на персоналізовані товари вимагають створення нових інструментів, що забезпечують зручний та ефективний процес онлайн-покупок. У спортивній індустрії, зокрема у сфері продажу тенісного інвентарю, важливим аспектом стає можливість налаштування виробу під індивідуальні потреби користувача. Тенісна ракетка — це не лише спортивний атрибут, але й ключовий інструмент, від якого залежать комфорт гри, техніка та навіть ризик отримання травм. Персоналізація інвентарю дозволяє спортсменам та аматорам налаштувати параметри (вага, баланс, розмір ручки, колір тощо) відповідно до своїх фізичних характеристик і стилю гри.

З розвитком web-технологій виникає можливість створення інтерактивних інструментів, що забезпечують користувачу динамічну візуалізацію та зручне налаштування товару. Одним із найефективніших рішень є web-застосунок, що дозволяє здійснювати кастомізацію ракетки у режимі реального часу безпосередньо в браузері. Такий підхід дозволяє покупцю краще розуміти, як виглядатиме готовий продукт, а також формує емоційний зв'язок з брендом завдяки відчуттю унікальності придбаного товару.

Web-застосунок для кастомізації тенісних ракеток постає як комплексний інструмент, що поєднує функціонал онлайн-магазину з індивідуальним налаштуванням. Він забезпечує доступ до ключових елементів конфігурації інвентарю, включаючи інтерактивну зміну параметрів, попередній перегляд, збереження налаштувань та оформлення замовлення. Крім того, застосунок адаптований до мобільних пристроїв, що робить його зручним для широкого кола користувачів.

Об'єктом дослідження є процес онлайн-кастомізації та продажу спортивного інвентарю, зокрема тенісних ракеток.

Предметом дослідження є web-застосунок для кастомізації і продажу тенісних ракеток з інтерактивним інтерфейсом. Метою роботи є спрощення процедури персоналізації та замовлення тенісних ракеток.

Для реалізації поставленої мети першочергово було проведено аналіз існуючих рішень на ринку онлайн-кастомізації спортивного інвентарю для формування вимог до майбутнього програмного продукту. Наступним кроком став вибір стеку технологій: HTML, CSS, JavaScript, Tailwind CSS для клієнтської частини, MockAPI для імітації бекенду, Figma для прототипування інтерфейсу, GitHub для контролю версій. Під час розробки було протестовано взаємодію компонентів, адаптивність інтерфейсу та швидкодію системи.

Наукова новизна роботи полягає в реалізації web-застосунку з функціональністю динамічної кастомізації тенісної ракетки в режимі реального часу, інтерактивним переглядом змін, адаптивною версткою та підтримкою збереження конфігурацій. Це рішення поєднує UX-дизайн, web-розробку та моделі поведінки користувача в одному продукті.

Практична значущість полягає у створенні інструменту, який може бути використаний в електронній комерції для підвищення залученості клієнтів, автоматизації процесу налаштування товару та покращення користувацького досвіду. Реалізований застосунок може бути адаптований і до інших категорій товарів, що потребують персоналізації.

1 АНАЛІЗ ЗАСТОСУНКУ ДЛЯ КАСТОМІЗАЦІЇ ТА ПРОДАЖУ ТЕНІСНИХ РАКЕТОК

1.1 Web-застосунок для кастомізації тенісних ракеток

Web-застосунок для кастомізації тенісних ракеток — це інтерактивна онлайн-платформа, що дозволяє користувачам налаштовувати спортивний інвентар відповідно до індивідуальних параметрів, таких як вага, баланс, розмір ручки, колір тощо, та здійснювати його покупку. Такий підхід задовольняє потребу сучасного споживача в персоналізації та гнучкості під час вибору товарів.

Основна мета web-застосунку полягає в автоматизації процесу підбору ракетки та підвищенні задоволеності користувача завдяки можливості налаштування та візуального перегляду конфігурації в реальному часі. Це особливо актуально для спортсменів, які потребують точного налаштування інвентарю під свій стиль гри, фізичні характеристики та естетичні вподобання.

Традиційний підхід до придбання ракеток у фізичних магазинах або онлайн передбачає обмежений вибір моделей. Користувачам зазвичай не пропонується можливість кастомізації інвентарю до його покупки. У професійних спортсменів є потреба в максимально точному налаштуванні ракетки під себе, але навіть аматори бажають мати можливість налаштувати колір, гравіювання, тип ручки тощо.

Дослідження в галузі маркетингу свідчать, що відчуття унікальності продукту, яке виникає при кастомізації, суттєво впливає на задоволення клієнта та формування емоційного зв'язку з брендом. Особливо це актуально для спортивного інвентарю, де персоналізація надає додаткову цінність не лише як функція, а і як складова стилю гри користувача [1].

Згідно з дослідженням Bouchrika et al. (2022)[2], персоналізація товарів в e-commerce значно підвищує лояльність клієнтів, рівень задоволеності покупкою та ймовірність повторного замовлення. У спортивній сфері персоналізація має додаткову цінність — вона прямо впливає на продуктивність користувача (Sharma

& Kumari, 2021)[3].

Основні компоненти web-застосунку для кастомізації включають:

- інтерфейс конфігурацій у виборі технічних параметрів ракетки (вага, баланс, розмір головки тощо);
- інтерактивна візуалізація - виведення кастомізованої ракетки в режимі реального часу;
- система збереження з можливістю зберігати обрані конфігурації на тоск-сервері;
- адаптивний інтерфейс, розробка оптимізація під мобільні пристрої;
- реалізація елементів електронної комерції таких як кошик і оформлення замовлення.

Ціль такого web-застосунку — зробити процес вибору спортивного інвентарю зручним, гнучким та максимально наближеним до індивідуальних потреб покупця.

Переваги:

- індивідуальний підбір інвентарю та персоналізація кожного параметра ракетки;
- візуальний контроль, аби користувач бачив, як змінюється зовнішній вигляд ракетки;
- зручність використання завдяки інтуїтивному інтерфейсу, який доступний з будь-якого пристрою;
- швидкість покупки, адже кастомізація і покупка відбуваються без необхідності відвідувати магазин.

Недоліки:

- складність реалізації 3D або псевдо-3D візуалізації у браузері;
- необхідність стабільного підключення до інтернету;
- обмеження ресурсів браузера для складних візуальних рендерів;
- потреба у підтримці широкої сумісності з різними пристроями.

У результаті аналізу стає очевидним, що web-застосунок для кастомізації тенісних ракеток є сучасним інструментом, який відповідає потребам як ринку

спортивного інвентарю, так і кінцевих споживачів. Впровадження такого інструменту дозволяє не лише покращити користувацький досвід, але й підвищити конкурентоспроможність бізнесу в галузі електронної комерції.

1.2 Специфіка кастомізації спортивного інвентарю

Кастомізація спортивного інвентарю, зокрема тенісних ракеток, є процесом індивідуального налаштування виробу відповідно до потреб конкретного користувача. Вона має як функціональне, так і емоційне значення - дозволяє досягти кращої продуктивності під час гри та водночас формує унікальну ідентичність гравця.

Основні параметри кастомізації тенісної ракетки:

- вага — впливає на швидкість замаху та контроль м'яча;
- баланс — співвідношення ваги між головою та ручкою;
- розмір головки — визначає площу удару (sweet spot);
- розмір ручки — критичний для зручності тримання;
- тип і натяг струн — впливають на віддачу та чутливість при ударі;
- колір і дизайн — мають естетичну функцію та індивідуалізують інвентар.

Реалізація таких змін у web-застосунку повинна бути максимально візуальною та інтуїтивною. Користувач має бачити результат кожної зміни у режимі реального часу без потреби перезавантаження сторінки.

Особливості розробки кастомізатора полягають у необхідності:

- підтримки динамічного оновлення інтерфейсу (JavaScript);
- використання легковагових методів візуалізації (Tailwind CSS);
- оптимізації під мобільні пристрої;
- мінімізації навантаження на браузер.

Таким чином, кастомізація — це не просто додатковий функціонал, а ключовий елемент, який формує унікальну цінність продукту для кінцевого користувача.

1.3 Цільова аудиторія

Web-застосунок орієнтований на кілька основних груп користувачів, які мають різні потреби, але об'єднані спільною метою — отримати ракетку, яка повністю відповідає їхнім очікуванням:

- професійні спортсмени — потребують точного налаштування параметрів для покращення результатів на змаганнях;
- аматори — зацікавлені у візуальній персоналізації та комфорті під час гри;
- тренери та клуби — замовляють інвентар для команд, важливим є одночасне налаштування багатьох одиниць;
- інтернет-магазини спортивного інвентарю — зацікавлені у впровадженні кастомайзера як додаткової конкурентної переваги;
- підлітки та молодь — орієнтовані на індивідуальний стиль, колір, гравіювання тощо.

Визначення цільової аудиторії допомагає сформулювати відповідні вимоги до інтерфейсу: простота, адаптивність, швидкодія, можливість збереження конфігурацій та подальшого редагування.

1.4 Дослідження існуючих аналогів

Для ефективної розробки web-застосунку з кастомізації тенісних ракеток важливо провести глибокий аналіз конкурентних рішень, які вже представлені на ринку. Це дозволяє:

- виявити сильні та слабкі сторони існуючих рішень;
- уникнути повторення помилок;
- запозичити кращі практики UX/UI та функціональності;
- знайти незаповнені ринкові ніші.

У межах дослідження було проаналізовано чотири актуальні сервіси, які частково або повністю реалізують функцію кастомізації тенісного інвентарю: Wilson Custom Tennis, RacketTier, Racquet Design і RAGE Custom.

1.4.1 Wilson Custom Tennis

Wilson[4] — один із найбільших і найвідоміших брендів у світі тенісу. Їхня платформа Wilson Custom Tennis дозволяє користувачам налаштовувати вигляд ракетки безпосередньо на сайті компанії.

Основні можливості:

- вибір кольору головки, ободу, логотипу та грипу;
- додавання персонального гравіювання;
- попередній візуальний перегляд змін;
- підтримка мобільних пристроїв;
- вибір з-поміж обмеженого числа базових моделей.

Переваги:

- відома торговельна марка підвищує довіру користувачів;
- привабливий та простий інтерфейс;
- візуалізація змін у реальному часі.

Недоліки:

- відсутність можливості змінювати технічні характеристики (баланс, вага, розмір ручки);
- відсутня підтримка конфігурацій користувача;
- фокус виключно на естетиці, а не функціональності.

Висновок. Wilson пропонує лише візуальну кастомізацію, що добре для аматорів, але не підходить для професіоналів, які потребують технічної персоналізації.

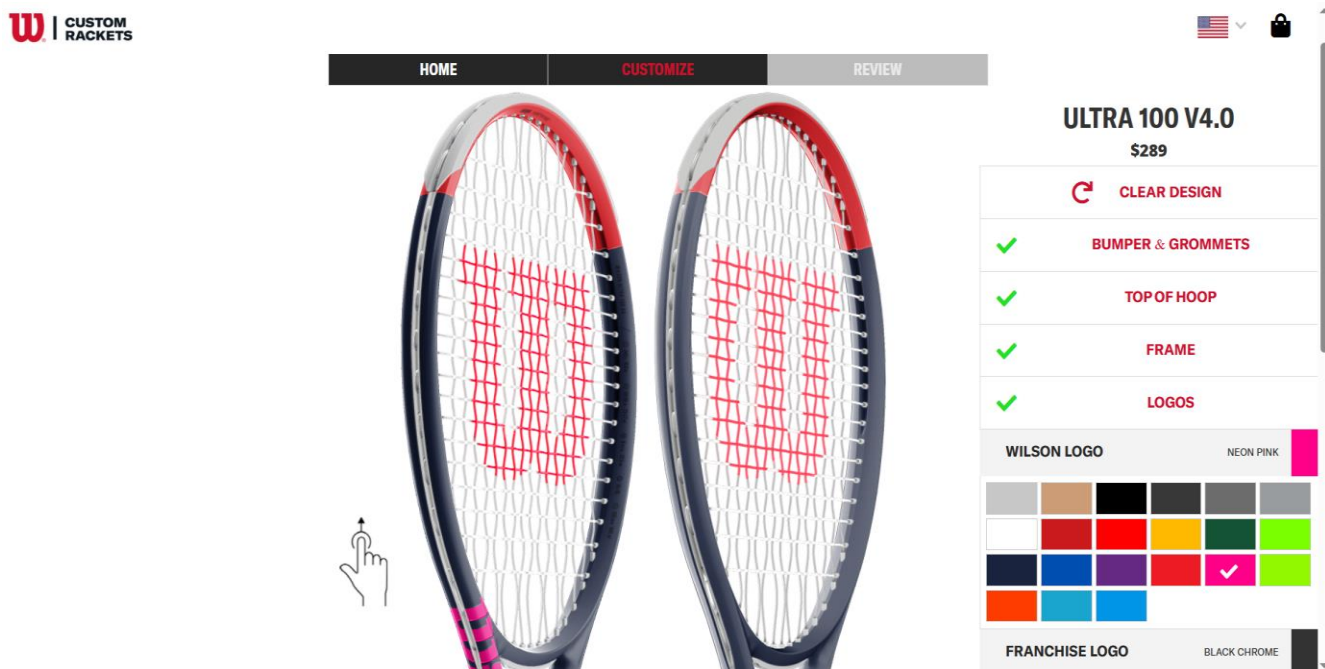


Рис. 1.1 Приклад кастомізації у web-застосунку Wilson Custom Tennis

1.4.2 RacketTier

RacketTier[5] — сервіс, що надає послуги з підбору та кастомізації тенісних ракеток на основі даних про гру користувача. Сервіс включає демо-сесії та аналітику для визначення оптимальної ракетки.

Для початку кастомізації потрібно заповнити Google – форму зі своїми персональними даними, особистими вподобаннями та побажаннями щодо майбутньої ракетки. Після цього потрібно довго чекати на відповідь менеджера для уточнення деталей щодо нашої майбутньої ракетки.

Функціонал:

- заповнення анкети з інформацією про стиль гри;
- проведення демо-сесій з тестуванням ракеток;
- підбір оптимального варіанту менеджером;
- немає онлайн-конфігуратора чи візуального налаштування.

Переваги:

- глибокий аналітичний підхід до підбору;
- проведення демо-сесій для практичного тестування;

- індивідуальні рекомендації.

Недоліки:

- кастомізація виконується офлайн (через менеджера);
- немає інтерактивного web-інструменту;
- затримки в обробці запиту;
- непридатний для швидкого або імпульсивного замовлення.

Висновок. Сильний аналітичний сервіс, але без візуального інструмента кастомізації, що знижує UX та масштабованість.



Рис. 1.2 Початковий екран у web-застосунку RacketTier

1.4.3 Racquet Design

Racquet Design[6] спеціалізується на створенні унікальних дизайнів для тенісних ракеток, включаючи кастомне фарбування та налаштування ваги, балансу та інших параметрів. Проте для початку роботи потрібно одразу контактувати з менеджером, що є великим недоліком, адже кастомізація «в моменті» є набагато комфортнішою для користувача.

Можливості:

- кастомне фарбування, гравіювання;
- налаштування ваги та балансу;
- створення унікального візуального стилю;

- орієнтація на ручну роботу.

Переваги:

- повна індивідуалізація;
- високий рівень кастомного сервісу;
- підходить для професійних гравців та команд.

Недоліки:

- відсутність онлайн-інтерфейсу;
- необхідність прямих контактів;
- довготривалий цикл виконання;
- складно відтворити/масштабувати такий сервіс.

Висновок. Racquet Design — якісний преміум-сервіс, однак надто залежний від ручної роботи, що суперечить концепції автоматизованого онлайн-застосунку.



From passion for tennis and love for design, we create unique and individual racquet paint jobs. The designs are made guaranteeing to be specially tailored for your look. Created to embed the play style and the personality of the player.

Our services:

Custom racquet painting
 Customising weight, balance and swing weight
 Stringing
 Grips change (base and overgrip)
 Pairing racquets

Your racquet can be the only one in the world! **Contact us!**

Рис. 1.3 Приклад екранної форми web-застосунку Racquet Design

1.4.4 RAGE Custom

RAGE Custom[7] пропонує повну кастомізацію ракеток для пляжного тенісу, дозволяючи користувачам налаштовувати як технічні характеристики, так і зовнішній вигляд ракетки. Сервіс також використовує екологічно чисті матеріали та процеси виробництва.

Функціонал:

- вибір форми, ваги, балансу, кольору;
- налаштування покриття, логотипу, дизайну ручки;
- використання екологічних матеріалів;
- 3D-візуалізація.

Переваги:

- розширена кастомізація (і технічна, і візуальна);
- гарний UX, сучасний інтерфейс;
- увага до екологічності.

Недоліки:

- фокус на пляжний теніс;
- повільніша обробка замовлення (через високу гнучкість);
- складність адаптації під класичний теніс.

Висновок. RAGE Custom демонструє сучасний рівень реалізації кастомізації, але його концепцію важко адаптувати до ринку класичного тенісу без суттєвих змін.

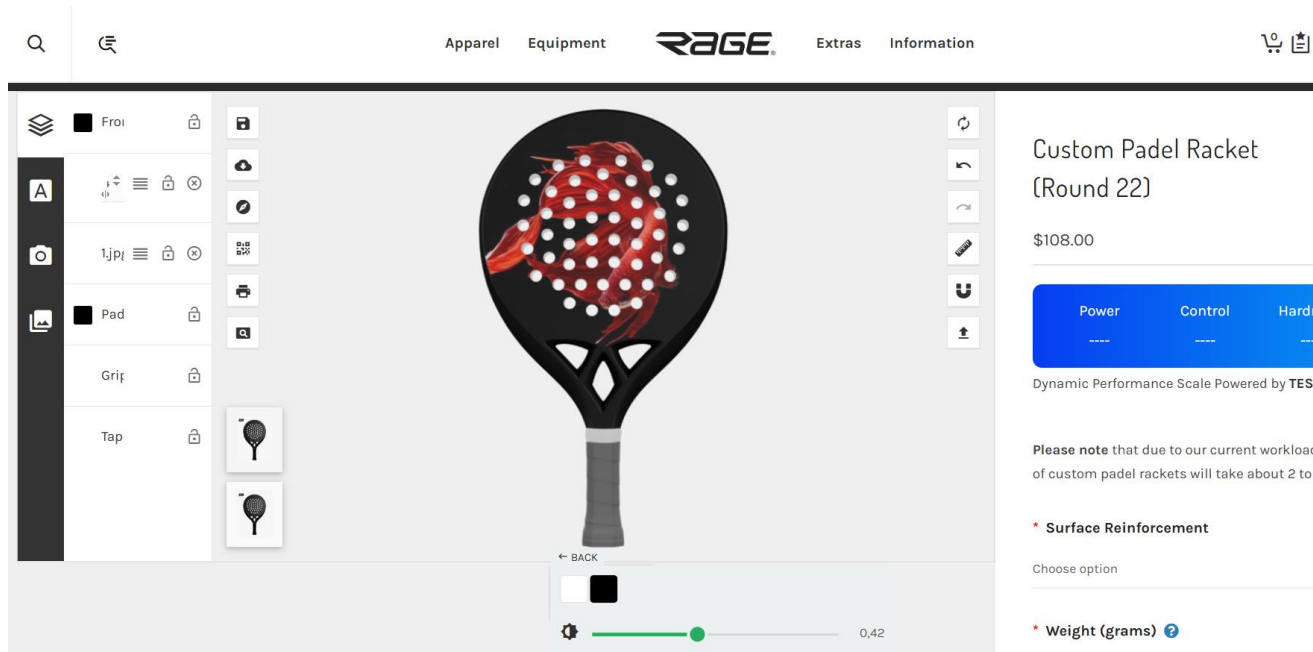


Рис. 1.4 Приклад кастомізації у web-застосунку RAGE Custom

*** Surface Reinforcement**

12K Carbon - Advanced (+ \$300.00) ▼ + \$300.00

For greater **Power**, increase weight and set Inner Core to Hard. For more **Control**, lessen weight and set Inner Core to Medium. Keep defaults for Balanced performance.

*** Weight (grams) ?**

360

340 350 360 370 380

*** Inner Core** *** Balance**

Dense (High-Grade EVA-E) ? Low (Around 260mm) ▼ ?

String Color (Optional) **Finish (Optional)**

Black ▼ Matte/Dull ▼

Рис. 1.5 Приклад кастомізації ракетки у RAGE Custom

1.4.5 Висновки

Зведені результати аналізу характеристик розглянутих додатків наведено у таблиці 1.1.

Таблиця 1.1

Зведені результати аналізу характеристик додатків для кастомізації тенісних ракеток

Компанія	Онлайн-кастомізація	Технічна кастомізація	Візуальна кастомізація	Орієнтація на класичний теніс	Орієнтація на пляжний теніс	Каталог готових товарів	Унікальні особливості
RacketTie r	-	+	-	+	-	-	Використання даних для персоналізованого підбору ракетки
Racquet Design	-	+	+	+	-	-	Індивідуальний підхід до дизайну та налаштування параметрів
Wilson Custom Tennis	+	-	+	+	-	-	Онлайн-інструмент для візуальної кастомізації
RAGE Custom	+	+	+	-	+	+	Повна кастомізація ракетки та використання екологічних матеріалів

Продовження таблиці 1.1

Зведені результати аналізу характеристик додатків для кастомізації тенісних ракеток

Компанія	Онлайн-кастомізація	Технічна кастомізація	Візуальна кастомізація	Орієнтація на класичний теніс	Орієнтація на пляжний теніс	Каталог готових товарів	Унікальні особливості
Tennis Lab	+	+	+	+	-	+	Кастомізація ракетки з онлайн відображенням, наявність каталогу готових ракеток

Отже, аналіз конкурентів у сфері кастомізації тенісних ракеток демонструє різноманітність підходів та рішень, що існують на ринку. Кожен з розглянутих сервісів має свої унікальні особливості, які можуть бути враховані при розробці власного продукту.

RacketTier зосереджується на технічній кастомізації, пропонуючи індивідуальні налаштування ваги, балансу та інших параметрів ракетки. Однак, відсутність онлайн-інструментів для візуальної кастомізації та необхідність фізичної присутності для демо-сесій можуть обмежувати доступність сервісу для ширшої аудиторії.

Racquet Design пропонує унікальний підхід до дизайну ракеток, включаючи кастомне фарбування та налаштування фізичних характеристик. Проте, відсутність

онлайн-платформи для кастомізації та необхідність прямого контакту з сервісом можуть ускладнювати процес для користувачів.

Wilson Custom Tennis надає зручний онлайн-інструмент для візуальної кастомізації, дозволяючи користувачам обирати кольори, додавати текст та налаштовувати інші елементи дизайну. Однак, можливості технічної кастомізації обмежені, і відсутні персоналізовані консультації або демо-сесії.

RAGE Custom пропонує повну кастомізацію ракеток для пляжного тенісу, дозволяючи користувачам налаштовувати як технічні характеристики, так і зовнішній вигляд ракетки. Сервіс також використовує екологічно чисті матеріали та процеси виробництва. Проте, орієнтація на пляжний теніс може обмежувати застосування для класичного тенісу, а процес кастомізації може займати більше часу через індивідуальний підхід.

Загалом, аналіз показує, що існує потреба в сервісі, який поєднує переваги всіх розглянутих конкурентів: зручність онлайн-кастомізації, можливість технічного налаштування та індивідуальний підхід до дизайну. У звіті Deloitte Digital «Personalizing Growth:[8] It's a value exchange between brands and customers» зазначено, що лише 22% компаній у сфері e-commerce надають «глибоку персоналізацію» клієнтам. Це свідчить про те, що більшість брендів ще не реалізували персоналізацію на високому рівні, незважаючи на її визнану важливість для підвищення лояльності клієнтів і зростання доходів. Розробка такого продукту може зайняти унікальну нішу на ринку та задовольнити потреби широкого кола користувачів.

1.5 Визначення вимог до застосунку

На основі аналізу предметної галузі, специфіки кастомізації спортивного інвентарю, визначення цільової аудиторії та вивчення існуючих аналогів сформульовано ключові функції, які повинні бути реалізовані у майбутньому додатку:

- можливість вибору ключових технічних характеристик ракетки (вага, баланс, розмір ручки, площа головки, колір);

- інтерактивна візуалізація змін конфігурації в реальному часі;

- зміна зовнішнього вигляду - вибір кольору ракетки;

- наявність каталогу для вибору готових виробів;

- візуальні підказки та поп-ап повідомлення;

- додавання кастомізованої ракетки до кошика та оформлення замовлення;

- навантаження - застосунок повинен працювати стабільно при мінімальному використанні ресурсів браузера;

- адаптивний інтерфейс, що коректно працює на мобільних пристроях(у сучасних умовах особливу увагу необхідно приділяти оптимізації інтерфейсу для мобільних пристроїв. Це стосується не лише адаптації візуального оформлення, а й структури інтерфейсу та логіки взаємодії, з урахуванням особливостей сенсорного введення, невеликих екранів і змінної орієнтації дисплею. Саме ці аспекти визначають якість мобільного користувацького досвіду)[9];

- час відгуку - інтерфейс має реагувати на дії користувача протягом <2 секунд(одним із критичних факторів успішності будь-якого web-застосунку є висока швидкодія та чітка реакція інтерфейсу на дії користувача. Згідно з результатами академічних досліджень, затримки понад кілька секунд можуть негативно впливати на лояльність користувачів, знижувати ймовірність завершення покупки та загальну оцінку досвіду)[10];

- взаємодія з тоск-сервером для збереження та отримання даних.

Реалізація зазначених вимог дозволить створити повноцінний, адаптивний і сучасний web-застосунок, що забезпечує зручний процес кастомізації тенісних ракеток і підвищує ефективність онлайн-продажу.

2 ЗАСОБИ РЕАЛІЗАЦІЇ

2.1 Середовище розробки

У процесі створення web-застосунку для кастомізації та продажу тенісних ракеток було критично важливо обрати інструменти, що забезпечують високу продуктивність, швидку інтерактивність, адаптивний дизайн, простоту інтеграції з серверною частиною, підтримку зберігання кастомних конфігурацій та сучасну UX/UI-архітектуру.

2.1.1 Visual Studio Code

Середовище розробки (IDE) є ключовим компонентом у процесі створення програмного забезпечення, оскільки визначає ефективність написання, тестування та налагодження коду. Для реалізації web-застосунку з кастомізації та продажу тенісних ракеток було проаналізовано кілька популярних середовищ, серед яких Visual Studio Code (VS Code), WebStorm та Sublime Text. Вибір остаточного інструменту ґрунтувався на критеріях зручності, продуктивності, підтримки сучасних технологій, інтеграції з додатковими сервісами та спільнотою користувачів.

Visual Studio Code[11] — це кросплатформенний редактор коду, розроблений компанією Microsoft, який стрімко набув популярності серед web-розробників. Завдяки відкритому коду, безкоштовній ліцензії, великій кількості доступних розширень і низьким системним вимогам, він став провідним інструментом у сфері фронтенд-розробки.

Основні переваги Visual Studio Code:

- підтримка сучасних мов програмування — таких як JavaScript, TypeScript, HTML, CSS, що є критично важливим для розробки web-застосунків;

- інтеграція з Git та GitHub — забезпечує швидке збереження змін, гілкування, коментування та злиття коду безпосередньо з середовища розробки;
- live Server — розширення, що дає змогу в режимі реального часу переглядати зміни в інтерфейсі без перезавантаження вручну;
- підтримка Tailwind CSS — завдяки плагіну користувач може отримувати підказки класів Tailwind, переглядати документацію і стилізувати компоненти без потреби у зовнішніх інструментах;
- можливість використання MockAPI — для емуляції запитів до серверної частини та тестування інтерактивності застосунку;
- плагіни Prettier та ESLint — забезпечують автозаміни стилю коду та дотримання правил написання, що суттєво підвищує читабельність та підтримуваність коду;
- вбудований термінал — дозволяє запускати команди `npm`, перевіряти статус репозиторію та виконувати скрипти без потреби відкривати окрему консоль.

Окрім технічних можливостей, важливим фактором на користь VS Code є велика спільнота користувачів та активна підтримка з боку розробників, що гарантує швидке оновлення функціоналу, доступ до документації та рішень найпоширеніших проблем.

WebStorm[12] — це потужне середовище розробки від компанії JetBrains, орієнтоване насамперед на JavaScript-розробку. Воно пропонує розширений функціонал: глибокий аналіз коду, автодоповнення з урахуванням контексту, інтеграцію з найпопулярнішими фреймворками та бібліотеками, зокрема React, Angular, Vue.js. Незважаючи на значні переваги, WebStorm є платним продуктом, що робить його менш привабливим варіантом для студентських і невеликих проєктів. Також його використання вимагає більше системних ресурсів, що може бути обмежувальним фактором.

Sublime Text[13] — легкий текстовий редактор з високою швидкістю роботи та зручним мінімалістичним інтерфейсом. Хоча він підтримує базове автодоповнення та плагіни, можливості інтеграції з сучасними інструментами, такими як Git, системи збору, сервери попереднього перегляду, є обмеженими без

додаткової конфігурації. Через це Sublime Text не був обраний як основне середовище розробки.

У результаті порівняльного аналізу було прийнято рішення використовувати Visual Studio Code як основне середовище розробки в межах цього проєкту. Його безкоштовність, гнучкість, широка підтримка розширень, адаптованість до сучасної web-розробки, підтримка інтеграції з Tailwind CSS та MockAPI, а також наявність інтерактивного терміналу роблять його ідеальним інструментом для реалізації проєктів середнього рівня складності в умовах обмеженого бюджету та часу.

2.2 Мови програмування

2.2.1 JavaScript

Однією з основних мов програмування для реалізації клієнтської частини web-застосунку було обрано JavaScript[14] — високорівневу, інтерпретовану мову, яка підтримується всіма сучасними web-браузерами та є ключовим інструментом для створення інтерактивних інтерфейсів.

Завдяки JavaScript вдалося реалізувати наступний функціонал:

- динамічну взаємодію з користувачем - зміна параметрів тенісної ракетки (наприклад, кольору, типу ручки чи ваги) у режимі реального часу без перезавантаження сторінки;
- обробку подій: взаємодію з кнопками, випадаючими списками, формами замовлення;
- інтеграцію з MockAPI - надсилання та отримання даних через REST API для збереження кастомізованих конфігурацій ракеток.

Переваги використання JavaScript у межах цього проєкту:

- повна підтримка усіма браузерами — не потребує встановлення додаткових розширень чи плагінів;
- гнучкий синтаксис і низький поріг входу, що дозволяє швидко розробляти функціонал;

- наявність широкого вибору бібліотек та інструментів;
- підтримка кількох парадигм програмування — об'єктно-орієнтованої, функціональної та подієво-орієнтованої.

Хоча мова TypeScript, яка є надбудовою над JavaScript, пропонує статичну типізацію, розширену систему типів і краще автодоповнення, у даному проєкті було вирішено використовувати саме JavaScript. Основними причинами стали:

- швидкість розробки, що є критично важливою в рамках обмеженого терміну виконання дипломного проєкту;
- краща сумісність із сторонніми REST-сервісами, зокрема MockAPI;
- зменшення обсягу конфігурації проєкту, що дозволяє зосередитись на основному функціоналі.

Таким чином, JavaScript було обрано як основну мову для реалізації логіки взаємодії користувача з інтерфейсом та взаємодії з серверною частиною.

2.2.2 HTML та CSS

У складі фронтенд-розробки web-застосунку основоположними технологіями є HTML та CSS — мови, що відповідають за структурування та візуальне оформлення web-сторінки.

HTML (HyperText Markup Language)[15] використовується для створення логічної структури інтерфейсу. У межах реалізації застосунку за допомогою HTML були створені наступні компоненти:

- секції для вибору параметрів кастомізації ракетки;
- форми для введення користувацьких даних та оформлення замовлення;
- навігаційні елементи (меню, кнопки, посилання);
- семантична розмітка для поліпшення доступності та SEO-оптимізації.

CSS (Cascading Style Sheets) забезпечує стилізацію HTML-структури та формує зовнішній вигляд сторінки. Основні напрямки використання CSS у проєкті:

- адаптивний дизайн — застосовано медіа-запити (@media) для забезпечення коректного відображення інтерфейсу на різних типах пристроїв (десктоп, планшет, смартфон);

- гнучке компонування блоків — з використанням Flexbox та Grid Layout;
- кастомне оформлення елементів — кнопки, випадаючі списки, колірні схеми та шрифти;
- забезпечення консистентного стилю інтерфейсу — через уніфіковані змінні кольорів, відступів, розмірів шрифтів тощо.

У поєднанні HTML і CSS створили базу для побудови зручного, інтуїтивно зрозумілого та візуально привабливого інтерфейсу. Для пришвидшення стилізації додатково використовувався фреймворк Tailwind CSS, що дозволив значно скоротити кількість кастомного CSS-коду завдяки утилітарному підходу до стилізації.

2.3 CSS-фреймворк Tailwind

Tailwind CSS[16] — сучасний фреймворк утилітарних класів, який дозволяє створювати адаптивні web-інтерфейси без написання окремих CSS-файлів. Він забезпечує гнучкість у стилізації, підтримує dark mode, breakpoints, а також дозволяє використовувати кастомні теми.

Переваги Tailwind у проєкті:

- прискорена розробка завдяки використанню готових класів;
- адаптивність до мобільних пристроїв;
- зниження обсягу CSS-коду через tree-shaking.

Tailwind активно використовується у стартапах та SaaS-проєктах через високу продуктивність розробки.

2.4 База даних і API

MockAPI — це онлайн-сервіс, що дозволяє створювати фейкові REST API для тестування клієнтських застосунків. Він підтримує всі основні HTTP-методи (GET, POST, PUT, DELETE), а також надає web-інтерфейс для перегляду і редагування даних.

У розробленому застосунку MockAPI використовується для:

- збереження конфігурацій користувачів;
- моделювання замовлень;
- тестування роботи клієнтської логіки без потреби у справжньому сервері.

Застосування MockAPI дозволяє швидко розгорнути backend-емуляцію і протестувати повну логіку роботи фронтенду, що особливо корисно на ранніх етапах розробки.

2.5 Система контролю версій

GitHub[17] — платформа для хостингу репозиторіїв коду, яка базується на системі контролю версій Git. У дипломному проєкті GitHub використовується для:

- зберігання коду застосунку;
- ведення історії змін;
- деплою сайту через GitHub Pages.

Це забезпечує надійність розробки, зручність командної роботи та можливість демонстрації продукту в реальному режимі.

2.6 Інструменти проектування інтерфейсу

Figma[18] — web-інструмент для UX/UI-дизайну, який дозволяє створювати інтерактивні прототипи, дизайн-макети та структури інтерфейсу.

У межах даного проєкту у Figma було:

- розроблено структуру сторінок (home, кастомізація, оформлення замовлення);
- погоджено зовнішній вигляд елементів кастомізатора;
- протестовано навігаційну логіку та зручність користування.

Використання Figma дозволило зменшити кількість ітерацій при верстці і спростити розробку HTML/CSS-інтерфейсу.

3 РЕАЛІЗАЦІЯ WEB-ЗАСТОСУНКУ

3.1 Проєктування архітектури застосунку

Архітектура web-застосунку для кастомізації та продажу тенісних ракеток побудована за принципом "клієнт-сервер" із використанням MockAPI як емулятора серверної частини. Застосунок є SPA (Single Page Application)[19] — це означає, що взаємодія з користувачем відбувається без перезавантаження сторінки, що забезпечує швидкий та зручний досвід використання.

Компоненти архітектури:

- користувач — взаємодіє із застосунком через браузер: здійснює кастомізацію ракетки, переглядає конфігурацію, додає її до кошика та оформлює замовлення;
- фронтенд (Web-застосунок) складається з кількох модулів:
 - інтерфейс (HTML/CSS) — відповідає за структуру та візуальне оформлення сторінок;
 - кастомізатор (JavaScript + Tailwind CSS) — реалізує зміну параметрів ракетки в реальному часі (колір, розмір ручки, вага, баланс тощо);
 - модуль навігації — забезпечує логіку переходів між сторінками без перезавантаження (router);
 - модуль кошика та оформлення — відповідає за збереження вибраних товарів і передачу замовлення на сервер.
- MockAPI (Backend-емулятор):
 - конфігурації ракеток — база, у якій тимчасово зберігаються параметри кастомізації;
 - замовлення — таблиця, в яку надсилаються фінальні дані замовлення користувача. MockAPI виконує роль REST API та обробляє HTTP-запити (GET, POST, DELETE тощо), імітуючи справжній сервер.

- адміністратор має доступ до MockAPI через панель керування, де може переглядати замовлення та змінювати конфігурації товарів.

- зовнішні сервіси:

- GitHub Pages — використовується для розгортання (деплой) застосунку та демонстрації його роботи онлайн;
- Figma — платформа для UX/UI-дизайну, на якій було створено макети інтерфейсу перед реалізацією.

Архітектуру web-застосунку зображено на наступній діаграмі:

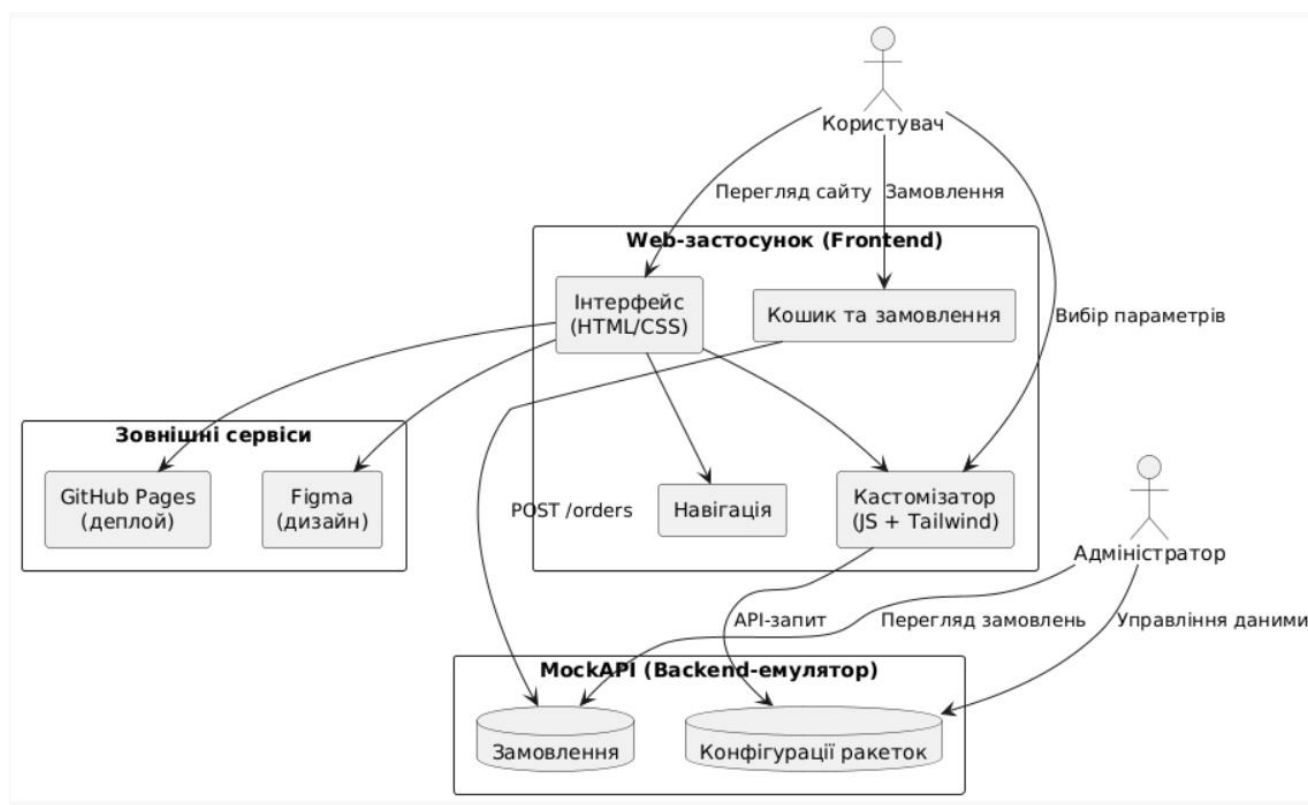


Рис. 3.1 Архітектура web-застосунку

На діаграмі архітектури зображено всі основні компоненти web-застосунку та взаємодію між ними. Користувач взаємодіє з інтерфейсом (HTML, CSS), який керується JavaScript-логікою, зокрема кастомізатором та кошиком. Дані передаються через API-запити до MockAPI, де вони зберігаються у відповідних колекціях. Адміністратор має доступ до цих колекцій з метою контролю, а розробка та деплой виконуються через GitHub Pages та Figma.

Такий підхід до архітектури дозволяє:

- швидко протестувати застосунок без створення складного бекенду;
- реалізувати повноцінну кастомізацію;
- зберігати дані користувача;
- підтримувати масштабованість та адаптивність інтерфейсу.

Таким чином, обрана архітектура є оптимальною для MVP-версії кастомізатора тенісних ракеток та може бути легко розширена у майбутньому — наприклад, шляхом підключення реального серверу та бази даних.

Також для кращого розуміння функціональної моделі web-застосунку було побудовано діаграми варіантів використання, які демонструють взаємодію користувача та адміністратора з системою. Діаграми відображають основні сценарії поведінки відповідно до ролей у системі.

Діаграма варіантів використання для користувача.

На рисунку 3.2 представлено варіанти використання, доступні для кінцевого користувача — гравця або покупця, який взаємодіє із застосунком через web-інтерфейс. Основними діями є:

- перегляд каталогу з готовими моделями ракеток;
- кастомізація тенісної ракетки (вибір параметрів: вага, баланс, розмір ручки, колір тощо);
- перегляд результату кастомізації в реальному часі;
- оформлення замовлення;
- перегляд історії раніше зроблених замовлень.



Рис. 3.2. Діаграма варіантів використання для користувача

Діаграма варіантів використання для адміністратора.

Адміністратор системи має доступ до MockAPI — інтерфейсу для управління замовленнями. На рисунку 3.3 зображено можливі дії адміністратора:

- перегляд усіх замовлень;
- додавання нових товарів;
- зміна конфігурацій товарів у системі.

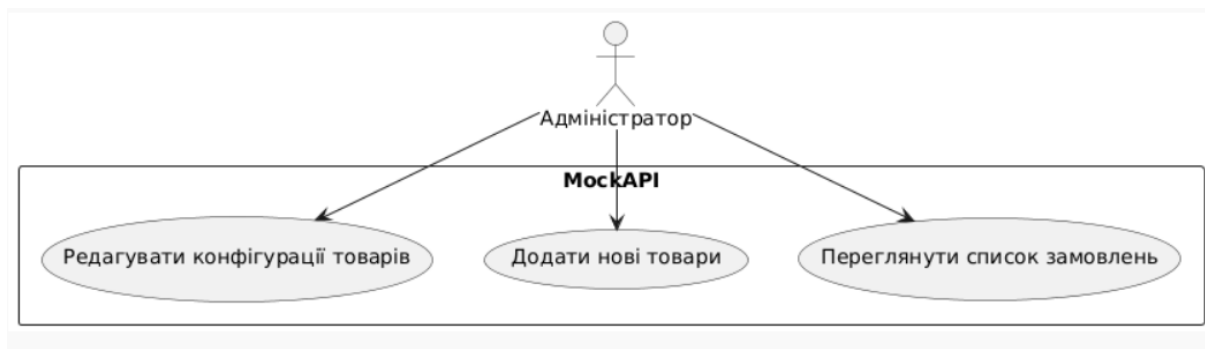


Рис. 3.3. Діаграма варіантів використання для адміністратора

Побудова діаграм варіантів використання дозволила чітко визначити основних користувачів системи та їхні взаємодії з функціональними компонентами web-застосунку. Це сприяло формуванню логічної структури інтерфейсу та полегшило етапи реалізації ключових сценаріїв роботи. Діаграми є основою для подальшого деталізованого проєктування та тестування системи.

3.2. Основні сторінки

Структура web-застосунку розроблена відповідно до принципів інтуїтивної навігації та логічної організації функціоналу. Загальна схема сторінок зображена на діаграмі сайту (Рис. 3.4), що ілюструє ієрархію переходів та взаємозв'язки між основними компонентами інтерфейсу.

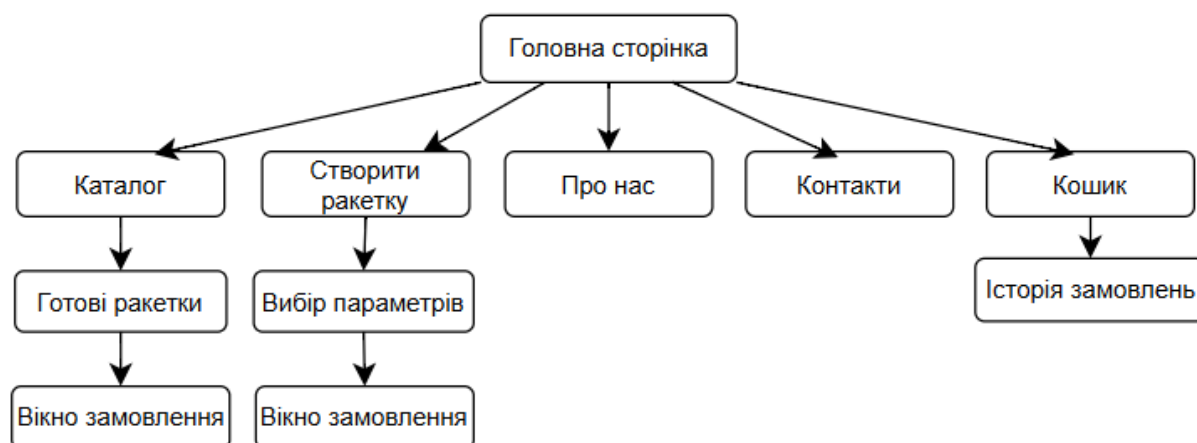


Рис. 3.4 Карта web-застосунку

Головна сторінка

Це стартова точка для користувача, з якої доступні всі основні функції застосунку. Вона містить навігаційне меню з переходами до розділів: «Каталог», «Створити ракетку», «Про нас», «Контакти» та «Кошик». На головній сторінці також розміщуються заклики до дії — наприклад, «Каталог» або «Створити ракетку», секція «Про нас», а також короткий опис кастомізації, де пояснено важливість всіх параметрів ракеток, з якими варто ознайомитися для вдалої роботи.

Каталог

Сторінка містить список вже створених, готових до замовлення тенісних ракеток. Користувач має можливість переглянути характеристики кожної моделі та перейти до вікна замовлення, де здійснюється оформлення покупки.

Створити ракетку

Цей розділ реалізує головну функціональність застосунку — кастомізацію. Він містить сторінку вибору параметрів, де користувач може змінювати вагу, баланс, розмір ручки та колір. Зміни відображаються інтерактивно у вигляді готової ракетки. Після завершення налаштування користувач переходить до вікна замовлення, де підтверджує конфігурацію і оформлює покупку.

Про нас

Інформаційна секція, яка містить опис компанії, її місію та переваги. Служить для підвищення довіри користувачів до сервісу.

Контакти

Містить контактні дані для зворотного зв'язку: електронну пошту та номер телефону. Дозволяє користувачеві зв'язатися з адміністрацією сервісу.

Кошик

Цей розділ зберігає всі замовлені товари (готові моделі або кастомізовані ракетки). Тут доступна історія замовлень, де користувач може переглянути свої попередні покупки, їх статус та деталі.

3.3 База даних

У рамках розробки MVP-версії web-застосунку для кастомізації та продажу тенісних ракеток було використано MockAPI як тимчасове рішення для зберігання даних. Це дозволило швидко реалізувати базову логіку збереження та обробки замовлень без необхідності розгортання повноцінного серверного середовища.

3.3.1 Структура ресурсу basket

Усі замовлення, які користувач оформлює (як з каталогу, так і кастомні), зберігаються у ресурсі basket. Кожен запис у цьому ресурсі має формат JSON-об'єкта зі структурою, залежною від типу товару:

Тип catalog (готова модель):

```
{  
  "id": "1",  
  "type": "catalog",  
  "title": "AeroStrike Vortex 98",  
  "price": 2000,  
  "customerName": "Аліна",  
  "customerSurname": "Мельничук",
```

```
"customerPhone": "+380999999999",
"date": "01.05.2025, 15:42:43"
}
```

Тип custom (кастомізована модель):

```
{
  "id": "2",
  "type": "custom",
  "title": "Кастомізована ракетка",
  "price": "-",
  "weight": "260",
  "balance": "Нейтральний баланс (320 мм)",
  "handleSize": "L1 (4 1/8\\")",
  "color": "rgb(255, 255, 0)"
}
```

Resource data

Edit/replace data for **basket** resource. Data must be an array and a valid JSON.

```
[
  {
    "id": "1",
    "type": "catalog",
    "title": "AeroStrike Vortex 98",
    "price": 2000,
    "customerName": "Аліна",
    "customerSurname": "Мельничук",
    "customerPhone": "+380999999999",
    "date": "01.05.2025, 15:42:43"
  },
  {
    "id": "2",
    "type": "custom",
    "title": "Кастомізована ракетка",
    "price": "-",
    "weight": "260",
    "balance": "Нейтральний баланс (320 мм)",
    "handleSize": "L1 (4 1/8\\")",
    "color": "rgb(255, 255, 0)"
  }
]
```

Close

Update

Рис. 3.5 Приклад створених замовлень в МоскAPI

Опис полів

- id – унікальний ідентифікатор запису;
- type – визначає джерело замовлення: catalog або custom;
- title – назва моделі;
- price – вартість (для custom поки що умовна або обраховується пізніше);
- customerName, customerSurname, customerPhone – контактні дані користувача;
- date – час оформлення замовлення;
- weight, balance, handleSize, color – технічні параметри, специфічні для кастомізованих моделей.

3.3.2 Структура ресурсу tennis

Окремий ресурс tennis у системі MockAPI відповідає за зберігання інформації про готові до замовлення тенісні ракетки, які відображаються на сторінці «Каталог». Кожен запис містить основні характеристики товару, включаючи зображення, ціну та посилання для перегляду деталей і оформлення замовлення.

Приклад структури одного елемента:

```
{
  "title": "AeroStrike Vortex 98",
  "image": "img/category_1.png",
  "price": 2000,
  "link": "tennis_1.html",
  "cartLink": "tennis_1.html"
}
```

Опис полів:

- title — назва моделі ракетки;
- image — шлях до зображення, яке використовується у вітрині каталогу;
- price — ціна ракетки у гривнях;
- link — сторінка з детальною інформацією про модель;

- cartLink — посилання для додавання товару до кошика (в даному випадку збігається з link).

Усього в базі зберігається п'ять моделей:

- AeroStrike Vortex 98 — бюджетна модель для новачків;
- BlazeCore ProTour 100 — професійна ракетка з високою точністю;
- TitanEdge Supreme 97 — універсальна модель для різних стилів гри;
- ShadowSpin Elite X1 — топова модель з агресивним дизайном;
- NovaFlex CarbonDrive 105 — легка ракетка для гравців, які цінують швидкість.

Призначення ресурсу tennis:

- використовується для динамічного відображення вмісту на сторінці «Каталог»;
- забезпечує зручне оновлення вмісту без зміни HTML-коду;
- дозволяє легко масштабувати перелік товарів (наприклад, при додаванні нових моделей).

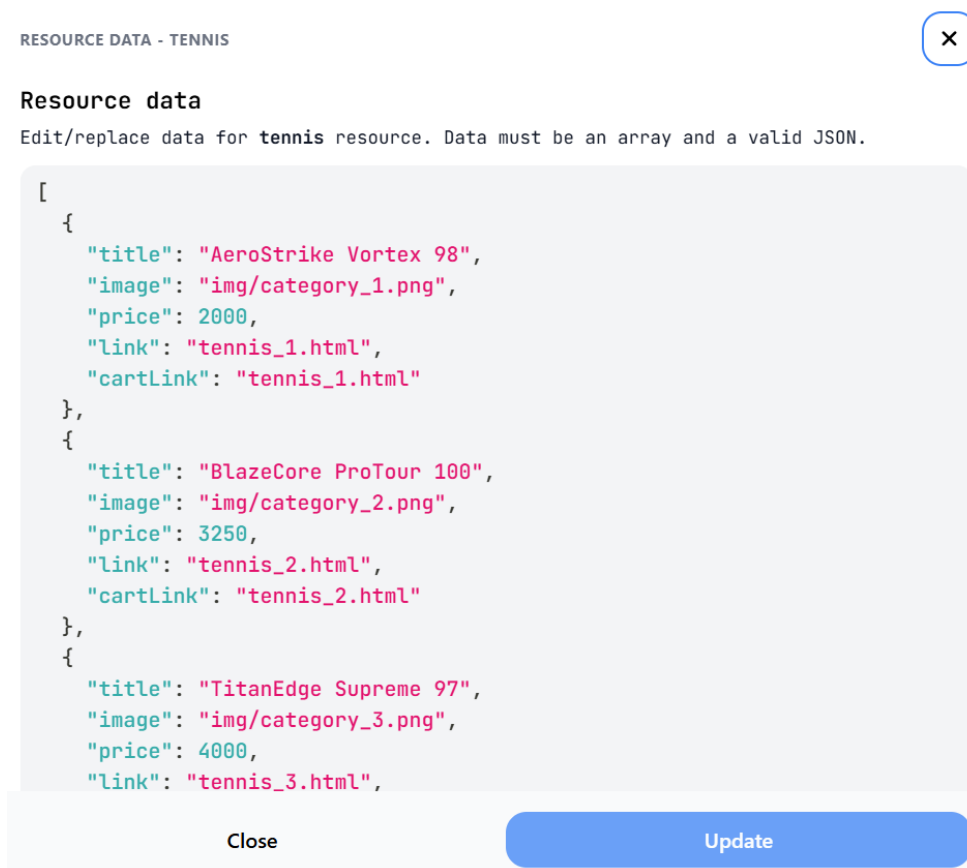


Рис. 3.6 Приклад ракеток з каталогу в MoskAPI

3.3.3 Переваги використання MockAPI

Онлайн сервіс MockAPI має багато переваг у використанні, а саме:

- швидкий запуск - не потрібно налаштовувати серверну частину;
- REST-інтерфейс - підтримуються всі базові HTTP-запити (GET, POST, PUT, DELETE);
- візуальна панель керування - зручно переглядати та редагувати дані;
- інтеграція з фронтом - дозволяє миттєво тестувати логіку взаємодії застосунку з API.

3.3.4 Перспективи розширення

MockAPI використовується як емулятор реального API, тому структура бази даних вже адаптована для подальшої інтеграції з повноцінною серверною платформою (наприклад, Node.js + MongoDB або Firebase). У майбутньому можливо:

- додати авторизацію користувачів;
- реалізувати фільтрацію та сортування замовлень;
- обчислювати ціну кастомних моделей динамічно;
- створити окрему таблицю для збереження шаблонів конфігурацій.

3.4 Структура проекту

Файлова структура web-застосунку організована відповідно до сучасних підходів до фронтенд-розробки та забезпечує логічний поділ компонентів для зручності супроводу, масштабування і тестування системи. Нижче наведено опис основних директорій і файлів проєкту:

- .vscode/ – службова папка середовища розробки Visual Studio Code, зберігає налаштування редактора;
- fonts/ – каталог зі шрифтами, що використовуються в інтерфейсі застосунку;
- img/ – зображення товарів, елементів дизайну тощо;

- node_modules/ – папка зі сторонніми залежностями, встановленими через Node.js;

- styles/ – додаткова папка для CSS-стилів, що дозволяє структурувати стилізацію за частинами інтерфейсу.

Основні HTML-файли:

- index.html – головна сторінка застосунку;
- categoryProduct.html – сторінка з категоріями тенісних ракеток;
- customTennis.html – інтерфейс для кастомізації тенісної ракетки;
- basket.html – сторінка кошика користувача;
- tennis_1.html – tennis_5.html – шаблони товарів з описами та характеристиками різних моделей ракеток.

JavaScript-файли:

- customTennis.js – обробка функціоналу кастомізації ракетки;
- item.js – робота з елементами товарів, наприклад, додавання до кошика або фільтрація;
- localData.js – зберігання даних для кошику з кешу сесії;
- tennisBasket.js – логіка взаємодії з кошиком.

CSS-файли:

- style.css – загальний файл стилів;
- mobile.css – стилі для мобільної версії;
- tablet.css – стилі для планшетів;
- tw.css – кастомні стилі, можливо, пов'язані з Tailwind CSS або іншими утилітами.

Конфігураційні файли:

- package.json – список залежностей та скриптів для роботи застосунку.
- package-lock.json – зафіксовані версії встановлених пакетів.

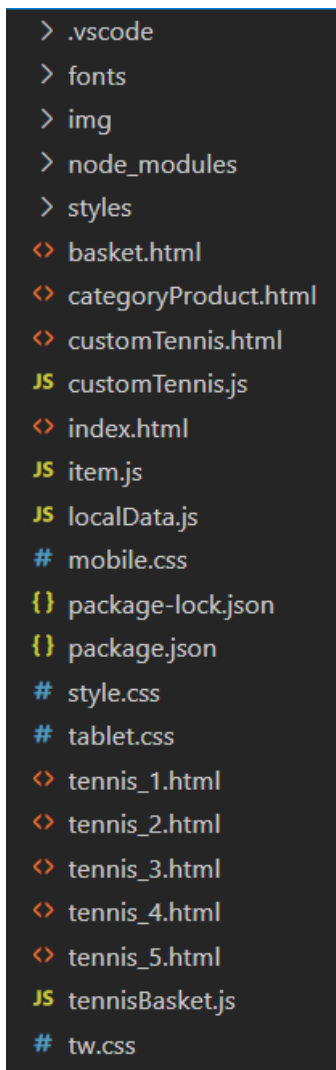


Рис. 3.7 Структура проекту

Основні компоненти:

- HTML-файли (index.html, tennis_1.html, basket.html) відповідають за структуру кожної сторінки інтерфейсу. Вони містять контейнер для кастомізатора, каталог товарів і форму оформлення замовлень;

- JavaScript-файли (item.js, tennisBasket.js, customTennis.js) реалізують динамічну взаємодію, обробку подій, кастомізацію параметрів ракетки, передачу даних до MockAPI і оновлення DOM;

- CSS-файли забезпечують стилізацію інтерфейсу. Структура стилів організована за принципом mobile-first, із окремими файлами для мобільних, планшетних та десктопних пристроїв. Також використано утилітарний підхід через Tailwind CSS;

- каталоги `img`, `fonts`, `styles` використовуються для зберігання ресурсів, зображень товарів, шрифтів, та стилів відповідно;

- `package.json` містить конфігурацію проєкту, включаючи залежності, скрипти запуску та сервісну інформацію, а `node_modules` — встановлені залежності.

3.5 Програмна реалізація web-застосунку

3.5.1 Кастомізація ракеток

У цьому підрозділі наведено фрагмент HTML-коду, який відповідає за реалізацію інтерфейсу вибору технічних параметрів тенісної ракетки. Зокрема, за допомогою `<input type="radio">` та відповідних міток `<label>` реалізовано вибір ваги (260–330 г) та балансу (на голову, нейтральний, на ручку). Така структура забезпечує інтуїтивно зрозумілу взаємодію користувача з кастомізатором і дозволяє динамічно оновлювати візуальне представлення обраної конфігурації без перезавантаження сторінки.

```
<div class="custom__panel--Grip">
  <div class="container--my">
    <input class="display__none" type="radio" id="grip_1" />
    <label class="rad active__Grip active" for="grip_1">260</label>
    <input class="display__none" type="radio" id="grip_2" />
    <label class="rad active__Grip" for="grip_2">280 </label>
    <input class="display__none" type="radio" id="grip_3" />
    <label class="rad active__Grip" for="grip_3">300 </label>
    <input class="display__none" type="radio" id="grip_4" />
    <label class="rad active__Grip" for="grip_4">330 </label>
  </div>
</div>
<div class="custom__panel--Flex">
  <div class="container--my">
```

```

<input class="display__none" type="radio" id="flex_1" />
<label class="rad active__Flex active" for="flex_1">Баланс на
голову (330+ мм)</label>
<input class="display__none" type="radio" id="flex_2" />
<label class="rad active__Flex" for="flex_2">Нейтральний баланс
(320 мм)</label>
<input class="display__none" type="radio" id="flex_3" />
<label class="rad active__Flex" for="flex_3">Баланс на ручку (310
мм і менше)</label>
</div>
</div>

```

3.5.2 Каталог ракеток

У даному підрозділі представлено код сторінки каталогу, яка відображає детальну інформацію про кожну модель ракетки. HTML-розмітка включає назву, ціну, опис моделі, варіанти сторони гри та розмір ручки. Крім того, реалізовано елементи керування кількістю одиниць товару та кнопку «Купити», яка додає обраний товар до кошика. Завдяки використанню класів Tailwind CSS забезпечено адаптивне та сучасне візуальне оформлення інтерфейсу.

```

<div class="w-3/5 item__description">
  <h2 class="title">AeroStrike Vortex 98</h2>
  <span class="price">2000 грн</span>
  <p class="mt-5">Сторона</p>
  <div class="flex gap-6 item__flex mt-4">
    <a id="flex_71" class="pt-3 pb-3 pr-8 pl-8 border rounded-xl item__btn
item__span--active" href="">Ліва </a>
    <a id="flex_65" class="pt-3 pb-3 pr-8 pl-8 border rounded-xl" href="">
Права </a>
  </div>
  <p class="mt-5">Розмір ручки</p>

```

```

<div class="flex gap-6 item__flex mt-4">
  <a id="curve_71" class="pt-3 pb-3 pr-8 pl-8 border rounded-xl
item__btn item__span--active" href=""> L3 (4 3/8") </a>
  <a id="curve_65" class="pt-3 pb-3 pr-8 pl-8 border rounded-xl"
href=""> L5 (4 5/8") </a>
</div>

```

```

<p class="mt-5 mb-5"></p>
<div class="flex gap-28 content-center item__container__btn">
  <div class="item__flex">
    <a id="quantityMinus" class="pt-3 pb-3 pr-8 pl-8 border rounded-xl"
href="">-</a>

```

```

    <div>
      <a class="item__flex__counter" id="quantityCounter" href=""></a>
    </div>
    <a id="quantityPlus" class="pt-3 pb-3 pr-8 pl-8 border rounded-xl" href="">+</a>
  </div>

```

```

  <a class="item__container__btn__buy rad" href="###42324#"> Купити
 </a>
</div>

```

```

<div class="flex flex-col gap-8 mt-6">
  <div class="flex flex-col gap-4">
    <h4 class="item--span">AeroStrike Vortex 98 — Точність та
швидкість у кожному русі</h4>

```

```

    <p>
      AeroStrike Vortex 98 — це ракетка для гравців, які цінують
контроль, чіткість ударів і максимальну маневреність. Завдяки вдосконаленій
аеродинамічній конструкції та технології Vortex Frame, ця модель забезпечує
потужну швидкість замаху та чудову стабільність навіть під час найдинамічніших
розіграшів. Головка розміром 98 кв. дюймів дарує більшу точність і відчуття м'яча,

```

що робить Vortex 98 ідеальним вибором для гравців з агресивним стилем гри, які хочуть повністю контролювати темп на корті.

</p>

</div>

<div>

<h4 class="mb-4 item--span">Ключові особливості:</h4>

<p>●Аеродинамічна рама з технологією Vortex Frame</p>

<p>● Підходить для просунутих гравців, які прагнуть до ідеальної точності</p>

<p>● Висока стабільність і швидкість замаху</p>

<p>● Ідеально підходить для гравців середнього та високого рівня</p>

</div>

</div>

</div>

3.5.3 Оформлення замовлень

Цей розділ містить реалізацію логіки кошика та форми для оформлення замовлення. Код дозволяє користувачу переглядати обрані товари, змінювати кількість або видаляти позиції, а також заповнювати контактну інформацію для завершення покупки. Після підтвердження форми замовлення надсилаються на сервер (через API-запити до MockAPI), а користувач отримує повідомлення про успішну обробку. Такий підхід забезпечує повноцінний цикл покупки у межах одного web-застосунку.

```
const order = {
  type: 'catalog',
  title: productTitle,
  price: productPrice,
```

```

    customerName: nameInput.value.trim(),
    customerSurname: surnameInput.value.trim(),
    customerPhone: phoneInput.value.trim(),
    date: new Date().toLocaleString('uk-UA')
  };

  fetch('https://67ffe0dcb72e9cfaf7262e72.mockapi.io/basket', {
    method: 'POST',
    headers: {
      'Content-Type': 'application/json'
    },
    body: JSON.stringify(order)
  })
  .then(response => {
    if (!response.ok) {
      throw new Error('Помилка при відправленні замовлення');
    }
    return response.json();
  })
  .then(data => {
    formMessage.textContent = 'Дякуємо за замовлення!';
    formMessage.className = 'success';
    document.getElementById('order-form').reset();
    setTimeout(function() {
      document.getElementById('form-container').style.display = 'none';
      formMessage.textContent = '';
    }, 2000);
  })

```

4 ДЕМОНСТРАЦІЯ РЕЗУЛЬТАТІВ ТА ТЕСТУВАННЯ

4.1 Демонстрація web-застосунку

Web-застосунок для кастомізації тенісних ракеток реалізовано як SPA (Single Page Application), що дозволяє користувачеві в режимі реального часу налаштовувати параметри інвентарю та оформлювати замовлення. Нижче представлено ключові екрани інтерфейсу.

Головна сторінка

На головній сторінці (рис. 4.1) розміщено основне меню навігації, банер із закликом до кастомізації, короткий опис сервісу, а також опис головних параметрів кастомізації. Користувач може перейти до каталогу або розпочати налаштування власної ракетки.

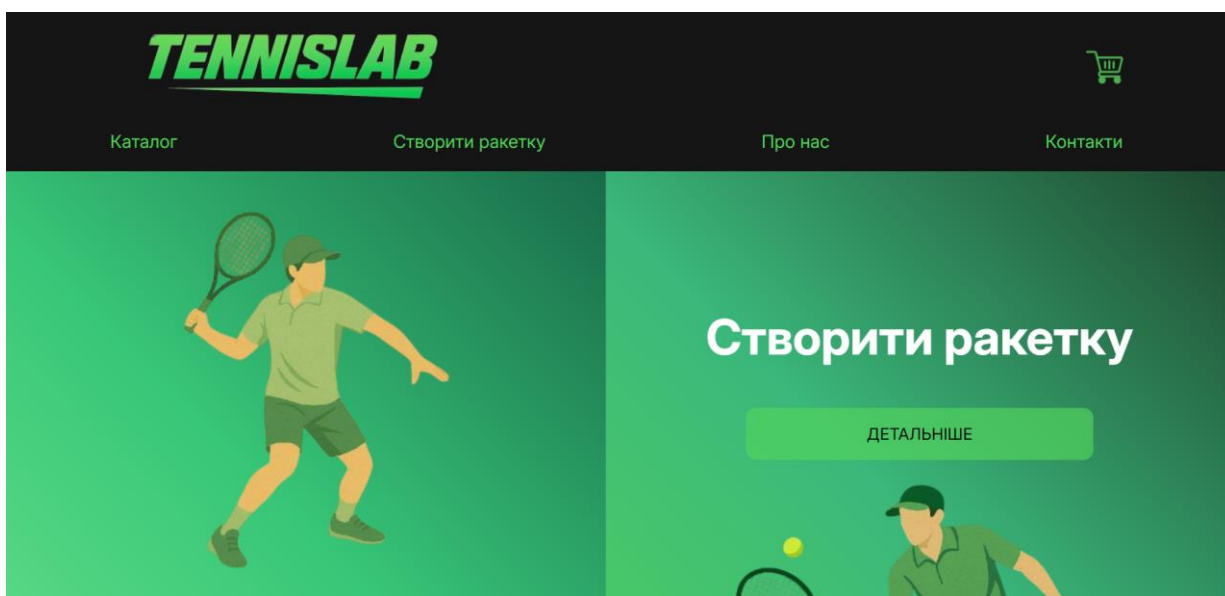


Рис. 4.1 Головне меню web-застосунку

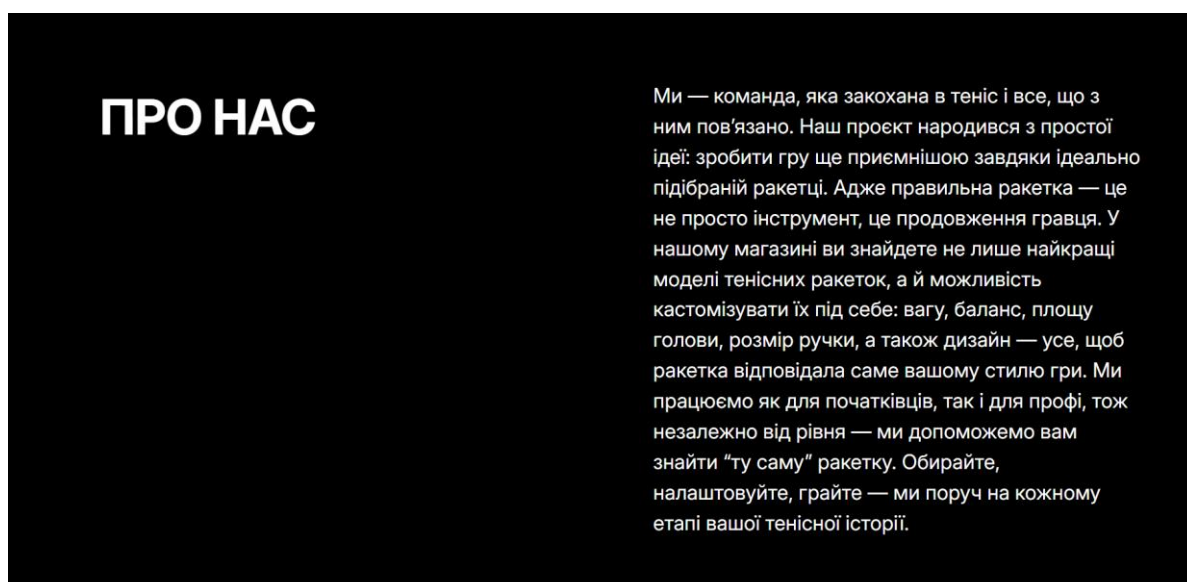


Рис. 4.2 Розділ «Про нас»

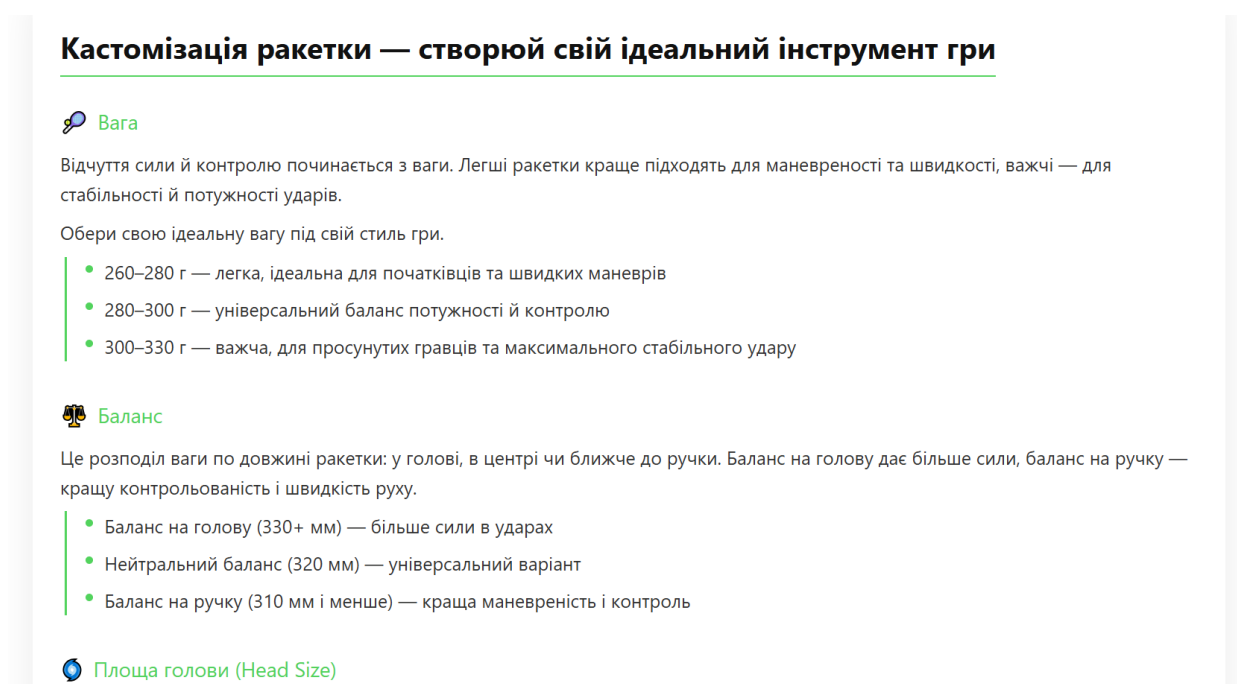


Рис. 4.3 Опис параметрів ракеток

Каталог товарів

На сторінці каталогу (рис. 4.4) відображаються готові до замовлення моделі тенісних ракеток з каталогу. Кожен товар має зображення, назву, ціну та кнопку для покупки.

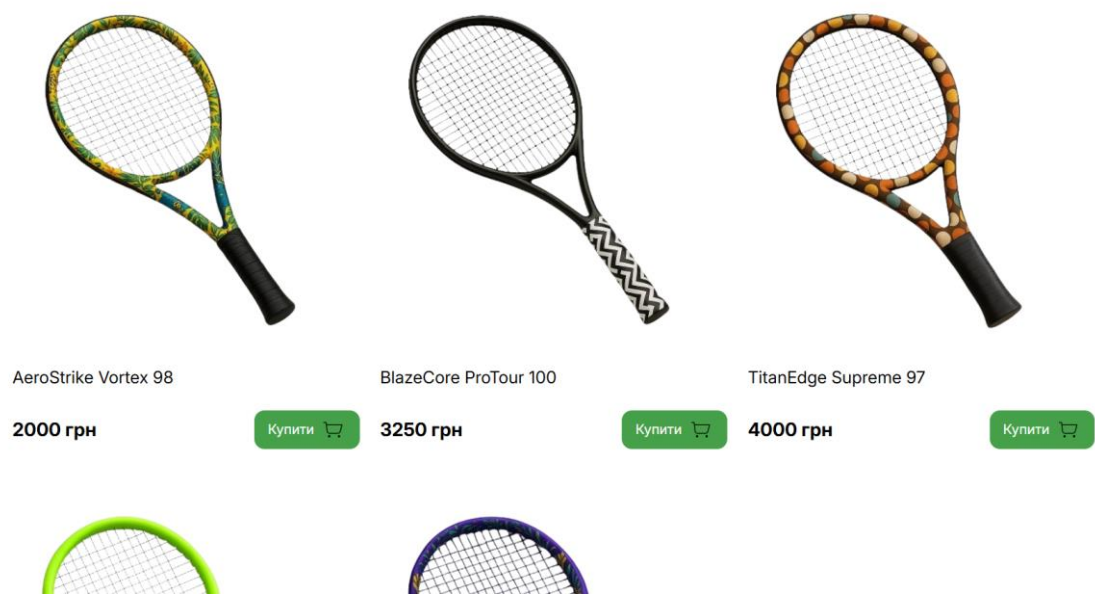


Рис. 4.4 Каталог готових ракеток

Сторінка готової ракетки

Для кожної готової ракетки є можливість обрати сторону ракетки та розмір її ручки, і після обрання параметрів придбати товар.

Каталог / AeroStrike Vortex 98



AeroStrike Vortex 98

2000 грн

Сторона

Ліва

Права

Розмір ручки

L3 (4 3/8")

L5 (4 5/8")

-

1

+

Купити

AeroStrike Vortex 98 — Точність та швидкість у кожному русі

AeroStrike Vortex 98 — це ракетка для гравців, які цінують контроль, чіткість ударів і максимальну маневреність. Завдяки вдосконаленій аеродинамічній конструкції та технології Vortex Frame, ця модель забезпечує потужну

Рис. 4.5 Приклад ракетки з каталогу

Кастомізація ракетки

Це ключовий функціонал застосунку (рис. 4.6). Користувач може вибрати вагу, баланс, розмір ручки та колір. Зміни одразу візуалізуються. Застосунок підтримує динамічне оновлення інтерфейсу без перезавантаження сторінки.

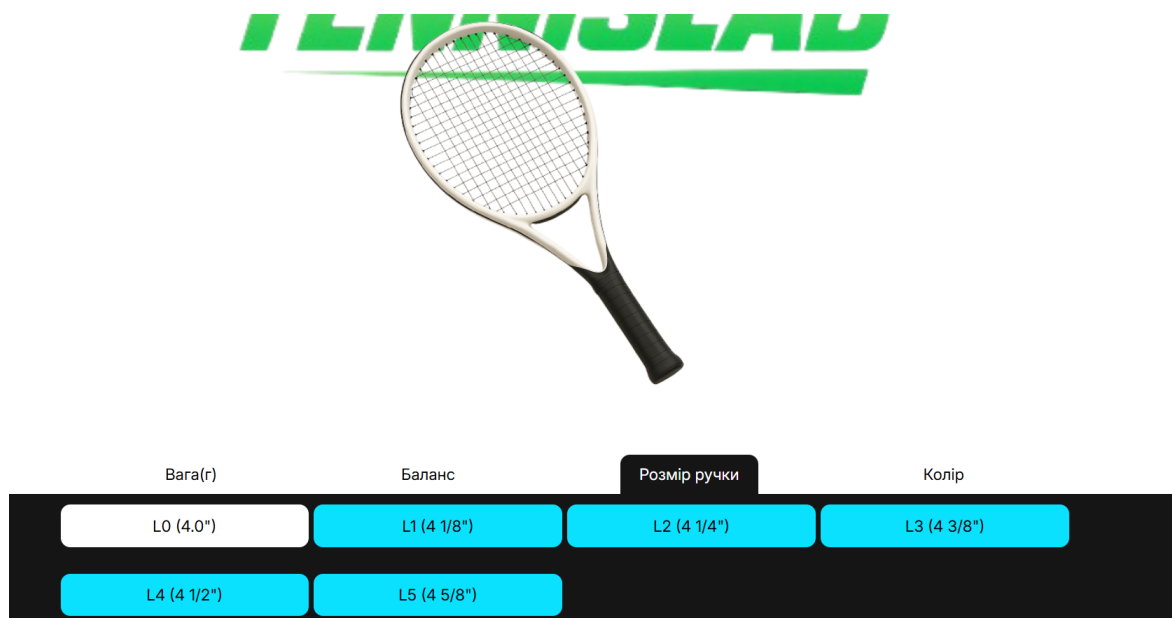


Рис. 4.6 Кастомізація параметрів ракетки

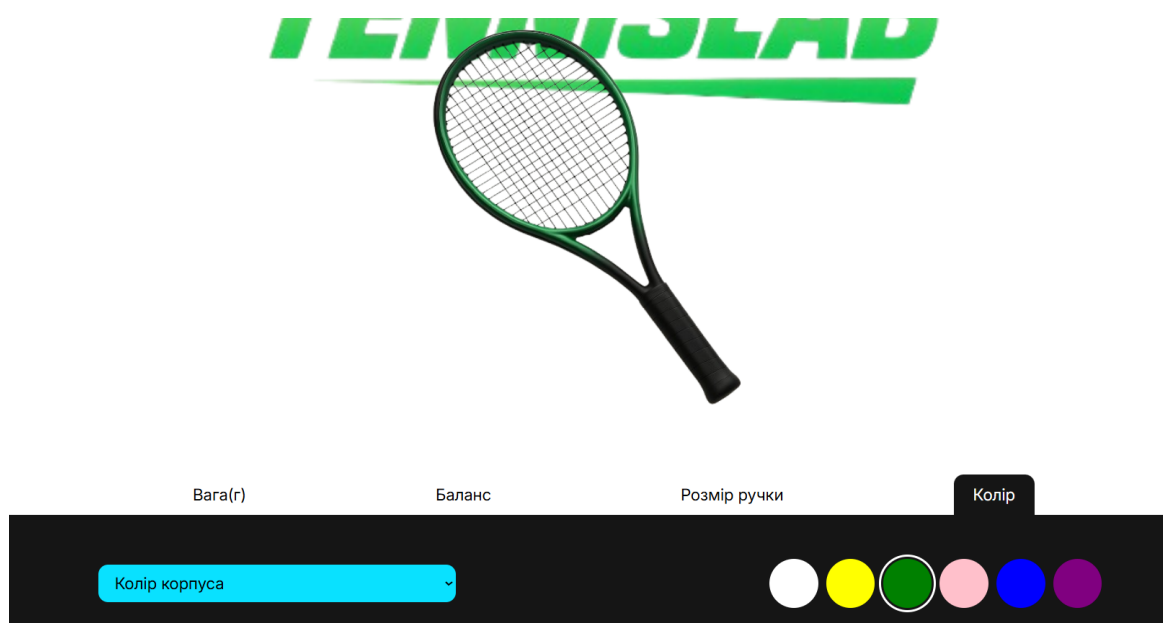


Рис. 4.7 Кастомізація параметрів ракетки

Форма оформлення замовлення

Після завершення кастомізації користувач переходить до форми для замовлення (рис. 4.8), де вказує прізвище, ім'я та номер телефону.

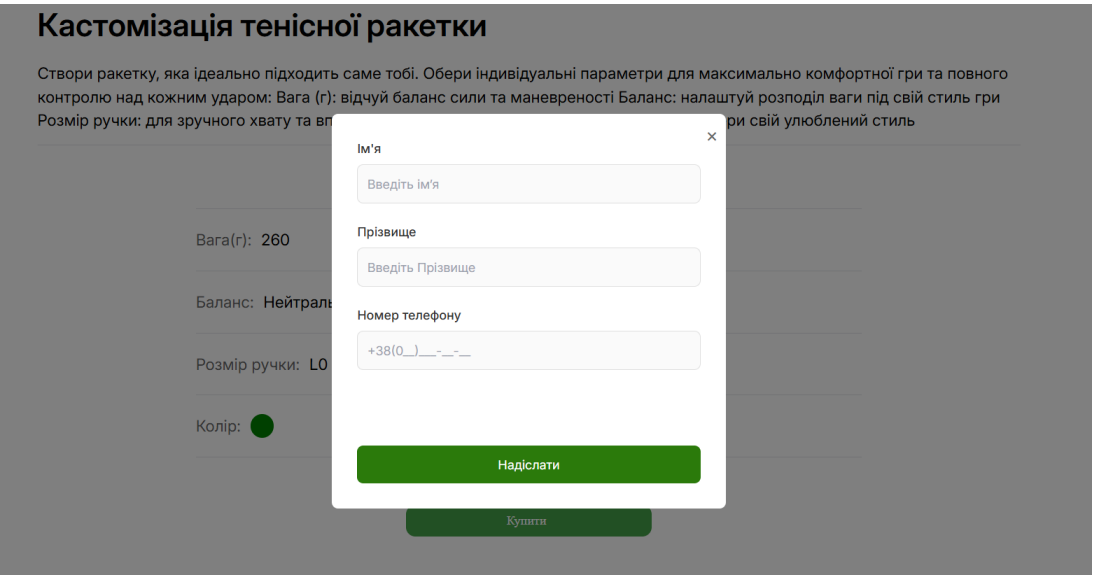


Рис. 4.8 Форма оформлення замовлення

Кошик замовлень

У кошику (рис. 4.9) зберігаються всі зроблені покупки — як готові, так і кастомні моделі. Відображаються основні характеристики, обрана конфігурація, дата замовлення та інформація про покупця. Інформація береться з бази даних, але демонструються тільки ті покупки, які зроблені з цього акаунту, дані беруться з кешу браузеру.

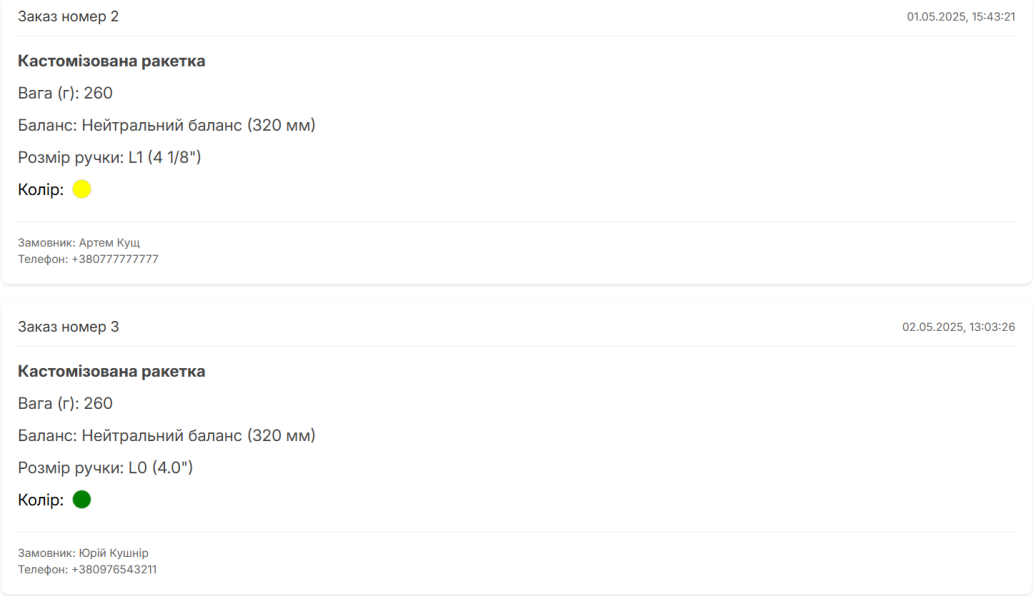


Рис. 4.9 Кошик замовлень

4.2 Тестування

4.2.1 Тест-кейси заповнення форми

У рамках тестування клієнтської частини web-застосунку було перевірено коректність роботи форми введення контактних даних, яка є частиною процесу замовлення кастомізованої тенісної ракетки. Основна мета — упевнитися, що валідація введених даних працює належним чином, запобігаючи помилкам та забезпечуючи зручність для користувача.

Тест-кейси для форми контактної інформації наведено у таблиці 4.1.

Таблиця 4.1

Тест-кейси для форми контактної інформації

№	Назва тесту	Вхідні дані	Очікуваний результат	Фактичний результат
1	Успішне заповнення форми	Ім'я: "Юрій", Прізвище: "Шевченко", Телефон: "+380991234567"	Дані приймаються, форма надсилається	Форма надіслана успішно
2	Менше 2 літер у полі "Ім'я"	Ім'я: "Ю"	Повідомлення: "Мінімум 2 українські літери"	Повідомлення виводиться, форма не надсилається
3	Порожнє поле "Прізвище"	Прізвище порожнє	Повідомлення про обов'язковість заповнення	Виводиться відповідне повідомлення
4	Некоректний номер телефону	Телефон: "0"	Повідомлення: "Введіть номер у форматі +380XXXXXXXXX"	Повідомлення виводиться, форма заблокована
5	Номер без коду країни	Телефон: "0931234567"	Повідомлення про неправильний формат	Валідація не проходить, повідомлення виведено

Продовження таблиці 4.1

Тест-кейси для форми контактної інформації

№	Назва тесту	Вхідні дані	Очікуваний результат	Фактичний результат
6	Порожня форма	Усі поля порожні	Повідомлення про обов'язковість для кожного поля	Усі поля підсвічуються червоним, виводяться повідомлення
7	Латиниця в полі "Ім'я"	Ім'я: "John"	Повідомлення про необхідність українських літер	Повідомлення виводиться, форма не надсилається
8	Телефон із зайвими цифрами	Телефон: "+380991234567890"	Повідомлення про помилкову довжину номера	Виводиться повідомлення, надсилання недоступне

Ім'я

Ю

Мінімум 2 українські літери

Прізвище

Введіть Прізвище

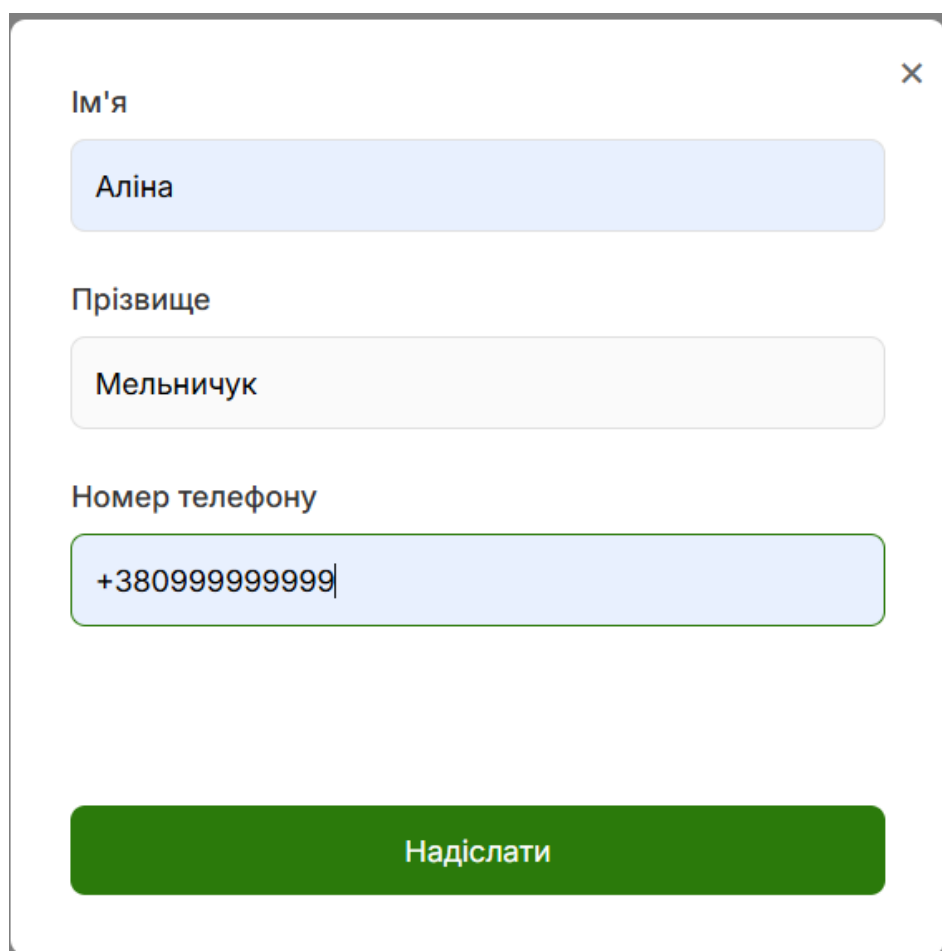
Номер телефону

0

Введіть номер у форматі +380XXXXXXXXX

Надіслати

Рис. 4.10 Приклад некоректного заповнення полів



The image shows a registration form with a close button (X) in the top right corner. It contains three input fields: 'Ім'я' (Name) with the value 'Аліна', 'Прізвище' (Surname) with the value 'Мельничук', and 'Номер телефону' (Phone Number) with the value '+380999999999'. Below the fields is a green button labeled 'Надіслати' (Send).

Рис. 4.11 Приклад правильного заповнення форми

4.2.2 Тестування адаптивності під різні пристрої

Адаптивність web-застосунку була протестована на різних пристроях і розмірах екранів, включаючи десктопні комп'ютери, планшети та мобільні телефони. Це дозволило перевірити, наскільки інтерфейс коректно масштабується та зберігає зручність використання на різних роздільностях.

На десктопній версії застосунок виглядає повноцінно - всі елементи інтерфейсу розташовані зручно, шрифт читабельний, відображаються повні меню, блоки товарів і детальна інформація про кастомізацію ракеток. Вікно займає всю ширину екрану, що дозволяє комфортно працювати з великим обсягом інформації.

Головна / Кошик

Кошик

Заказ номер 1	01.05.2025, 15:42:43
AeroStrike Vortex 98	
Ціна: 2000 грн	
Замовник: Аліна Мельничук Телефон: +380999999999	

Заказ номер 2	01.05.2025, 15:43:21
Кастомізована ракетка	
Вага (г): 260	
Баланс: Нейтральний баланс (320 мм)	
Розмір ручки: L1 (4 1/8")	
Колір: ●	

Рис. 4.12 Десктопна версія застосунку

На планшетній версії (розмір екрану приблизно 768–1024 пікселі) відображення автоматично адаптується: меню згруповане у "бургер"-кнопку, а блоки розміщуються один під одним. Така побудова дозволяє користувачу легко взаємодіяти із застосунком за допомогою сенсорного керування.

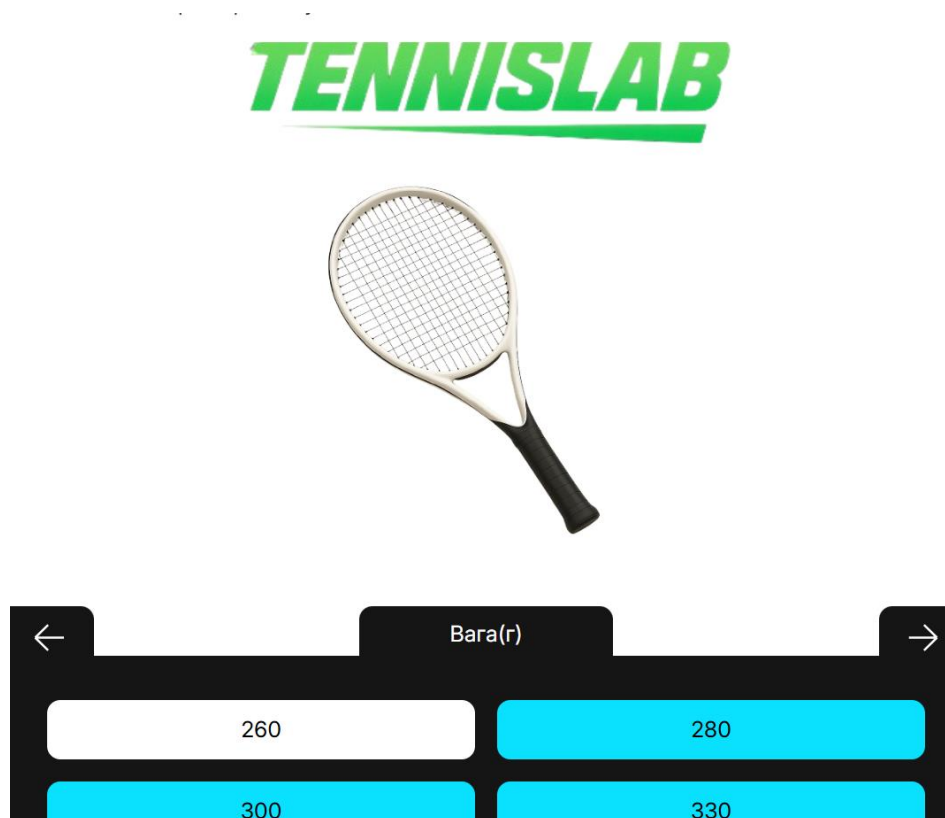


Рис. 4.13 Планшетна версія застосунку

На мобільній версії (розмір екрану 320–480 пікселів) структура сторінки ще більше спрощується. Застосунок переходить у вертикальну орієнтацію: елементи відображаються в одну колонку, шрифти та кнопки масштабуються відповідно до розміру екрана. Форма кастомізації та кнопка оформлення замовлення залишаються доступними, що дозволяє користувачу повноцінно користуватися всіма функціями навіть з телефону.

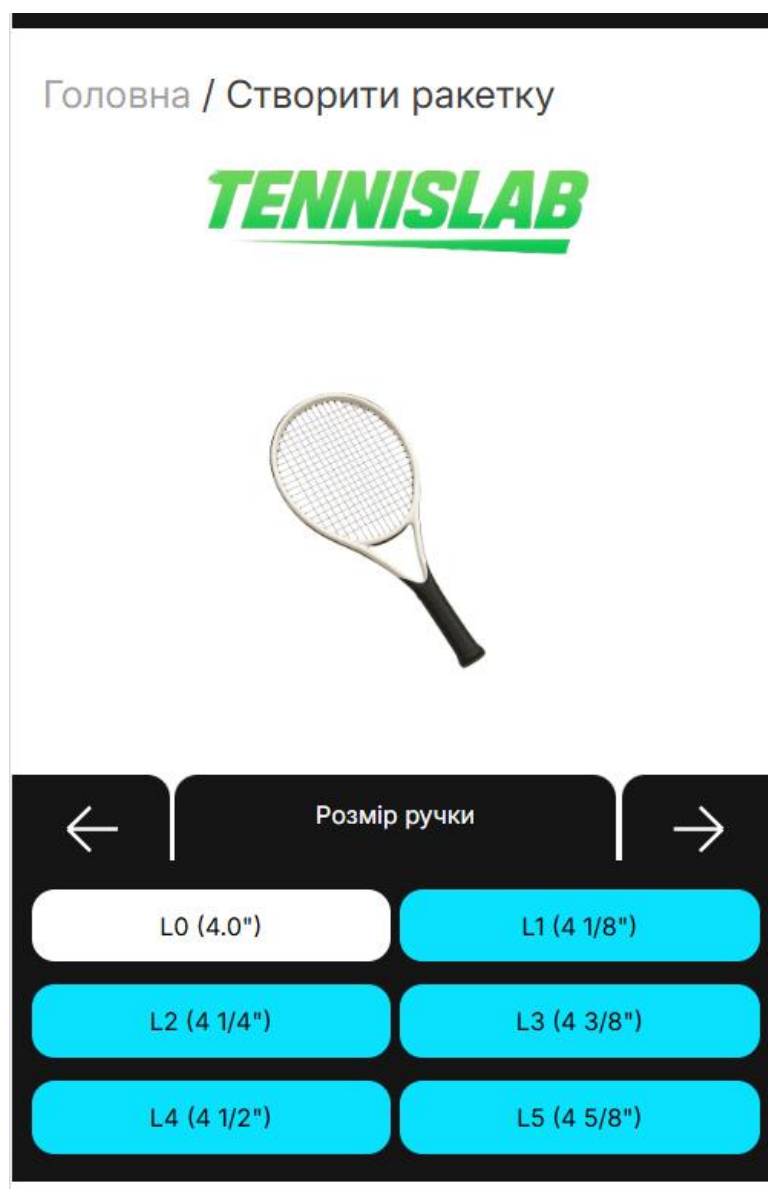


Рис. 4.14 Мобільна версія застосунку

Кошик

Заказ номер 1

01.05.2025, 15:42:43

AeroStrike Vortex 98

Ціна: 2000 грн

Замовник: Аліна Мельничук

Телефон: +380999999999

Заказ номер 2

01.05.2025, 15:43:21

Кастомізована ракетка

Вага (г): 260

Баланс: Нейтральний баланс (320 мм)

Розмір ручки: L1 (4 1/8")

Рис. 4.15 Мобільна версія застосунку

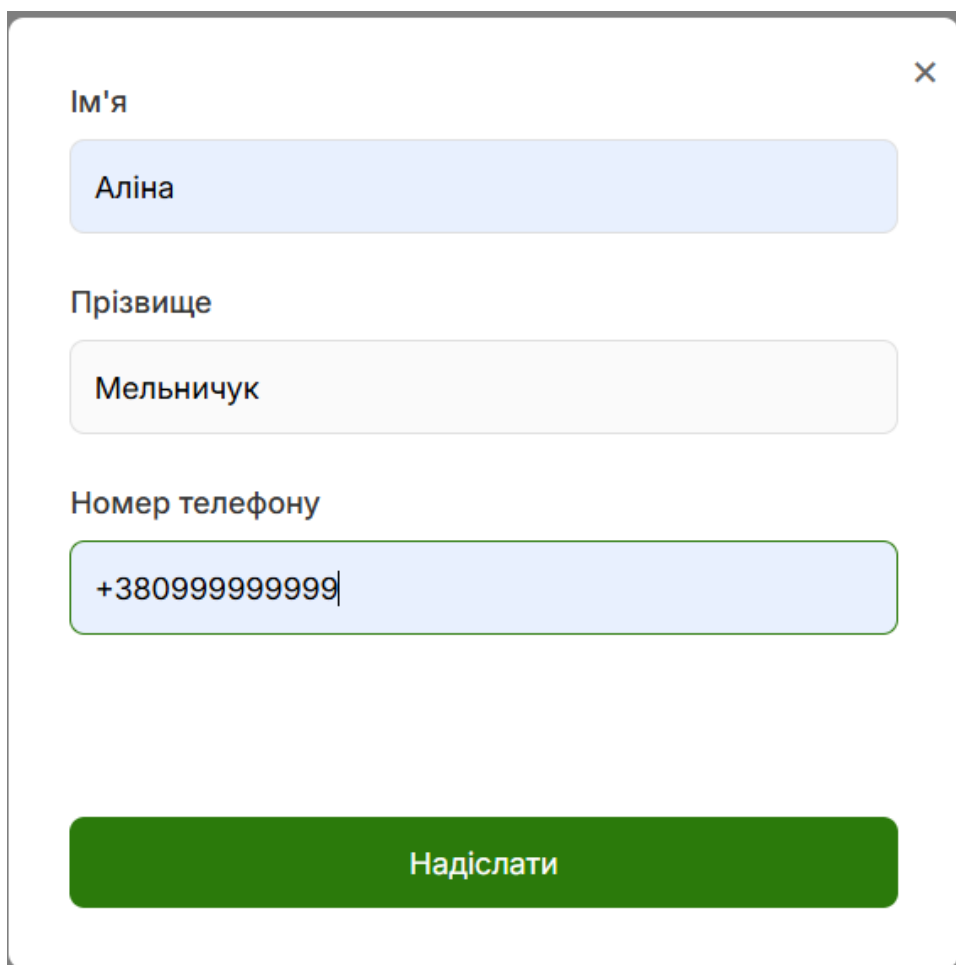
Таким чином, проведене тестування підтвердило, що web-застосунок є повністю адаптивним та зручним для користування на будь-якому типі пристрою.

4.2.3 Тестування збереження замовлень у MockAPI

Для збереження замовлень було використано сервіс MockAPI. Після заповнення форми замовлення в інтерфейсі сайту та натискання кнопки “Оформити замовлення”, дані надсилаються на MockAPI.

На скріншотах видно, що нове замовлення з’являється у списку на сайті MockAPI. У ньому відображаються поля з іменем користувача, email, моделлю ракетки та вибраними параметрами кастомізації. Також зазначено дату створення замовлення.

Це підтверджує, що передача даних працює коректно, і замовлення успішно зберігаються у зовнішньому API.



Ім'я

Аліна

Прізвище

Мельничук

Номер телефону

+380999999999

Надіслати

Рис. 4.16 Оформлене замовлення

```
{
  "id": "4",
  "type": "catalog",
  "title": "NovaFlex CarbonDrive 105",
  "price": 4900,
  "customerName": "Аліна",
  "customerSurname": "Мельничук",
  "customerPhone": "+380999999999",
  "date": "02.05.2025, 14:52:36"
}
```

Рис. 4.17 Відображення інформації про замовлення у MockAPI

ВИСНОВКИ

У межах виконання кваліфікаційної роботи було здійснено повний цикл розробки web-застосунку для кастомізації та продажу тенісних ракеток. На початковому етапі було проаналізовано предметну область електронної комерції та особливості процесу кастомізації спортивного інвентарю, що дозволило виявити ключові вимоги користувачів і сформулювати технічні характеристики майбутнього продукту. Окрему увагу було приділено вивченню актуальних аналогів, таких як Wilson Custom Tennis, RacketTier, Racquet Design та RAGE Custom, функціональні можливості яких стали основою для обґрунтованого проектування архітектури застосунку.

У результаті аналізу було окреслено вимоги до системи та створено проєкт web-застосунку з урахуванням принципів адаптивного дизайну, UX/UI-орієнтації та інтерактивності. Для реалізації було обрано відповідні інструменти: HTML, CSS, JavaScript, Tailwind CSS, Figma, MockAPI та GitHub. У процесі розробки було створено SPA-застосунок, що включає основні сторінки, зокрема головну, каталог, інтерфейс кастомізації, кошик та модуль оформлення замовлення. Реалізація клієнтської частини забезпечила динамічну взаємодію з користувачем у реальному часі, що відповідає сучасним стандартам зручності та швидкодії.

MockAPI було використано як інструмент для зберігання кастомізованих конфігурацій і оформлених замовлень, що дозволило протестувати серверну логіку без потреби у створенні повноцінного backend-рішення. Це рішення також забезпечує масштабованість системи та її готовність до подальшої інтеграції з реальними базами даних.

Розроблений застосунок пройшов комплексне тестування, яке охоплювало перевірку функціональності, коректності обробки замовлень, адаптивності інтерфейсу до різних пристроїв, а також відповідності поведінки системи очікуванням користувача.

Було підготовлено демонстраційні матеріали, що включають скріншоти інтерфейсу та структури проекту, а також фрагменти коду основних модулів, які підтверджують працездатність і завершеність реалізації. Результатом роботи став повноцінний web-застосунок, який дозволяє користувачеві не лише налаштовувати технічні та візуальні параметри тенісної ракетки, а й отримати чітке уявлення про кінцевий вигляд товару перед його придбанням.

Таким чином, проєкт реалізував усі поставлені завдання — від аналізу аналогів і формування вимог до створення інтерфейсу, використання API, тестування функціоналу та підготовки демонстраційних матеріалів. Розроблений застосунок покращує користувацький досвід, сприяє зростанню зацікавленості клієнтів у продукті та підвищує конкурентоспроможність компаній у сфері спортивної електронної комерції. Подальший розвиток системи може включати інтеграцію з платіжними сервісами, реалізацію користувацьких профілів, історії замовлень, а також використання аналітики для автоматизованого підбору оптимальних параметрів ракетки залежно від стилю гри та уподобань користувача.

Робота пройшла апробацію. За її результатами було опубліковано наступні тези доповідей:

1. Мельничук А.Ю., Замрій І.В. Визначення вимог до web-застосунку для кастомізації і продажу тенісних ракеток: Матеріали Всеукраїнської науково-технічної конференції «Застосування програмного забезпечення в ІКТ» 24 квітня 2025р., Київ Державний університет інформаційно-комунікаційних технологій. Збірник тез. К.: ДУІКТ, 2025. С.23-26.

2. Мельничук А.Ю., Замрій І.В. Проєктування архітектури та функціональних можливостей web-застосунку для кастомізації і продажу тенісних ракеток: Матеріали VI Науково-технічної конференції «Сучасний стан та перспективи розвитку IoT» 15 квітня 2025р., Київ Державний університет інформаційно-комунікаційних технологій. Збірник тез. К.: ДУІКТ, 2025. (Подано до друку).

ПЕРЕЛІК ПОСИЛАНЬ

1. Franke N., Schreier M. Product uniqueness as a driver of customer utility in mass customization // Marketing Letters. 2008. Vol. 19, № 2. P. 93–107. DOI: <https://doi.org/10.1007/s11002-007-9029-7>.
2. Bouchrika I., Harrathi B., Al-Dhaqm A., Singh M. The impact of product personalization on customer loyalty in e-commerce // Journal of Retailing and Consumer Services. 2022. Vol. 66. Article 102901. DOI: <https://doi.org/10.1016/j.jretconser.2021.102901>.
3. Sharma A., Kumari S. Effect of customized sports equipment on player performance: A review // International Journal of Sports Science & Engineering. 2021. Vol. 15, № 2. P. 98–105.
4. Wilson Custom Tennis – Create Your Own Wilson Custom Racket [Електронний ресурс]. URL: <https://www.wilson.com/en-us/customizer/tennis/custom-tennis-rackets> (дата звернення: 06.03.2025).
5. RacketTier – Custom Tennis Rackets for Your Game [Електронний ресурс]. URL: <https://rackettier.com> (дата звернення: 08.03.2025).
6. Racquet Design – Custom Designed Tennis Rackets [Електронний ресурс]. URL: <https://racquetdesign.com> (дата звернення: 08.03.2025).
7. RAGE Custom – Design Your Own Beach Tennis Racket [Електронний ресурс]. URL: <https://ragecustom.com> (дата звернення: 10.03.2025).
8. Deloitte Digital. Personalizing Growth: It's a value exchange between brands and customers [Електронний ресурс]. 2022. URL: <https://www.deloitte.com/global/en/insights/topics/marketing-and-sales-operations/personalizing-growth.html> (дата звернення: 07.05.2025).
9. Nielsen J., Budiu R. Mobile Usability. New Riders, 2012. 234 p.
10. Nah F. F.-H., Lau R. J., Kuang J. Critical factors for successful e-commerce site design: usability and speed // Proceedings of the 7th Americas Conference on Information Systems (AMCIS 2001). 2001. P. 54.

11. Khanna A. Mastering Visual Studio Code: A Beginner's Guide. Birmingham: Packt Publishing, 2020. 250 p.
12. WebStorm [Электронный ресурс]. URL: <https://www.jetbrains.com/webstorm/> (дата звернення: 10.05.2025).
13. Sublime Text [Электронный ресурс]. URL: <https://www.sublimetext.com/> (дата звернення: 03.05.2025).
14. JavaScript technologies overview [Электронный ресурс]. Mozilla Developer Network. URL: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/JavaScript_technologies_overview (дата звернення: 07.05.2025).
15. Wilson K. The Absolute Beginner's Guide to HTML and CSS: A Step-by-Step Guide with Examples and Lab Exercises. Berkeley, CA: Apress, 2023. 240 p.
16. Gherchev I. Tailwind CSS: From Zero to Production. Birmingham: Packt Publishing, 2023. 356 p.
17. About GitHub [Электронный ресурс]. GitHub Docs. URL: <https://docs.github.com/en/get-started/quickstart/about-github> (дата звернення: 05.05.2025).
18. How Figma works – Official Guide [Электронный ресурс]. Figma. URL: <https://help.figma.com/hc/en-us/articles/360040514193-How-Figma-works> (дата звернення: 11.05.2025).
19. Single-page application (SPA) [Электронный ресурс]. Mozilla Developer Network. URL: <https://developer.mozilla.org/en-US/docs/Glossary/SPA> (дата звернення: 03.05.2025).
20. PlantText – UML diagram generator [Электронный ресурс]. URL: <https://www.planttext.com/> (дата звернення: 07.05.2025).
21. Diagrams.net – Online diagram software [Электронный ресурс]. URL: <https://app.diagrams.net/> (дата звернення: 07.05.2025).
22. Вольф Ю. HTML and CSS: The Comprehensive Guide. Київ: Rheinwerk Publishing, 2023.

23. Pedro J., Faria P., Ribeiro P., Bernardino J. Analysis, Specification and Design of an e-Commerce Platform That Supports Live Product Customization [Электронный ресурс] // ResearchGate. 2016. URL: https://www.researchgate.net/publication/308960197_Analysis_Specification_and_Design_of_an_e-Commerce_Platform_That_Supports_Live_Product_Customization (дата обращения: 07.05.2025).

24. Bashiir A. A. Engineering Sports Equipment: Enhancing Athlete Performance // International Journal of Engineering and Technology. 2025. Vol. 14, № 3. P. 45–58.

25. Sukumar B. Impact of Biofeedback-Enhanced Sports Equipment on Athlete Performance: A Professional Analysis // International Journal of Physical Education, Sports and Health. 2024. Vol. 11, № 5. P. 215–222.

ДОДАТОК А. ДЕМОНСТРАЦІЙНІ МАТЕРІАЛ



ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ
ТЕХНОЛОГІЙ

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ

КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ



Розробка web-застосунку для кастомізації і продажу тенісних ракеток

Виконала студентка 4 курсу

групи ПД-41

Мельничук Аліна Юріївна

Керівник роботи

завідувач кафедри Інженерії програмного забезпечення, доктор технічних наук, професор Замрій

Ірина Вікторівна

Київ – 2025

МЕТА, ОБ'ЄКТ ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ

Мета роботи: спрощення процедури персоналізації та замовлення тенісних ракеток.

Об'єкт дослідження: процес онлайн-кастомізації та продажу спортивного інвентарю, зокрема тенісних ракеток.

Предмет дослідження: засоби та технології розробки web-застосунків для кастомізації та продажу товарів, зокрема інструменти інтерактивної візуалізації, веб-програмування та організації клієнт-серверної взаємодії в контексті електронної комерції.

ЗАДАЧІ ДИПЛОМНОЇ РОБОТИ

1. Аналіз аналогів та підходів до онлайн-кастомізації товарів у сфері електронної комерції.
2. Визначення вимог та проектування web-застосунку для кастомізації та продажу тенісних ракеток.
3. Реалізація клієнтської частини застосунку.
4. Використання MockAPI для збереження кастомізованих конфігурацій і оформлення замовлень.
5. Тестування функціональності та адаптивності застосунку.
6. Розробка демонстраційних матеріалів та оформлення проекту.

3

АНАЛІЗ АНАЛОГІВ

Компанія	Онлайн-кастомізація	Технічна кастомізація	Візуальна кастомізація	Каталог готових товарів	Унікальні особливості
RacketTier	-	+	-	-	Використання даних для персоналізованого підбору ракетки
Racquet Design	-	+	+	-	Індивідуальний підхід до дизайну та налаштування параметрів
Wilson Custom Tennis	+	-	+	-	Онлайн-інструмент для візуальної кастомізації
RAGE Custom	+	+	+	+	Повна кастомізація ракетки та використання екологічних матеріалів
TennisLab	+	+	+	+	Кастомізація ракетки з онлайн відображенням, наявність каталогу готових ракеток

4

ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

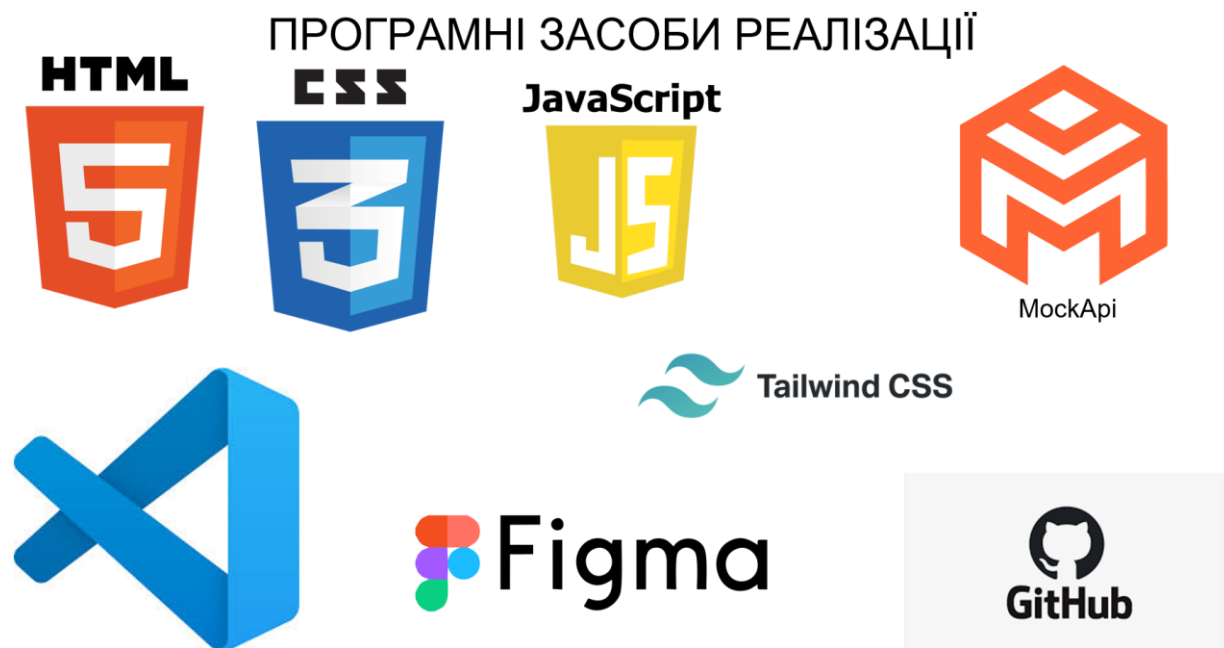
Функціональні вимоги:

- Вибір параметрів ракетки (вага, баланс, площа голови, розмір ручки, дизайн).
- Візуалізація конфігурації в реальному часі.
- Наявність каталогу готових виробів.
- Додавання товару до кошика та оформлення замовлення.

Нефункціональні вимоги:

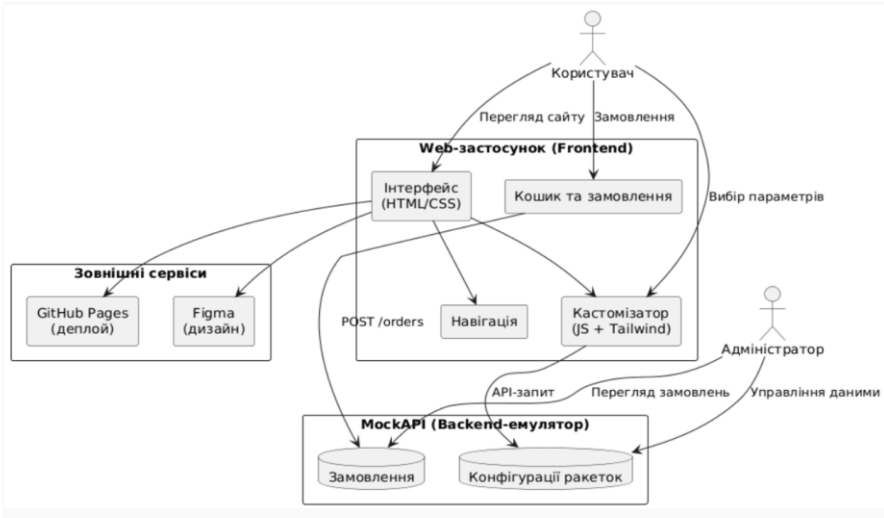
- Сумісність з сучасними браузерями (Google Chrome, Mozilla Firefox, Microsoft Edge).
- Висока швидкість завантаження сторінок(до 2 секунд).
- Адаптивність під різні пристрої.

5

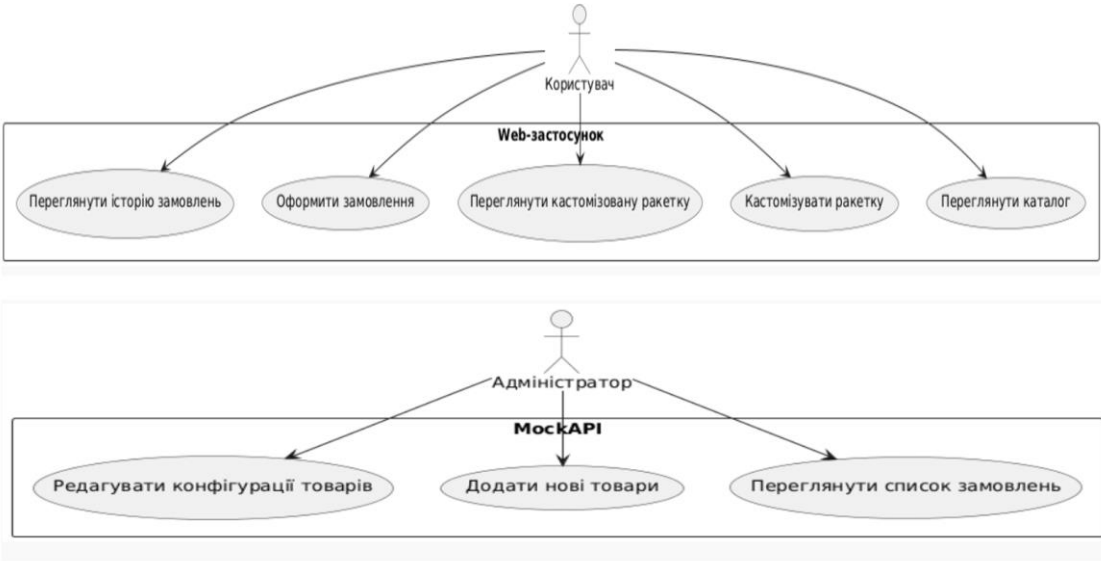


6

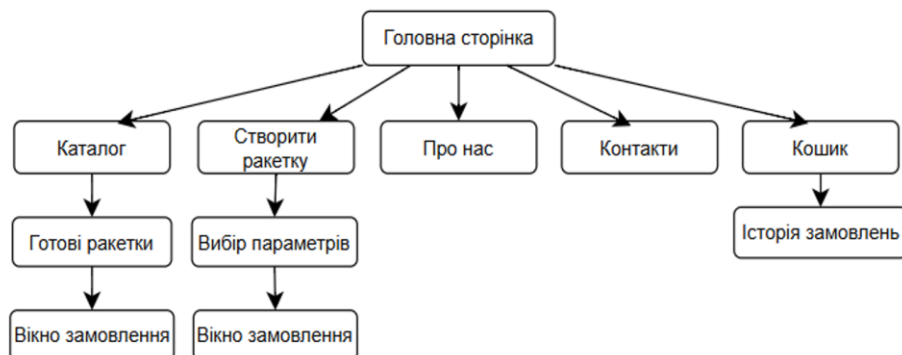
Архітектура web-застосунку



Діаграми варіантів використання

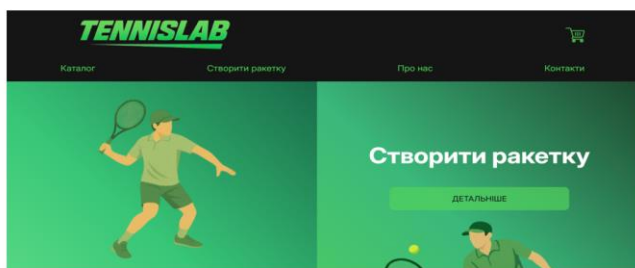


Карта web-застосунку

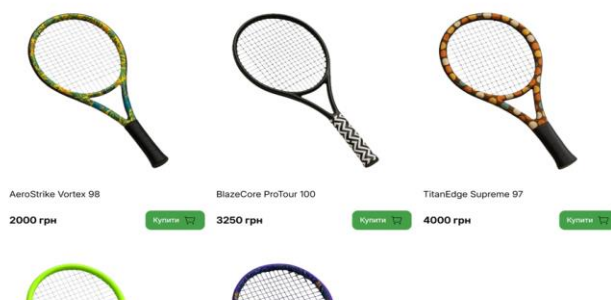


9

ЕКРАННІ ФОРМИ



Головне меню web-застосунку



Каталог готових ракеток

10

ЕКРАННІ ФОРМИ



Форма оформлення замовлення

11

АПРОБАЦІЯ РЕЗУЛЬТАТІВ ДОСЛІДЖЕННЯ

1. Мельничук А.Ю., Замрій І.В. Визначення вимог до web-застосунку для кастомізації і продажу тенісних ракеток: Матеріали Всеукраїнської науково-технічної конференції «Застосування програмного забезпечення в ІКТ» 24 квітня 2025р., Київ Державний університет інформаційно-комунікаційних технологій. Збірник тез. К.: ДУІКТ, 2025. – С.23-26.
2. Мельничук А.Ю., Замрій І.В. Проєктування архітектури та функціональних можливостей web-застосунку для кастомізації і продажу тенісних ракеток: Матеріали VI Науково-технічної конференції «Сучасний стан та перспективи розвитку IoT» 15 квітня 2025р., Київ Державний університет інформаційно-комунікаційних технологій. Збірник тез. К.: ДУІКТ, 2025. (Подано до друку).

ВИСНОВКИ

1. Досліджено сучасні підходи до онлайн-кастомізації товарів у сфері електронної комерції, що дозволило сформулювати вимоги до функціоналу та інтерфейсу web-застосунку.
2. Проаналізовано наявні рішення та типові архітектурні підходи, що дало змогу визначити оптимальну структуру застосунку та методи реалізації динамічного інтерфейсу.
3. Сформовано функціональні та нефункціональні вимоги, які стали основою для подальшого проєктування.
4. Реалізовано клієнтську частину застосунку з використанням HTML, CSS, JavaScript та Tailwind CSS, що забезпечило адаптивний, зручний та швидкий інтерфейс.
5. Інтегровано MockAPI для імітації серверної взаємодії, що дозволило ефективно протестувати збереження та обробку конфігурацій.
6. Проведено тестування застосунку, що підтвердило його працездатність, адаптивність і відповідність вимогам.

ДОДАТОК Б. ЛІСТИНГИ ОСНОВНИХ МОДУЛІВ

basket.html

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width,
initial-scale=1.0" />
    <title>TennisLab - Кошик</title>
    <link rel="stylesheet" href="style.css" />
    <link rel="stylesheet" href="tablet.css" />
    <link rel="stylesheet" href="mobile.css" />
    <link rel="stylesheet" href="tw.css" />
  </head>
  <body>
    <header>
      <div class="container--my">
        <div class="header__logo">
          <a href="index.html" class="header__logo--
tennis">
            
          </a>
          <a href="basket.html">  </a>
        </div>
        <div>
          <nav class="menu">
            <ul>
              <li><a
href="categoryProduct.html">Каталог</a></li>
              <li class="nav--2"><a
href="customTennis.html">Створити ракетку</a></li>
              <li class="nav--5"><a
href="index.html#aboutUs">Про нас</a></li>
              <li><a href="#contact">Контакти</a></li>
            </ul>
          </nav>
        </div>
      </div>
    </header>
    <div class="pageItem">
      <div class="container--my">
        <div class="container__pageItem"><a
href="index.html">Головна</a> /
        <span>Кошик</span></div>
      </div>
    </div>
    <section class="basket-section">
      <div class="container--my">
        <h1>Кошик</h1>
        <div id="orders-container">
          <!-- Тут будуть відображатися замовлення -->
        </div>
      </div>
    </section>
    <footer id="contact">
      <div class="footer--padding">
        <div class="container--my">
          <div class="footer__navigation flex--gap">
            <li><a
href="categoryProduct.html">Каталог</a></li>
            <li class="nav--2"><a
href="customTennis.html">Створити ракетку</a></li>

```

```

            <li class="nav--5"><a
href="index.html#aboutUs">Про нас</a></li>
            <li><a href="#contact">Контакти</a></li>
          </div>
        </div>
      <div class="footer__info">
        <div>
          <div class="flex--gap">
            <a href="">+380 (63) 648 36 11</a>
            <a href="">tennis@gmail.com</a>
          </div>
        </div>
        <a href="index.html">
          
        </a>
      </div>
    </div>
  </body>
</html>
<script>
  // Функція для відображення замовлень з
localStorage
  function displayOrders() {
    const ordersContainer =
document.getElementById('orders-container');

    // Отримуємо дані з localStorage
    const orders =
JSON.parse(localStorage.getItem('customOrders')) || [];

    if (orders.length === 0) {
      ordersContainer.innerHTML = '<p class="empty-
basket">Ваш кошик порожній</p>';
      return;
    }

    let ordersHTML = '';
    orders.forEach((order, index) => {
      if (order.type === 'custom') {
        // Відображення кастомізованої ракетки
        ordersHTML += `
        <div class="order-item">
          <div class="order-header">
            <h3>Заказ номер ${index + 1}</h3>
            <span class="order-
date">${order.date}</span>
          </div>
          <div class="order-details">
            <p><strong>${order.title}</strong></p>
            <p>Bara (r): ${order.weight}</p>
            <p>Баланс: ${order.balance}</p>
            <p>Розмір ручки: ${order.handleSize}</p>
            <div class="color-display">
              <span>Колір:</span>
              <span class="color-sample"
style="background-color: ${order.color}"></span>
            </div>
            <div class="customer-info">
              <p>Замовник: ${order.customerName}
${order.customerSurname}</p>
              <p>Телефон:
${order.customerPhone}</p>
            </div>
          </div>
        `;
      }
    });
    ordersContainer.innerHTML += ordersHTML;
  }

```

```

    </div>
  `;
  } else {
    // Відображення готової ракетки з каталогу
    ordersHTML += `
    <div class="order-item">
      <div class="order-header">
        <h3>Заказ номер ${index + 1}</h3>
        <span class="order-
date">${order.date}</span>
      </div>
      <div class="order-details">
        <p><strong>${order.title}</strong></p>
        <p>Ціна: ${order.price} грн</p>
        <div class="customer-info">
          <p>Замовник: ${order.customerName}
${order.customerSurname}</p>
          <p>Телефон:
${order.customerPhone}</p>
        </div>
      </div>
    </div>
  `;
  }
});

ordersContainer.innerHTML = ordersHTML;
}

// Викликаємо функцію при завантаженні сторінки
document.addEventListener('DOMContentLoaded',
displayOrders);
</script>
</body>
</html>

```

index.html

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width,
initial-scale=1.0" />
    <title>TennisLab</title>
    <link rel="stylesheet" href="style.css" />
    <link rel="stylesheet" href="tablet.css" />
    <link rel="stylesheet" href="mobile.css" />
    <link rel="stylesheet" href="tw.css" />
  </head>
  <body>
    <header>
      <div class="container--my">
        <div class="header__logo">
          <a href="index.html" class="header__logo--
tennis">
            
          </a>
          <a href="basket.html">  </a>
        </div>
        <div>
          <nav class="menu">
            <ul>
              <li><a
href="categoryProduct.html">Каталог</a></li>
              <li class="nav--2"><a
href="customTennis.html">Створити ракетку</a></li>

```

```

            <li class="nav--5"><a href="#aboutUs">Про
нас</a></li>
            <li><a href="#contact">Контакти</a></li>
          </ul>
        </nav>
      </div>
    </div>
  </header>

  <section class="section__shop">
    <div class="left__ringette">
      <a href="categoryProduct.html">
        
      </a>
      <a href="categoryProduct.html">
        <h2 class="section__shop--text">Каталог</h2>
      </a>
      <a href="categoryProduct.html" class="btn__shop
btn__left--color rad">ДЕТАЛЬНІШЕ</a>
    </div>
    <div class="right__hockey">
      <div>
        <a href="customTennis.html">
          <h2 class="section__shop--text">Створити
ракетку</h2>
        </a>
        <a href="customTennis.html" class="btn__shop
btn__right--color rad">ДЕТАЛЬНІШЕ</a>
      </div>
      <a href="customTennis.html" style="margin-top:
1rem">
        
      </a>
    </div>
  </section>

  <section id="aboutUs" class="section_about">
    <div class="container--my">
      <h3>ПРО НАС</h3>
      <p>
        Ми — команда, яка закохана в теніс і все, що з
ним пов'язано. Наш проєкт народився з простої ідеї:
зробити гру ще приємнішою завдяки ідеально підібраній
ракетці. Адже правильна ракетка — це не просто
інструмент, це продовження гравця. У нашому магазині ви
знайдете не лише
        найкращі моделі тенісних ракеток, а й
        можливість кастомізувати їх під себе: вагу, баланс, площу
        голови, розмір ручки, а також дизайн — усе, щоб ракетка
        відповідала саме вашому стилю гри. Ми працюємо як для
        початківців, так і для профі, тож незалежно від рівня — ми
        допоможемо вам
        знайти “ту саму” ракетку. Обирайте,
        налаштовуйте, грайте — ми поруч на кожному етапі вашої
        тенісної історії.
      </p>
    </div>
  </section>
  <section class="section__img">
    <div class="container--my">
      <div class="customizer">
        <h2>Кастомізація ракетки — створюй свій
ідеальний інструмент гри</h2>
        <div class="custom-section">
          <h3>Bara</h3>
          <p>Відчуття сили й контролю починається з
ваги. Легші ракетки краще підходять для маневреності та

```

швидкості, важчі — для стабільності й потужності ударів.</p>

<p>Обери свою ідеальну вагу під свій стиль гри.</p>

```
<div class="custom-options">
  <p>260–280 г — легка, ідеальна для
початківців та швидких маневрів</p>
  <p>280–300 г — універсальний баланс
потужності й контролю</p>
  <p>300–330 г — важча, для просунутих
гравців та максимального стабільного удару</p>
</div>
</div>
```

```
<div class="custom-section">
  <h3>Баланс</h3>
  <p>Це розподіл ваги по довжині ракетки: у
голові, в центрі чи ближче до ручки. Баланс на голову дає
більше сили, баланс на ручку — кращу контрольованість і
швидкість руху.</p>
  <div class="custom-options">
    <p>Баланс на голову (330+ мм) — більше
сили в ударах</p>
    <p>Нейтральний баланс (320 мм) —
універсальний варіант</p>
    <p>Баланс на ручку (310 мм і менше) —
краща маневреність і контроль</p>
  </div>
</div>
```

```
<div class="custom-section">
  <h3>Площа голови (Head Size)</h3>
  <p>Більша площа голови (oversize) — більше
прощення помилок і потужні удари. Менша — більше
контролю для точних гравців.</p>
  <p>Обирай, що важливіше саме тобі.</p>
  <div class="custom-options">
    <p>85–95 in2 — для профі, максимальний
контроль</p>
    <p>96–105 in2 — середній розмір,
ідеальний баланс</p>
    <p>106–115+ in2 — великі голови, більше
сили та прощення</p>
  </div>
</div>
```

```
<div class="custom-section">
  <h3>Розмір ручки (Grip Size)</h3>
  <p>Комфорт гри починається з правильного
хвату. Занадто тонка — напруга в руці, занадто товста —
втрата чутливості.</p>
  <p>Ми допоможемо підібрати розмір, який
ідеально “ляже” в руку.</p>
  <div class="custom-options">
    <p>L0 (4.0") — дуже тонка</p>
    <p>L1 (4 1/8")</p>
    <p>L2 (4 1/4")</p>
    <p>L3 (4 3/8") — стандартний для
багатьох гравців</p>
    <p>L4 (4 1/2")</p>
    <p>L5 (4 5/8") — товста ручка, для
великих рук</p>
  </div>
</div>
</div>
</section>
<footer id="contact">
  <div class="footer--padding">
```

```
<div class="container--my">
  <div class="footer__navigation flex--gap">
    <li><a
href="categoryProduct.html">Каталог</a></li>
    <li class="nav--2"><a
href="customTennis.html">Створити ракетку</a></li>
    <li class="nav--5"><a
href="index.html#aboutUs">Про нас</a></li>
    <li><a href="#contact">Контакти</a></li>
  </div>
  <div class="footer__info">
    <div>
      <div class="flex--gap">
        <a href="">+380 (63) 648 36 11</a>
        <a href="">tennis@gmail.com</a>
      </div>
    </div>
    <a href="index.html">
      
    </a>
  </div>
</div>
</body>
</html>
```

tennis_1.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width,
initial-scale=1.0" />
    <title>TennisLab</title>
    <link rel="stylesheet" href="style.css" />
    <link rel="stylesheet" href="tw.css" />
    <link rel="stylesheet" href="tablet.css" />
    <link rel="stylesheet" href="mobile.css" />
    <script src="item.js" defer</script>
    <script src="tennisBasket.js" defer</script>
  </head>
  <body>
    <header>
      <div class="container--my">
        <div class="header__logo">
          <a href="index.html" class="header__logo--
tennis">
            
          </a>
          <a href="basket.html">  </a>
        </div>
        <div>
          <nav class="menu">
            <ul>
              <li><a
href="categoryProduct.html">Каталог</a></li>
              <li class="nav--2"><a
href="customTennis.html">Створити ракетку</a></li>
              <li class="nav--5"><a
href="index.html#aboutUs">Про нас</a></li>
              <li><a href="#contact">Контакти</a></li>
            </ul>
          </nav>
        </div>
      </div>
    </header>
```

```

<div class="pageItem">
  <div class="container--my">
    <div class="container__pageItem"><a
href="categoryProduct.html">Каталог </a>
<span>AeroStrike Vortex 98</span></div>
    </div>
    <section class="section__item">
      <div class="container--my">
        <div class="flex pb-64 pt-14">
          <div class="w-2/5 flex-column gap-y-12">
            
            <div class="grid grid-cols-2 gap-y-12 mt-12
item__img"></div>
            </div>
            <div class="tennis__img" style="display: flex;
justify-content: center">
              
            </div>

          <div class="w-3/5 item__description">
            <h2 class="title">AeroStrike Vortex 98</h2>
            <span class="price">2000 грн</span>
            <p class="mt-5">Сторона</p>
            <div class="flex gap-6 item__flex mt-4">
              <a id="flex_71" class="pt-3 pb-3 pr-8 pl-8
border rounded-xl item__btn item__span--active" href="">
Ліва </a>
              <a id="flex_65" class="pt-3 pb-3 pr-8 pl-8
border rounded-xl" href=""> Права </a>
            </div>
            <p class="mt-5">Розмір ручки</p>
            <div class="flex gap-6 item__flex mt-4">
              <a id="curve_71" class="pt-3 pb-3 pr-8 pl-8
border rounded-xl item__btn item__span--active" href=""> L3
(4 3/8") </a>
              <a id="curve_65" class="pt-3 pb-3 pr-8 pl-8
border rounded-xl" href=""> L5 (4 5/8") </a>
            </div>
            <p class="mt-5 mb-5"></p>
            <div class="flex gap-28 content-center
item__container__btn">
              <div class="item__flex">
                <a id="quantityMinus" class="pt-3 pb-3 pr-
8 pl-8 border rounded-xl" href=""></a>
              </div>
              <a class="item__flex__counter"
id="quantityCounter" href=""></a>
            </div>

            <a id="quantityPlus" class="pt-3 pb-3 pr-8
pl-8 border rounded-xl" href="">+</a>
          </div>
          <a class="item__container__btn__buy rad"
href="##42324#"> Купити  </a>
        </div>
        <div class="flex flex-col gap-8 mt-6">
          <div class="flex flex-col gap-4">
            <h4 class="item--span">AeroStrike Vortex
98 — Точність та швидкість у кожному русі</h4>
            <p>
              AeroStrike Vortex 98 — це ракетка для
гравців, які цінують контроль, чіткість ударів і
максимальну маневреність. Завдяки вдосконаленій
аеродинамічній конструкції та технології Vortex Frame, ця
модель забезпечує потужну швидкість замаху та чудову
стабільність навіть під час найдинамічніших розіграшів.

```

Головка розміром 98 кв. дюймів дарує більшу точність і відчуття м'яча, що робить Vortex 98 ідеальним вибором для гравців з агресивним стилем гри, які хочуть повністю контролювати темп на корті.

```

    </p>
  </div>
  <div>
    <h4 class="mb-4 item--span">Ключові
особливості:</h4>
    <p>●<span class="item--
span">Аеродинамічна рама з технологією Vortex
Frame</p>
    <p>●<span class="item--
span">Підходить для просунутих гравців, які прагнуть до
ідеальної точності</p>
    <p>●<span class="item--span">Висока
стабільність і швидкість замаху</p>
    <p>●<span class="item--span">Ідеально
підходить для гравців середнього та високого рівня</p>
  </div>
</div>
</div>
</div>
</section>

```

```

  <div class="form-container" id="form-container"
style="display: none;">
    <div class="form-content">
      <div>
        <button class="close-btn" id="close-
btn">&times;</button>
      </div>
      <form id="order-form">
        <div>
          <label for="name">Ім'я</label>
          <input type="text" id="name" name="name"
placeholder="Введіть ім'я" required>
        </div>
        <div>
          <label for="surname">Прізвище</label>
          <input type="text" id="surname"
name="surname" placeholder="Введіть Прізвище" required>
        </div>
        <div>
          <label for="phone">Номер телефону</label>
          <input type="text" id="phone" name="phone"
placeholder="+38(0____)____-____" required>
        </div>
      </form>
      <div id="form-message"></div>
      <button type="submit">Надіслати</button>
    </div>
    <div id="form-message"></div>
  </div>
  <footer id="contact">
    <div class="footer--padding">
      <div class="container--my">
        <div class="footer__navigation flex--gap">
          <li><a
href="categoryProduct.html">Каталог</a></li>

```



```

    if (!phoneInput.value) { showFieldError(phoneInput, "Це
поле обов'язкове"); errors++; }
    else if (!validatePhone(phoneInput.value)) {
showFieldError(phoneInput, "Формат: +380XXXXXXXXXX");
errors++; }
    else clearFieldError(phoneInput);

    if (errors > 0) {
        formMessage.textContent = 'Виправте помилки у
формі!';
        formMessage.className = 'error';
        return;
    }

    const productTitle =
document.querySelector('h2.title').textContent.trim() || 'Без
назви';
    const priceText =
document.querySelector('span.price').textContent.trim() || '0
грн';
    const productPrice = parseInt(priceText.replace(/[\^d]/g, ""),
10);

    const order = {
        type: 'catalog',
        title: productTitle,
        price: productPrice,
        customerName: nameInput.value.trim(),
        customerSurname: surnameInput.value.trim(),
        customerPhone: phoneInput.value.trim(),
        date: new Date().toLocaleString('uk-UA')
    };

    saveCustomOrder(order);

    fetch('https://67ffe0dcb72e9cfaf7262e72.mockapi.io/basket',
{
    method: 'POST',
    headers: {
        'Content-Type': 'application/json'
    },
    body: JSON.stringify(order)
})
.then(response => {
    if (!response.ok) {
        throw new Error('Помилка при відправленні
замовлення');
    }
    return response.json();
})
.then(data => {
    formMessage.textContent = 'Дякуємо за замовлення!';
    formMessage.className = 'success';
    document.getElementById('order-form').reset();
    setTimeout(function() {
        document.getElementById('form-
container').style.display = 'none';
        formMessage.textContent = "";
    }, 2000);
})
.catch(error => {
    console.error('Помилка:', error);
    formMessage.textContent = 'Не вдалося відправити
замовлення. Спробуйте пізніше.';
    formMessage.className = 'error';
});
});
</script>

```