

Міністерство освіти і науки України
Луцький національний технічний університет
Факультет комп'ютерних та інформаційних технологій
Кафедра інженерії програмного забезпечення

КВАЛІФІКАЦІЙНА РОБОТА
ЗА СТУПЕНЕМ ВИЩОЇ ОСВІТИ «БАКАЛАВР»

РЕАЛІЗАЦІЯ ВЕБ САЙТУ «ОНЛАЙН МАГАЗИН ДЛЯ
ОГЛЯДУ ТА ОФОРМЛЕННЯ ЗБРОЇ» З ВИКОРИСТАННЯМ
СТЕКУ PERN

IMPLEMENTATION OF THE WEBSITE «ONLINE STORE FOR
INSPECTION AND REGISTRATION OF WEAPONS» USING
PERN STACK

спеціальність 121 Інженерія програмного забезпечення
(шифр і назва спеціальності)

освітня програма «Інженерія програмного забезпечення»
(назва освітньої програми)

Виконав: здобувач вищої освіти
Групи ПІЗ-41

Лець Євген Михайлович


(підпис)

Керівник:

к.т.н., професор

Повстяна Юлія Славомирівна


(підпис)

Кваліфікаційну роботу

допущено до захисту

«4» 06 2023р.

к.т.н., доцент

Гарант освітньої програми:

Ліщина Наталія Миколаївна


(підпис)

Факультет комп'ютерних та інформаційних технологій
Кафедра інженерії програмного забезпечення
Ступінь вищої освіти: бакалавр
Галузь знань: 12 Інформаційні технології
Спеціальність: 121 Інженерія програмного забезпечення
Освітня програма: «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ
Завідувач кафедри

«02» 02 2023 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ВИЩОЇ ОСВІТИ

Лещю Євгену Михайловичу

1. Тема кваліфікаційної роботи: Реалізація веб сайту «Онлайн магазин для огляду та оформлення зброї» з використанням стеку PERN / Implementation of the Website «Online store for Inspection and Registration of Weapons» Using the PERN Stack

керівник роботи: к.т.н., доцент Повстяна Ю. С.

затверджені наказом закладу вищої освіти від «03» лютого 2023 р. № 80-05-35

2. Строк подання здобувачем роботи: 19.06.2023 р.

3. Вихідні дані до роботи Законопроект №5708 «Про обіг цивільної вогнепальної зброї», Документація API для Node.js, Документація Bootstrap-React, документація React.

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

Аналіз проблеми вільного обігу зброї сьогодні. Порівняння ситуації з вільним обігом зброї в інших країнах. Аналіз конкуренції. Аналіз, визначення вимог до розроблюваного програмного забезпечення та проектування програмного забезпечення. Детальний огляд інструментарію PERN стеку. Розробка та підключення бази даних. Методи та алгоритми розробки серверу та клієнту для сайту. Процес розробки серверу та клієнту для сайту.

5. Перелік графічного матеріалу: 13 рис; 27 лістингів; 1 діаграма

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис	
		завдання видав	завдання прийняв
Аналіз предметної області	Повстяна Ю. С.		
Специфікація вимог до розробленої системи	Повстяна Ю. С.		
Розробка об'єкта проектування	Повстяна Ю. С.		
Нормоконтроль	Повстяна Ю. С.		
Гарант ОП	Ліщина Н.М.		
Показник запозичень тексту		8,11%	
Академічна доброчесність	Яцук А.А.		

7. Дата видачі завдання «2» лютого 2023 р.

КАЛЕНДАРНИЙ ПЛАН

№ :№ 3/п	Назва етапів кваліфікаційної роботи бакалавра	Строк виконання етапів роботи	Примітка
1 1	Огляд літературних джерел по темі кваліфікаційної роботи бакалавра	05.02.2023 р.	Виконано
2 2	Аналіз проблеми розробки та впровадження об'єкту проектування	27.02.2023 р.	Виконано
3 3	Обґрунтування вибору шляхів, технологій (алгоритмів) і засобів вирішення поставленого завдання	10.03.2023 р.	Виконано
4 4	Розробка функціонально-структурної схеми роботи об'єкта проектування та проектування бази даних	17.04.2023 р.	Виконано
5 5	Практична реалізація об'єкта проектування та розробка бази даних	18.05.2023 р.	Виконано
6 6	Тестування та налагодження об'єкта проектування	01.06.2023	Виконано
7 7	Здача чистового варіанту кваліфікаційної роботи бакалавра на кафедрі	08.06.2023 р.	Виконано

Здобувач вищої освіти

(підпис)
(прізвище, ініціали)

Керівник кваліфікаційної роботи

(підпис)
(прізвище, ініціали)

Міністерство освіти і науки України
Луцький національний технічний університет

Рецензія
на кваліфікаційну роботу

Здобувач вищої освіти:

Лець Євген Михайлович

Тема: Реалізація веб сайту « Онлайн магазин для огляду та оформлення зброї» з використанням стеку PERN / Implementation of the Website « Online store for Inspection and Registration of Weapons » Using the PERN Stack.

Коротка характеристика кваліфікаційної роботи:

Робота складається з трьох розділів, протягом яких дипломантом здійснюється огляд поставленої перед ним проблеми, пошук шляхів її вирішення та опис їх реалізації. Для реалізації інтернет-магазину було використано наступні програмні засоби: база даних postgresSQL, програмна платформа для компіляції JavaScript коду — Node.js , фреймворк для розробки серверної частини — Express.js, бібліотека для створення інтерфейсу користувача — React.js.

Самостійні розробки і пропозиції автора:

Кандидат пропонує використати стеку PERN для створення онлайн магазину зі зброєю, що включає реалізацію системи логіну, фільтрації товарів за типами й брдами та пагінацію забезпечить зручну навігацію та безпеку. PERN гарантує ефективну роботу з базою даних та швидко розробку. Універсальність та масштабованість стеку підходить для проекту.

Практичне значення роботи:

Кандидат створив простий фулл стек проект, що може легко використовуватись будь-ким, як фундамент, для створення свого власного інтернет магазину.

Недоліки:

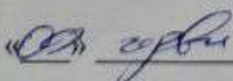
У роботі не виявлено новації – вона більш походить на альтернативу існуючим варіантам вирішення проблеми, проте слід помітити що складна структура роботи добре висвітлює вміння дипломанта.

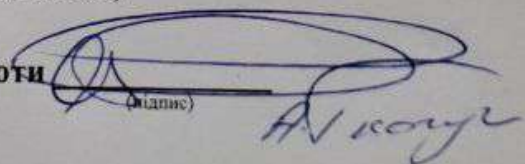
Загальний висновок:

Рівень якості виконаної роботи є задовільним. Усі поставлені завдання успішно й повноцінно виконані. Кваліфікаційна робота виконана на якісному рівні тож рекомендується до захисту та заслуговує на високу оцінку.

Рецензент: _____ доцент, к.т.н Ткачук А.А.
(прізвище, ім'я, по-батькові, посада)

Рецензент кваліфікаційної роботи

 2023 р.


(підпис)

АНОТАЦІЯ

Лець Є. М. Реалізація веб сайту «Онлайн магазин для огляду та оформлення зброї» з використанням стеку PERN.

Кваліфікаційна робота бакалавра ОП «Інженерія програмного забезпечення». Луцький національний технічний університет. Луцьк, 2023.

Кваліфікаційна робота бакалавра складається зі вступу, трьох розділів, висновків до кожного розділу, загальних висновків, списку використаних джерел та додатків.

У першому розділі досліджено предметну область де вивсвітлено ризики, переваги та недоліки теми, перспективи проекту. Другий розділ пояснює процес збору та оброки інформації мною та підготовку до створення проекту. Третій розділ пояснює процес розробки продукту з посиланнями на лістинги коду у додатках. У висновках узагальнено інформацію, відображену у розділах. Результат розробки може бути застосований в практичній діяльності, а саме продажу якогось продукту на сайті, не обов'язково зброї.

Ключові слова: стек, інтернет-магазин, сайт, сервер, клієнт, база даних, система CRUD.

ANNOTATION

Lets E. M. Creation of website «Online Store for Weapons Review and Purchase» using the PERN stack.

Qualification work of Bachelor's Degree in Software Engineering. Lutsk National Technical University. Lutsk, 2023.

The Bachelor's qualification work consists of an introduction, three chapters, conclusions for each chapter, a list of used sources, and appendices.

The first chapter explores the subject area, highlighting the risks, advantages, and disadvantages of the topic, as well as the project's prospects. The second chapter explains the process of gathering and processing information by me and preparing for the project's creation. The third chapter explains the product development process with references to code listings in the appendices. The conclusions summarize the information presented in the chapters. The result of the development can be applied in practical activities, namely the sale of any product on the website, not necessarily weapons.

Keywords: stack, online store, website, server, client, database, CRUD system.

ЗМІСТ

ВСТУП.....	6
РОЗДІЛ 1 АНАЛІЗ ОБІГУ ТА ПРОДАЖУ ЗБРОЇ В УКРАЇНІ ОНЛАЙН І ПОСТАНОВКА ЗАВДАНЬ НА КВАЛІФІКАЦІЙНУ РОБОТУ.....	7
1.1 Аналіз сучасного стану проблеми.....	7
1.2 Постановка завдання на кваліфікаційну роботу бакалавра.....	14
Висновки до розділу 1.....	15
РОЗДІЛ 2 СПЕЦИФІКАЦІЯ ВИМОГ ДО РОЗРОБЛЮВАНОЇ СИСТЕМИ.....	16
2.1 Аналіз, визначення вимог до розроблюваного програмного забезпечення та проектування програмного забезпечення.....	16
2.2 Вибір засобів, методів та алгоритмів вирішення поставленого завдання....	21
Висновки до розділу 2.....	25
РОЗДІЛ 3 РОЗРОБКА ПРОГРАМНОГО ПРОДУКТУ – «ОНЛАЙН МАГАЗИН ДЛЯ ОГЛЯДУ ТА ОФОРМЛЕННЯ ЗБРОЇ» З ВИКОРИСТАННЯМ СТЕКУ PERN.....	27
3.1 Практична реалізація об’єкта програмування.....	27
3.2 Тестування та налагодження інформаційно-комп’ютерної системи.....	42
3.3 Розробка бази даних.....	45
Висновки до розділу 3.....	48
ВИСНОВКИ.....	49
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	50
ДОДАТКИ.....	51

ВСТУП

Актуальність теми проявляється в тому, що сучасний світ, завдяки швидкому розвитку технологій та Інтернету, спосіб здійснення покупок та комерційних операцій зазнав значних змін. Інтернет-магазини стали невід'ємною складовою сучасного ринку, надаючи споживачам зручність, доступність та безліч варіантів вибору. Впровадження та розвиток інтернет-магазинів привнесли значні зміни в традиційну роздрібну торгівлю, відкривши нові можливості як для підприємців, так і для споживачів.

Метою дослідження є аналіз та розробка основних функціональних можливостей інтернет-магазину, який буде працювати на базі повного стеку технологій PERN (PostgreSQL, Express.js, React.js, Node.js).

Об'єктом дослідження даної кваліфікаційної роботи є інтернет-магазин для огляду та оформлення зброї.

Предметом дослідження – є технології та методи розробки сайтів з використанням стеку PERN.

Важливим аспектом розробки є використання інструментів технологічного стеку PERN, які надають широкі можливості для реалізації функціональності інтернет-магазину. У даній дипломній роботі будуть детально розглянуті та продемонстровані базові можливості кожної компоненти стеку.

Результати даної роботи внесуть вагомий внесок у розуміння феномену інтернет-магазинів і нададуть рекомендації для підприємців, які мають намір створити свій власний інтернет-магазин зі схожим асортиментом товарів, а також для споживачів, які планують здійснювати покупки в Інтернеті.

РОЗДІЛ 1

АНАЛІЗ ОБІГУ ТА ПРОДАЖУ ЗБРОЇ В УКРАЇНІ ОНЛАЙН І ПОСТАНОВКА ЗАВДАНЬ НА КВАЛІФІКАЦІЙНУ РОБОТУ

1.1 Аналіз сучасного стану проблеми

Кожна людина сьогодні має свій особистий спосіб зв'язку – телефон, або приватний комп'ютер: дома, на роботі. Люди, через свої «гаджети»: спілкуються, заводять стосунки, працюють, будують бізнес і кар'єру, купляють та продають речі. Загалом усі веб-застосунки, та веб-сторінки зосереджені на економії часу для користувачів. Те що змушувало вас вийти з дому, сісти у громадський транспорт і кудись відправитись – зараз, завдячуючи інтернет-технологіям, можна зробити в один-два кліки мишкою.

Звичайно в інтернет магазині є свої, великі переваги, порівняно з фізичними. Повторюсь, для початку – людині, банально, не прийдеться з крісла вставати для того, щоб знайти й купити те, що вона хоче. Користувач просто замовить кур'єра, який доставить пакунок прямо до дверей.

Також важливо виділити простоту та точність знаходження потрібного товару. Сьогодні кожен сайт для продажу різних товарів має можливість фільтрації контенту для користувачів за певними тегами, брендами тощо. Якщо на сайті є те, що тобі потрібно – тобі достатньо просто обрати бажані характеристики предмету, й сервер видасть тобі те, що ти хотів.

Великою перевагою для покупців є також і система оплати. Клієнти можуть використати будь-яку валюту, що автоматично буде конвертуватись банком за курсом, або взагалі – заплатити криптовалютою.

Звичайно й на стороні продавця є багато переваг. Оновляти продукцію, оперувати її цінами через просту панель в рази легше, як бігати по гігантській площі гіпермаркету й шукати потрібний предмет, щоб прикріпити на нього бирку про знижку або акцію. Працівник просто заходить до спеціальної панелі на сайті, до якої

мають доступ лише члени компанії та додає, або видаляє потрібні товари, міняє їхній статус та тому подібне.

Сьогодні купити й продати в інтернеті модна усе – навіть зброю.

25 червня 2021 року до Верховної ради України був внесений законопроект №5708 «Проект Закону про право на цивільну вогнепальну зброю», що передбачав, як можна зрозуміти з назви, носіння вогнепальної зброї цивільними особами в Україні. Та вже 13-го липня цей законопроект отримав «альтернативний» варіант №5708-1 [1].

Зі слів одного з авторів законопроекту №5708-1 народного депутата від партії «Голос», Андрія Шараськіного, їхній документ представляв інтереси пересічних українців, тоді як законопроект №5708, що є «основним», відображає інтереси «лобі МВС» у парламенті [1].

У законопроекті №5708, що був прийнятий у першому читанні поділяють усю вогнепальну зброю лише на п'ять категорій, а саме:

- категорія А – автоматична вогнепальна зброя;
- категорія В – травматична зброя;
- категорія С – короткоствольна вогнепальна зброя, крім травматичної;
- категорія D – довгоствольна вогнепальна гладкоствольна зброя (мисливська);
- категорія Е – довгоствольна вогнепальна нарізна та комбінована вогнепальна зброя.

Пневматична зброя, калібром до 4,5 міліметрів і швидкістю польоту бойової частини до 100 м/сек і, звичайно, револьвери під патрон Флореба, зброєю не вважаються.

У законопроекті згадуються й категорії, що не входять до цивільного обороту зброї та є забороненими. Як приклад можна навести ті, що створені з матеріалів, які не можуть вловлюватись металодетектором, перероблені з іншої категорії, калібром понад 12,7 міліметрів для нарізної зброї та 27 міліметрів для гладкоствольної,

револьвери. Також під заборону попадають нарізні дула (стволи) до гладкоствольної стрілецької зброї.

Право на володіння зброєю залежить від категорії. Зброєю категорії «А», тобто автоматичною вогнепальною, пропонується заборонити фізичним особам взагалі.

Ось вже зброєю категорії «В», травматичною, зможуть володіти люди професія яких пов'язана з ризиком для здоров'я, або життя, наприклад: журналісти, співробітники силових структур, судді тощо. На сьогодні такі правила уже дійсні.

Зброєю категорії «D», мисливською, можна буде володіти фізичним особам, що досягли 18-ти років, а категорія «Е», самозарядні гвинтівки з магазином не більше 10 набоїв, можуть володіти люди з 25-ти років.

Окремо виділяють пістолети, що відносять до категорії «С». Вони будуть доступні спортсменам, які досягли 18-ти років, та що беруть участь не рідше, як два рази на рік у спортивних змаганнях всеукраїнського та міжнародного рівнів.

Між тим пістолети, як і зараз, можуть бути нагородою для фізичної особи.

Та законопроект №5708 просто легалізує обіг цивільної вогнепальної зброї, а запропонований альтернативний законопроект №5708-1 вже дає можливість на самозахист.

На сьогоднішній день сфера продажу зброї в інтернеті стрімко розвивається. За запитом у Google «Купити зброю в Україні» ви з легкістю отримаєте десятки результатів.

Сайти з продажу зброї мають вигляд та оформлення звичайних інтернет магазинів, таких як Rozetka, Comfy, або будь-який інший. Єдина різниця зі звичайними онлайн магазинами така, що ви можете лише оформити замовлення на товар онлайн, а отримати на руки, або зареєструвати на себе ви повинні вже в фізичному магазині, або спеціальній установі. Тож можна зробити висновок, що такі сайти слугують, переважно, лише як онлайн каталог, де користувачі можуть побачити, що зараз є на складі у продавця, чого немає, а також забронювати товар, або замовити не виходячи з дому. Купити, як звичайний товар ви можете лише

травматичну зброю, що по закону зброєю як такою не є, та сувеніри, якщо вони там присутні.

Серед найпривабливіших продавців я виділю двох: «ІБІС» та «STVOL». Сайти цих двох платформ мають не велику відмінність, але вони найпопулярніші й пошукових відображає їх першими.

«ІБІС» представляє собою сайт насичений інформацією починаючи вже з головної сторінки. Користувачу пропонуються яскраві категорії, такі як: популярні товари, новинки, акції. Візуальний дизайн приємний, якісно підібрані кольори з лісової палітри прийдуть до душі будь-якому любителю полювання, або риболовлі (рис. 1.1).

Окрім зброї каталог товарів має категорії пов'язані з риболовлею, кемпінгом, іграшками та сувенірами та іншим.

Окремо хочеться виділити досконало оформлений елемент одиниці товару де вказано все, від ціни, оцінки й назви товару, до бренду, статусу й особливостей (знижки, статусу, що товар закінчується, та статусу, що товар лише для самовивозу).

На сторінці товару наявна уся потрібна інформація: опис, характеристики, відгуки. Також присутній такий важливий елемент, як відео де показується приклад користування зброєю, її тестування на полігоні тощо.

Реєстрація на сайті звичайна. Користувач може створити обліковий запис, або увійти з допомогою Google та Facebook. Кошик користувача також не має нічого особливого, він відображає товари, що користувач додає туди, та підраховує їхню загальну ціну.

Оформлення замовлення просте й не має нічого зайвого: Користувач вводить особисту інформацію – це якщо він не має зареєстрованого аккаунту, або просто оформлює замовлення користуючись інформацією свого, вже існуючого, облікового запису.

«STVOL», у свою чергу, вже має більш строгу палітру переважно з чорного та

білого кольорів. Головна сторінка має великий банер, список брендів-партнерів, особливі переваги сайту, категорії товару та спеціальні пропозиції. З виду сайт більше мілітаристський у порівнянні з попереднім (рис. 1.2).

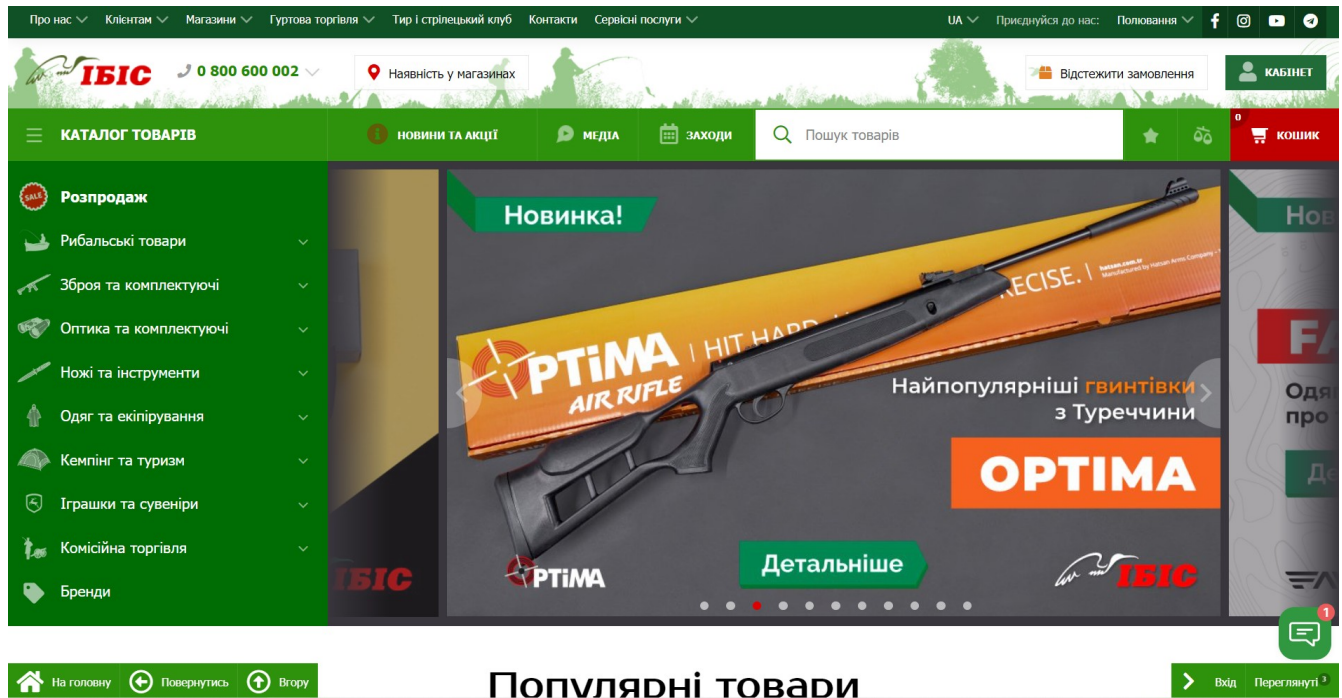


Рисунок 1.1 – Головна сторінка інтернет-магазину з продажу зброї «ІБІС» [2]

На відміну від магазину ІБІС сайт має більш вузький каталог товарів, що передбачає лише асортимент пов'язаний зі зброєю, або саму зброю.

На сайті є розгорнутий пошук по характеристикам продукції будь-якої категорії – це дуже круто.

Реєстрація не відрізняється процесом від сайту «ІБІС».

На сторінці товару присутня уся потрібна інформація, а також важливо виділити нагадування, що на певні товари потрібний дозвіл на користування. Про це згадується, також, при натисканні на клавішу «Купити».

Кошик виконує звичайні функції підсумовування продукції, яку хоче купити людина.

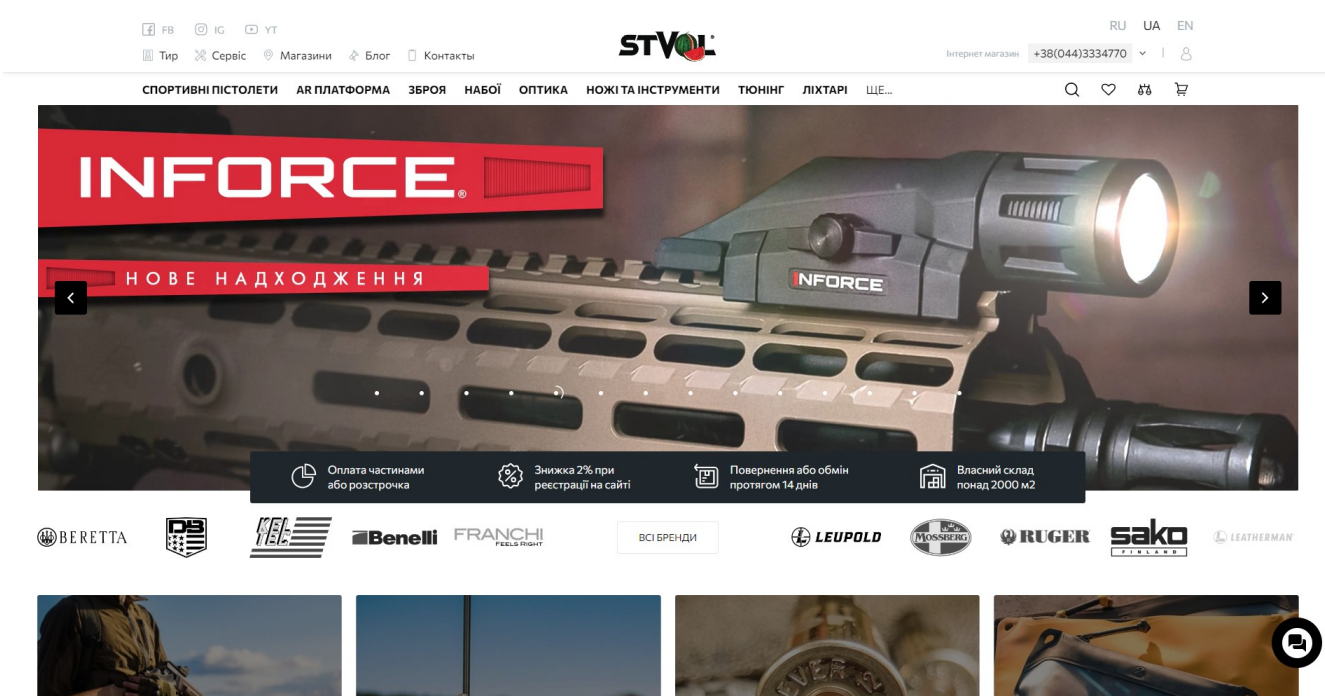


Рисунок 1.2 – Головна сторінка інтернет-магазину з продажу зброї «STVOL»[3]

Процес оформлення замовлення не має нічого кардинально відмінного від попередньо-описаного сайту «ІБІС».

Сьогоднішні реалії життя в Україні тяжкі та сумні. Усі громадяни держави страждають від збройної агресії росії – країни терориста, що знищує інфраструктуру держави та вбиває її беззахисних жителів.

Ось уже з перших слів можна побачити істину, причину потреби масової популяризації зброї в Україні – це все для самозахисту, як у приватних так і колективних цілях. На державу, що начинена стрілковою зброєю й населення якої має можливість захистити свій дім ніхто не буде нападати, так як прийдеться боротись не лише з армією, а з кожним. Навіть терористи, як росія, що знищують усе під ряд, при спробі збройного конфлікту будуть мати в сотню раз більше клопоту з збройними перепалками й захопленням територій.

Під час мирного життя в державі також стаються злочини де слабкі, мирні люди стають жертвами сильних і безвідповідальних, а так з можливістю мати при собі напівавтоматичну гвинтівку чоловік зможе захистити свій дім від нападу

озброєного грабіжника, або жінка зможе зупинити напад гвалтівника. Вогнепальна й травматична зброя може дати людині можливість зберегти своє життя.

Багато людей скажуть, що через наявність у кожного зброї підвищиться рівень злочинності й вбивств – вони будуть не праві. Рівень злочинності залежить від рівня життя й освіченості суспільства, а так навпаки, будь хто хто поспробує причинити комусь шкоду буде думати сім раз перед можливою смертю. Бо насправді, будемо чесні, у нас завжди є можливість причинити людині фізичну шкоду та вбити її, без будь-яких автоматів чи пістолетів, для цього потрібне лише бажання.

Також зброя може слугувати й можливою розвагою для людей, особливо для чоловіків. Люди більше будуть відвідувати полігони, тири, матимуть досвід користування різним особистим озброєнням – це навіть може скоротити програму навчання у закладах правоохоронних органів, армії, або інших силових структурах. Тим більше попит на таке дороге задоволення збільшить виробництво зброї в Україні, формування приватних виробників тільки зростатиме це підніме нас на світовій торговій арені по експорту та зробить додатковий дохід для держави. Знаючи талановитість нашого народу українські вироби будуть мати гігантській попит по усьому земному шару.

Потрібно зауважити, що розширення сфери торгівлі стрілецькою та холодною зброєю буде лишати доходів тіньові бізнеси зв'язані з контрабандою.

Вогнепальна зброя це річ не проста, процеси її передачі з рук в руки до волі не простий, а наслідки за не дотримання правил поводження з нею тяжкі. Тож процес її продажу, тим більше онлайн повинен передбачати усі можливі перевірки особистості та дотримання правил, це не товар як телефон, що можна просто відправити по «Новій пошті» і через два дні він буде в покупця. Продавець повинен давати товар з рук у руки попередньо перевірявши та узгодивши усі документи та інформацію, що підтвердять особу клієнта. Сайт повинен бути лише платформою для оформлення, а покупки забиратись лише у спеціалізованих пунктах видачі.

Також потрібно окремо виділити зберігання подібного асортименту. Набої різних калібрів вогненебезпечні, вони повинні зберігатись у сухому місці без доступу до прямого сонячного світла. Увесь товар повинен бути під строгим наглядом охорони, щоб нічого нікуди незаконно не зникло.

Якщо на сайті буде можливість прив'язки документів потрібно забезпечити якісний захист користувацьких аккаунтів. Паролі, двохрівнева аутентифікація. Підтримка зв'язку з замовником, також потрібна.

Гарантії також є невід'ємною складовою довіри, що повертатимуть людей до нашого магазину раз за разом. Налагодивши постачання від загально відомих та перевірених брендів ми зможемо гарантувати клієнтам, що зброя не підведе в відповідний момент, а набої не розірвуться в середині зброї.

Хтось може подумати, для чого новий сайт з продажу зброї, якщо вже існує стільки альтернатив – це для більшої популяризації та розвитку ринку.

З розвитком подібної сфери в Україні все більше людей будуть цікавитись можливістю самозахисту, колекціонерства, або чогось пов'язаного зі стрілецьким спортом, тому для людей потрібно більше альтернатив, кращі пропозиції, краща якість обслуговування.

1.2 Постановка завдання на кваліфікаційну роботу бакалавра

Суть проекту проста – створення «фулл стек» проекту, що буде виконувати функції інтернет-магазину для огляду, ознайомлення та покупки продукції мілітаристського характеру для цивільного користувача.

Цілі проекту які повинні бути досягнуті в результаті виконання атестаційної роботи можна поділи на такі пункти:

- аналіз предметної області, можливостей та перспективи, ризиків та переваг подібної діяльності в Україні;

- ознайомлення з подібними, уже представленими проектами, що отримали популярність серед людей;
- вибір інструментарію та створення моделі проєкту;
- підключення бази даних зі записами даних користувачів, товарів, паролів тощо та налаштування зв'язків між цими елементами таблиці;
- створення Backend для обробки й передачі інформації між клієнтом, що відвідує сайт, та продавцем з використанням;
- формування та оздоблення FrontEnd частини сайту для оптимізації подачі інформації для користувача з використанням;
- забезпечення безпеки даних.

Висновки до розділу 1

У першому розділі було аналізовано сучасний стан проблеми де було описано усі нюанси роботи з темою поводження та обігу зброї в Україні, згадано та враховано зміни внесені до законодавства України у законопроекті №5708 та його альтернативі, що передбачає обіг зброї серед цивільних осіб держави. Було проведено визначення уже наявних конкурентів у сфері інтернет торгівлі подібним асортиментом товарів. Проаналізовано ризики. Також у розділі зазначено потреби та обґрунтування реалізації подібного проєкту. Описано суть та цілі проєкту.

РОЗДІЛ 2

СПЕЦИФІКАЦІЯ ВИМОГ ДО РОЗРОБЛЮВАНОЇ СИСТЕМИ

2.1 Аналіз, визначення вимог до розроблюваного програмного забезпечення та проектування програмного забезпечення

Для моделювання програмної системи та опису її елементів я обрав UML діаграму розгортання, так як ця діаграма чудово підходить для короткого опису та моделювання системи роботи веб-сторінки. На ній зображаються обчислювальні вузли, під час роботи програми, компоненти та об'єкти, що виконуються в цих вузлах. Компоненти відповідають представляють робочих екземплярів одиниць коду.

Перше за все від інтернет-магазину користувачі очікують побачити той продукт, що вони шукають. Тобто зайшовши на сайт людина повинна розібратись де вона й як вона повинна знайти те що хоче – чи то через поле пошуку, чи то через фільтр продуктів.

Основними орієнтирами для мене, щоб створити привабливий сайт є:

- привабливий та мінімалістичний інтерфейс виконаний у кольорах, що резонують між собою, щоб не злякати людину через незвичайний контраст, або дивне розміщення основної навігації по сторінкам;
- фільтр, і/або поле пошуку для знаходження користувачем потрібного йому продукту;
- можливість реєстрації, та збереження даних користувача для подальшого логіну до сайту;
- «корзина» для збереження і підрахунку загальної цінності продукту обраного користувачем на сайті, для подальшої, його, покупки;
- забезпечення панелі адміністратора для користувачів особливим статусом (персоналу сайту) для легкого оперування даними бази даних (додавання/видалення) продуктів, брендів, типів тощо на сайті.

Так як товар у цьому інтернет магазині до волі специфічний сама магазин повинен мати ліцензію на продаж зброї. Також потрібно проводити усіх інших правил, пов'язаних з продажем зброї, зокрема, проводити перевірку клієнтів на наявність ліцензії на носіння та користування зброєю.

Окрім цього, інтернет-магазин з продажу зброї повинен мати у своєму асортименті товарів тільки легальну зброю, яка відповідає всім вимогам законодавства. Крім того, такий магазин повинен мати відповідні сертифікати на продукцію, яку він продає, та надавати гарантії на свої товари.

Основні функціональні вимоги до такої системи можна сформулювати наступним чином:

- 1) реєстрація користувачів – система повинна мати можливість реєструвати нових користувачів, які бажають купувати зброї;
- 2) пошук зброї по типу й бренду – система повинна мати можливість шукати інформацію про зброї, які продаються в магазині, включаючи опис, ціну, фотографії;
- 3) сторінка товару – для повного відображення усіх якостей товару;
- 4) додавання товарів до кошика – користувачі повинні мати можливість додавати зброї до свого кошика під час перегляду товарів.

Нижче буде наведено загальну специфікацію вимог до інтернет магазину з продажу зброї:

– безпека:

- 1) сайт повинен забезпечувати безпеку відносно інформації про продукти і користувачів. Наприклад, необхідно забезпечити захист від атак злоумисників та викрадення даних;
- 2) магазин повинен дотримуватися усіх законів та правил, пов'язаних з продажем зброї в Україні;
- 3) потенційні покупці повинні отримувати товар лише у фізичних магазинах, попередньо показавши усі потрібні документи.

– функціональність:

1) сайт повинен мати простий інтерфейс та зручну навігацію для користувачів;

2) покупці повинні мати можливість оплати замовлення з карти, або криптовалютою;

3) на сайті повинен бути опис кожного продукту, включаючи технічні характеристики, фотографії та відео.

– управління товарами:

1) адміністратор сайту повинен мати можливість додавати нові товари, оновлювати та видаляти існуючі;

2) наявність управління запасами для відстеження кількості наявних товарів.

– оплата:

1) сайт повинен мати безпечну систему оплати, щоб покупці могли здійснювати покупки онлайн;

2) потрібно вказати доступні методи оплати, наприклад, оплата карткою, електронними грошима чи платіжними системами.

– служба підтримки:

1) сайт повинен мати контактну інформацію, щоб покупці могли зв'язатися зі службою підтримки та отримати відповіді на свої запитання;

2) сайт повинен мати FAQ-розділ для зв'язку користувачів з персоналом.

Далі буде зображена модель елементів програмного додатку, елементів та зв'язків між ними.

На рисунку 2.1 зображено діаграму розгортання де показана структура сайту, серверної частини та бази даних.

Показана звичайна структура сайту, де користувач з допомогою базової периферії персонального комп'ютера (клавіатури, мишки, монітору) взаємодіє зі фронтендом – своїм робочим простором. Запити різного роду, від браузера,

надсилаються на сервер де обробляються та їхній результат повертається користувачу.

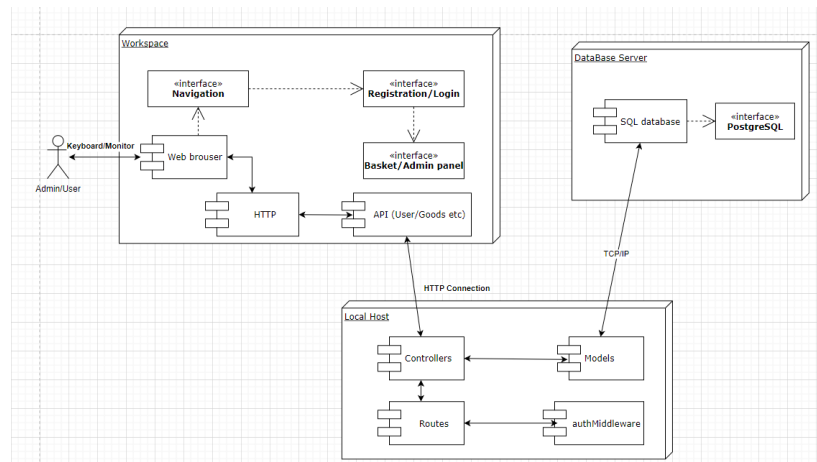


Рисунок 2.1 – Діаграма розгортання для проекту «PLUS guns»

Так само й працівники, що мають статус адміну через «панель адміністратора» на сайті, можуть подати запити до сервера та змінити базу даних додавши, або прибравши певні товари.

Зв'язок персоналу з користувачем буде відбуватись через соціальні мережі, або контактні дані (телефон/електронна пошта).

На діаграмі можна приблизно побачити, як працює сайт. На своєму пристрої користувач запрошує та подає запити до сервера взаємодіючи з різними навігаційними елементами сайту. Далі через API клієнт зв'язується з сервером та передає запит контролеру де в свою чергу контролер з роутером, в залежності від запиту, проводить маніпуляцію з тими даними – додає, видаляє та просто витягує один, або кілька елементів якоїсь таблиці. Роутер також перевіряє чи авторизований (правильність авторизації у випадку процесу входу до системи) користувач на сайті для виконання певної дії. Наприклад: користувач не зможе додавати або видаляти продукти з «кошик», або банально не зможе увійти до свого облікового запису.

Контролер взаємодіє з моделями де описані усі зв'язки та значення елементів і таблиць бази даних.

Тепер я можу приступити до розробки UI дизайну мого сайту.

Дизайн інтерфейсу користувача – це процес розробки інтерфейсу користувача для програмного забезпечення, сайтів, як на мобільні пристрої так і на персональні комп'ютери, при якому досягають максимально зручний та інтуїтивний спосіб їх використання користувачами [4].

Для виконання даного етапу роботи я буду використовувати онлайн засіб Figma – це спеціальний сайт для розробки користувацьких інтерфейсів, що доступний прямо у браузері [5]. Його принцип роботи простий, він подібний до якогось графічного редактора по типу Photoshop де користувач за допомогою блоків, або фігур (рис. 2.2) та шарів (рис. 2.3).

На рисунку 2.2 показано набір інструментів:

- оперування файлом для його збереження, завантаження тощо;
- курсор, для перетаскування, або збільшення розміру об'єктів;
- засіб для секторування об'єкту, з допомогою нього на об'єкт можна нанести сітку, розрізати його, або обрати секцію власного розміру;
- фігури й зображення;
- ручка та олівець;
- текст;
- ресурси, засіб для перетягування робочого простору та коментарі.

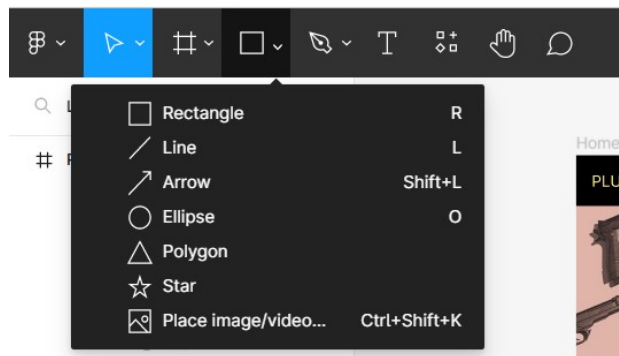


Рисунок 2.2 – Панель інструментів Figma

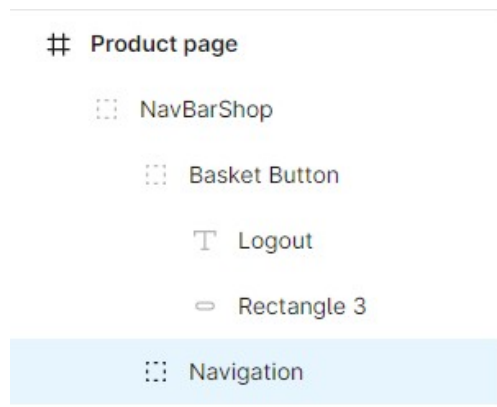


Рисунок 2.3 – шари Figma

Далі я приступив до дизайну сторінок (додаток А – Г).

На головній сторінці є банер з пропозицією переходу на сторінку магазину, банер зі брендами зброї, перелік деякої зброї. Звичайно навігаційна панель і футтер.

На сторінці магазину відображаються способи сортування (по бренду, типу) та товар.

На сторінці товару знаходиться уся інформація про товар (відгуки не реалізовані, присутня лише верстка).

У кошику користувач може побачити свої товари, що він хоче купити й загальну їхню вартість.

2.2 Вибір засобів, методів та алгоритмів вирішення поставленого завдання

Для реалізації поставлених задач буду використовувати PERN стек – це стек технологій, що складається з наступних інструментів:

PostgreSQL – об’єктно-реляційна база даних, яка забезпечує зберігання та організацію даних. Перевагою даної бази є те, що PostgreSQL підтримує великий набір вбудованих типів даних (21 тип). Крім того, користувач може самостійно створювати нові потрібні йому типи та програмувати для них свої механізми індексування за допомогою GIST [6].

Express – це легкий і гнучкий веб-фреймворк для Node.js, що використовується для написання серверної частини – бекенду. Його перевага в тому, що сам по собі мінімалістичний та простий, але також має й велику кількість плагінів, що підключаються [7].

React – це популярна бібліотека JavaScript для створення інтерактивних користувацьких інтерфейсів. Вона дозволяє розробникам створювати багатofункціональні, динамічні та легко-налаштовувані інтерфейси.

Node.js – це виконавче середовище JavaScript, яке дозволяє розробникам створювати серверні додатки на JavaScript. Він дозволяє створювати швидкі, масштабовані і надійні серверні додатки [6].

Я обрав PERN стек, так як використовується для створення повноцінних веб-додатків з динамічними інтерфейсами та високою продуктивністю. Таке поєднання технологій дозволяє створювати додатки з використанням однієї мови програмування (JavaScript), що робить стек легким для вивчення та налаштування.

Загалом поєднання таких елементів стеку корисне тим, як я вже згадував, що він використовує технології, для роботи з якими, потрібна лише одна мова програмування – JavaScript. Та звичайно кожен з цих елементів по своєму корисний та унікальний.

PostgreSQL – база даних, що має такі переваги:

1) надійність та стабільність: PostgreSQL має високу надійність та стабільність завдяки тому, що вона підтримує транзакції ACID (Atomicity, Consistency, Isolation, Durability), що дозволяє запобігти втраті даних в разі виникнення помилок або збоїв [6];

2) підтримка ряду типів даних: PostgreSQL має розширену підтримку різних типів даних, таких як географічні дані, JSON-документи та інші, що дозволяє легко працювати зі складними даними [6];

- 3) масштабованість: PostgreSQL може бути масштабована на великі об'єми даних та велику кількість одночасних користувачів, що робить її ідеальним вибором для підприємств та веб-додатків з великим навантаженням;
- 4) розширюваність: PostgreSQL дозволяє додавати розширення, що дозволяє просто розширювати функціональність бази даних відповідно до потреб проекту;
- 5) відкритість: PostgreSQL – це вільне та відкрите програмне забезпечення, що дозволяє мені безкоштовно використовувати та змінювати її відповідно до моїх потреб;
- 6) також можна виділити підтримку геоданих: PostgreSQL має вбудовану підтримку геоданих, що дозволяє розробникам працювати з географічними об'єктами та геоданими, забезпечуючи високу точність та ефективність роботи [6].

Тепер можна перейти до переваг Express.js:

- 1) легкість в освоєнні та використанні: Express.js має досить простий та зрозумілий API, що дозволяє розробникам швидко освоюватись та розробляти веб-додатки;
- 2) розширюваність: Express.js має велику кількість модулів та розширень, які дають можливість додавати нові функціональності до своєї програми;
- 3) підтримка middleware: Express.js має механізм middleware, що дозволяє додавати функціональність до запитів та відповідей, забезпечуючи більш гнучкий та розширюваний підхід до розробки [7];
- 4) висока швидкість: Express.js має високу швидкість роботи завдяки мінімальному набору функцій, який дозволяє оптимізувати виконання коду та забезпечувати швидке відгук сервера на запити;
- 5) підтримка RESTful API: Express.js добре підходить для розробки RESTful API, що є стандартом для створення веб-сервісів з використанням HTTP [7];

6) підтримка шаблонів: Express.js має вбудовану підтримку шаблонів, що дозволяє мені створювати статичні та динамічні веб-сторінки з використанням різних шаблонних мов, таких як Pug, EJS та інші;

7) підтримка WebSocket: Express.js має вбудовану підтримку WebSocket, що дає можливість розробникам створювати багатокористувацькі додатки в режимі реального часу [7].

Далі можна розглянути особливості React:

1) компонентна структура: React побудований на основі компонентної архітектури, це дає змогу створювати складні інтерфейси з використанням невеликих та простих компонентів [8];

2) висока продуктивність: React використовує Virtual DOM, що дозволяє ефективно оновлювати інтерфейс, зменшуючи кількість маніпуляцій з реальним DOM та покращуючи продуктивність додатка [8];

3) можливість використовувати JSX: React надає змогу користуватись JSX – це спеціальна мова розмітки, яка поєднує HTML та JavaScript. Це дозволяє писати компоненти в більш зрозумілому та інтуїтивно зрозумілому синтаксисі [8];

4) велика спільнота: React має гігантську та активну спільноту розробників, з якою можна швидко знаходити відповіді на питання та проблеми, а також ділитись знаннями та досвідом з іншими розробниками;

5) підтримка серверного рендерингу: React дозволяє розробникам створювати додатки з підтримкою серверного рендерингу, що покращує продуктивність додатка та оптимізує його для пошукових систем;

6) можливість використовувати з іншими бібліотеками та фреймворками: React можна використовувати в поєднанні з іншими бібліотеками та фреймворками – це дає ще більш розгорнуті можливості для створення застосунків різного роду;

7) повторне використання коду: у React можна створювати компоненти, які можуть бути використані більше одного разу, що заощаджує час та зменшує кількість коду, який потрібно писати.

Під кінець подивимось на переваги Node.js:

- 1) швидкість: Node.js заснований на движку V8, який використовується в браузері Google Chrome (найпопулярніший веб браузер сьогодні), що забезпечує високу швидкість виконання JavaScript коду. В результаті, Node.js дозволяє створювати швидкі та ефективні серверні додатки [9];
- 2) однорідність: Node.js дозволяє використовувати JavaScript на стороні клієнта та сервера, що забезпечує однорідність в розробці. Це означає, що розробники можуть використовувати одні й ті ж інструменти та технології на обох сторонах [9];
- 3) асинхронність: Node.js використовує асинхронний підхід до обробки запитів, що забезпечує високу продуктивність та масштабованість. Запити оброблюються без блокування виконання інших запитів, що дозволяє обслуговувати більшу кількість за один час;
- 4) масштабованість: Node.js дозволяє легко масштабувати додатки через клонування процесів або використовуючи технології, такі як кластеризація. Це дозволяє обслуговувати велику кількість запитів та забезпечує високу доступність додатку;
- 5) інструменти та бібліотеки: Node.js має велику кількість інструментів та бібліотек, що дозволяє швидко та ефективно розробляти серверні додатки. Той же самий Express.js, що використовується для розробки цього продукту.

Висновки до розділу 2

У другому розділі було проаналізовано вимоги до розроблюваного програмного забезпечення. Було описано процес проектування інтернет-магазину, показано та охарактеризовано систему та елементи продукту, що зображені на рисунку 2.1.

Також я представив UI дизайн інтернет магазину, що представляє собою майбутній вигляд фронтенд частини сайту. Та на завершення у розділі описані переваги інструментарію, що я обрав для виконання поставлених задач.

РОЗДІЛ 3

РОЗРОБКА ПРОГРАМНОГО ПРОДУКТУ – ІНТЕРНЕТ-МАГАЗИНУ «PLUS GUNS»

3.1 Практична реалізація об'єкта програмування

Перша, та головна задача практичної реалізації проекту було підключення та налаштування бази даних. Інструментом місця реалізації, як я уже згадував вище, стала СУБД й сервер – postgresSQL.

З початку я повинен занести на сервер дані для отримання доступу до окремої, потрібної мені, бази даних. Для цього я створюю файл з найменуванням «.env» де вказую: порт (на якому буде запускатись сервер), найменування бази даних, ім'я користувача (який оперує базою даних), пароль, вид хостингу, порт бази даних (так як я використовую postgresSQL я повинен вказувати порт саме цієї СУБД), секретний ключ. Далі приклад коду наведений у лістингу 3.1.

Лістинг 3.1 – Код файлу «.env»

```
PORT=5000
DB_NAME=plusguns_store
DB_USER=postgres
DB_PASSWORD=*****
DB_HOST=localhost
DB_PORT=5432
SECRET_KEY=*****
```

кінець лістингу 3.1

Далі я створюю файл для передачі даних з .env до ініціалізуючого коду, називаю цей файл – «db.js», описано у лістингу 3.2.

Лістинг 3.2 – Код файлу «db.js» мовою програмування JavaScript

```
const {Sequelize} = require(`sequelize`)
module.exports = new Sequelize(
  process.env.DB_NAME, // Назва бази даних
  process.env.DB_USER, // Ім'я користувача
  process.env.DB_PASSWORD, // пароль
  {
    dialect: `postgres`,
    host: process.env.DB_HOST, // хост
    port: process.env.DB_PORT // порт
  })
```

кінець лістингу 3.2

Потім у файлі «index.js» я прописую асинхронну функцію для ініціалізації підключення до бази даних, описно в лістингу 3.3. Де через try/catch автентифікую базу та синхронізуюю її. Під кінець, якщо все пройшло успішно, з'явиться повідомлення з портом старту (5000) – у іншому випадку консоль покаже повідомлення про помилку.

Лістинг 3.3 – Код файлу «index.js» мовою програмування JavaScript

```
app.use(errorHandler)
const start = async () =>{
  try {
    await sequelize.authenticate()
    await sequelize.sync()
    app.listen(PORT, () => console.log(`Server started on port ${PORT}`))
  } catch (e){console.log(e)}
}
start()
```

кінець лістингу 3.3

Підключити базу даних через консоль можна з допомогою такої команди: `npm run dev`. Її потрібно прописати перебуваючи у файлі серверу. Команд просто запускає код файлу `index.js`. Вона прописана у файлі «`package.json`»

Подальшим кроком є пов'язування таблиць між собою різним типом зв'язків – створення моделі бази даних.

У файлі `models.js` я поступово починаю описувати кожну таблицю та її колонки. Наприклад: таблиця Користувачів має чотири, різні види даних – колонки – це ідентифікаційний код (`id`), електронна пошта (`email`), пароль, роль. Ідентифікаційний код буде числом (`integer`) й виступатиме первинним ключем для таблиці, також він буде автоінкрементуватись, так як він повинен бути унікальним. Електронна пошта повинна бути унікальною строкою. Пароль – це просто строка. Роль також строка, яка має стандартне значення «`USER`».

Зразок коду з описом зв'язків у таблиці можна побачити у додатку Д.

Далі я прописую залежності для таблиць (один до одного, один до багатьох і так далі).

Наприклад кожен користувач може мати лише одну корзину, а одна корзина може мати багато зброї в собі (`BasketWeapon`) тощо, описано у лістингу у додатку Е.

Під кінець я просто екпортую усі таблиці, приклад наведено у лістингу 3.4.

Лістинг 3.4 – Код файлу «`models.js`» (експорт) мовою програмування JavaScript

```
module.exports = {
  User,
  Weapon,
  Type,
  Brand,
  Rating,
  TypeBrand,
  WeaponInfo,
  Basket,
  BasketWeapon
}
```

кінець лістингу 3.4

Далі я повинен створити middleware для користувачів, що буде перевіряти чи має користувач права на виконання певної маніпуляції з базою даних. Мені потрібно перевіряти чи користувач зареєстрований, наприклад: щоб додати якийсь товар до корзини й чи в користувача є роль адміністратора, наприклад: для додавання, або видалення якогось товару з магазину.

Якщо користувач авторизований, то в нього повинен бути особистий токен, тож потрібно перевірити його наявність, наведено у лістингу 3.5.

Лістинг 3.5 – Код файлу «checkRoleMiddleware.js» мовою програмування JavaScript

```
const token = req.headers.authorization.split(' ')[1]
if (!token) {return res.status(401).json({message:"Not registered"})}
```

кінець лістингу 3.5

Тепер потрібно перевірити чи в користувача достатньо прав на виконання якоїсь дії, наведено у лістингу 3.6.

Лістинг 3.6 – Код файлу «checkRoleMiddleware.js» мовою програмування JavaScript

```
const decoded = jwt.verify(token, process.env.SECRET_KEY)
if (decoded.role !== role) {
  return res.status(403).json({message:"Not enough permissions"})
}
req.user = decoded;
```

кінець лістингу 3.6

Тепер, коли сервер може перевіряти, хто хоче виконати якусь дію – я можу зробити роутери й контролери для API сайту. У контролерах буде записаний код, який буде проводити певні процеси з даними таблиці, а в роутерах вид запитів для функцій з роутерів. Вони будуть отримувати дані з API на фронтенді.

Мені потрібні контролери й роутери для корзини, брендів, типів, користувачів та самого товару. Усі вони створюються за подібним принципом, у них повинні бути

функції створення видалення, за потреби – функції знаходження одного, або кількох елементів.

З початку контролери. Як приклад візьму лише один контролер – товарний (зброї).

У товару у нас є багато характеристик: ім'я, ціна, рейтинг, зображення, тип, бренд та окремо: інформація, що записується в окрему таблицю. Так, якщо в зброї є інформація, то потрібно записувати дані й у таблицю з інформації, для окремої одиниці товару. Приклад коду реалізації створення одиниці товару, наведено у лістингу 3.7.

Лістинг 3.7 – Код файлу «weaponController.js» мовою програмування JavaScript

```

async create(req, res, next){
  try{
    let {name, price, brandId, typeId, info} = req.body
    const {img} = req.files
    let fileName = uuid.v4() + ".jpg"
    img.mv(path.resolve(__dirname, '..', 'static', fileName))
    const weapons = await Weapon.create({name, price, brandId, typeId, img:
    fileName});
    if(info){
      info = JSON.parse(info)
      info.forEach(i =>
        WeaponInfo.create({
          title: i.title,
          description: i.description,
          weaponId: weapons.id
        })
      )
    }
    return res.json(weapons)
  } catch (e) {
    next(ApiError.badRequest(e.message))
  }
}

```

кінець лістингу 3.7

Далі в контролері описана функція для сортування товарів за брендом і типом. Можливі варіанти сортування: немає сортування (!brandId && !typeId), сортування лише за брендом (brandId && !typeId), сортування лише за типом (!brandId && typeId) та сортування за типом і брендом (brandId && typeId), приклад наведено у лістингу 3.8.

Лістинг 3.8 – Код файлу «weaponController.js» мовою програмування JavaScript

```

async getAll (req, res) {
    let {brandId, typeId, limit, page} = req.query
        page = page || 1
        limit = limit || 9
    let offset = page * limit - limit
    let weapons;
    if(!brandId && !typeId){
        weapons = await Weapon.findAndCountAll({limit, offset})
    }
    if(brandId && !typeId){
        weapons = await Weapon.findAndCountAll({where: {brandId}, limit, offset})
    }
    if(!brandId && typeId){
        weapons = await Weapon.findAndCountAll({where: {typeId}, limit, offset})
    }
    if(brandId && typeId){
        weapons = await Weapon.findAndCountAll({where: {brandId, typeId}, limit,
        offset})
    }
    return res.json(weapons)
}

```

кінець лістингу 3.8

Наступною створюю функцію для отримання лише одного товару за ідентифікаційним номером, приклад наведено у лістингу 3.9.

Лістинг 3.9 – Код файлу «weaponController.js» мовою програмування JavaScript

```
async findOne(req, res){
    const {id} = req.params
    const weapon = await Weapon.findOne({
        where: {id},
        include: [{model: WeaponInfo, as: 'info'}]
    },
    )
    return res.json(weapon)
}
```

кінець лістингу 3.9

Також, під кінець – функція видалення, приклад у лістингу 3.10.

Лістинг 3.10 – Код файлу «weaponController.js» мовою програмування JavaScript

```
async delete(req, res){
    const id = req.params.id
    const weapon = await Weapon.destroy({where: {id}})
    return res.json(weapon)
}
```

кінець лістингу 3.10

З контролером завершили. Тепер можна створити роутер де я надам функціям методи обробки даних в таблиці. У мене є одна функція створення даних, дві з запитом на дані й одна для видалення даних. Тож нам потрібен один POST метод, два GET і один DELETE. Код роутера виглядає так, приклад – лістинг 3.11.

Контролери й роутери для інших таблиць створені за подібним принципом й їхній код згадувати я не буду, так як його до волі багато.

На цьому огляд створення фундаментальних файлів серверу завершено.

Лістинг 3.11 – Код файлу «weaponRouter.js» мовою програмування JavaScript

```
router.post('/', weaponController.create)
router.get('/', weaponController.getAll)
router.get('/:id', weaponController.getOne)
router.delete('/:id', weaponController.delete)
```

кінець лістингу 3.11

Далі потрібно створити усі потрібні мені сторінки, а це: сторінка магазину, для однієї зброї, головна, кошик і панель адміністратора. Після того, як я створив файли потрібно правильно налаштувати роутинг між сторінками, для навігації. Щоб було зручно я створю змінні, які буду використовувати для посилання на іншу сторінку – лістинг 3.12.

Далі я створю файл де посилання будуть поділятися на ті, які може відвідувати незареєстрований користувач і ті, які не може. Для зареєстрованих нам потрібні лише панель адміністратора й кошик.

Лістинг 3.12 – Код файлу «const.js» мовою програмування JavaScript

```
export const ADMIN_ROUTE = '/Admin'
export const LOGIN_ROUTE = '/login'
export const REGISTRATION_ROUTE = '/registration'
export const SHOP_ROUTE = '/'
export const BASKET_ROUTE = '/basket'
export const WEAPON_ROUTE = '/weapon'
```

```
export const MAIN_ROUTE = '/main'
```

кінець лістингу 3.12

Тож імпортувавши усі файли сторінок та змінні (лістинг 3.13), що я створював для них у файлі «const.js» я можу створити роутинг, приклад наведено в лістингу, в додатку Є.

 Лістинг 3.13 – Код файлу «routes.js» мовою програмування JavaScript

```
import Admin from "../pages/Admin"
import Auth from "../pages/Auth"
import Basket from "../pages/Basket"
import Shop from "../pages/shop"
import WeaponPage from "../pages/WeaponPage"
import Main from "../pages/Main"
import { ADMIN_ROUTE,
BASKET_ROUTE,
LOGIN_ROUTE, MAIN_ROUTE,
REGISTRATION_ROUTE, SHOP_ROUTE,
WEAPON_ROUTE } from
"./utils/const"
```

кінець лістингу 3.13

Подальшим кроком стало створення лейаута для магазину. Тобто мені потрібно було зробити навігаційну, статичну панель і місце де буде загрузатись контент з магазину, корзини і так далі. Для цього у файлі «App» я повинен розмістити створені мною компоненти навігаційної панелі й «футтера», а також компонент що буде підгружати динамічний компонент – application router, з допомогою вище представленого роутинга. У файлі «AppRouter», за аналогом файлу «routes», я прописую шляхи для користувачів і адміністраторів і окремо для незареєстрованих клієнтів, приклад у лістингу 3.14.

 Лістинг 3.14 – Код файлу «AppRouter.js» мовою програмування JavaScript

```
<Routes>
  {user.isAuth && authRoutes.map(({path, Component}) =>
    <Route key={path} path={path} element={<Component/>} exact/> )}
  {publicRoutes.map(({path, Component}) =>
    <Route key={path} path={path} element={<Component/>} exact/>)}
    <Route path='*' element={<Navigate to='/'/>} />
</Routes>
```

кінець лістингу 3.14

Тепер можна взятись за створення навігаційної панелі сайту. На такій панелі розміщуються посилання сторінок магазину, кнопки для логізації тощо. Тож з допомогою об'єктів з React-Bootstrap я створюю варіативну навігацію, знову ж таки, для зареєстрованих та не зареєстрованих людей.

Якщо заходить новий клієнт то бачить перед собою посилання на сторінки й кнопки для логіну, а якщо користувач входить в акаунт – у роботу вступає функція яка обирає спосіб відображення клавiш для входу в обліковий запис, адмін панель, кошик, приклад наведено у лістингу 3.15.

Лістинг 3.15 – Код файлу «AppRouter.js» мовою програмування JavaScript

```
{user.isAuth ?
  <Nav className="navigationBar">
    {user.isRole === 'ADMIN' ?
      <Button className="navButton" onClick={() =>
history(ADMIN_ROUTE)}>Admin panel</Button>
      :
      <Button className="navButton"
onClick={()=>history(BASKET_ROUTE)}>Cart</Button>
    }
    <Button className="navButton" onClick={() => logout()}>Logout</Button>
  </Nav>
  :
  <Nav className="navigationBar"><Button className="navButton" onClick={()
=> history(LOGIN_ROUTE)}>Autorisation</Button></Nav>
}
```

кінець лістингу 3.15

На кінець я створив футтер, але його код не буду демонструвати, так як це лише верстка – займе багато місця.

Усе, App файл готовий, для виду я додав анімацію завантаження сторінки (лістинг 3.16).

Лістинг 3.16 – Код файлу «App.js» мовою програмування JavaScript

```
useEffect(() => {  
  setTimeout(() => {  
    check().then(data => {  
      user.setRole(data.role)  
      user.setUser(data.id)  
      user.setIsAuth(true)  
    }).finally(() => setLoading(false))  
  }, 10)  
})  
if (loading) {  
  return <Spinner animation={"border"} />  
}  
return (  
  <BrowserRouter>  
    <NavBar />  
    <AppRouter />  
    <Footer /> </BrowserRouter>);
```

кінець лістингу 3.16

Тепер настав час для того щоб пов'язати BackEnd та Frontend.

Це можна зробити з допомогою API, що буде передавати дані в обидві сторони у форматі json [10].

Для початку я створюю змінні, що будуть присвоюватись авторизованим та незареєстрованим користувачам, також додаю вибірку для токена, приклад наведено у лістингу 3.17.

Тепер можна перейти до створення API для зв'язку з роутерами й контролерами сервера. Тобто мені потрібно зробити функції, що будуть брати дані з форм й віддавати їх серверу, а той у свою чергу їх обробить і запише в базу даних, або видалить тощо.

Отже для облікового запису треба зробити три функції: реєстрація, авторизація та перевірку авторизації, приклад наведено в лістингу, в додатку Ж.

Лістинг 3.17 – Код файлу «index.js» мовою програмування JavaScript

```
const $host = axios.create({
  baseURL: process.env.REACT_APP_API_URL})
const $authHost = axios.create({
  baseURL: process.env.REACT_APP_API_URL})
const authInterceptor = config => {
  config.headers.authorization = `Bearer ${localStorage.getItem('token')}`
  return config}
$authHost.interceptors.request.use(authInterceptor)
export{$host, $authHost}
```

кінець лістингу 3.17

Далі потрібно налаштувати API для типів, брендів та самого товару – зброї. Згідно з роутером на сервері ці в ці таблиці буде записуватись інформація та видалятись звідти. Також потрібно буде отримувати один, або кілька елементів для пагінації на сторінці магазину, або для редіректу на окремий товар. Приклад усіх API у додатку 3.

Завершальним етапом створення продукту буде верстка сторінок, налаштування на під'єднання клавiш і форм.

Для початку потрібно створити сторінки реєстрації та входу в обліковий запис. Обидві сторінки я об'єднаю в одну – створивши у формі двостороннє посилання. З початку користувач переходить на сторінку логіну, а якщо він не зареєстрований, але бажає це зробити він зможе перейти за посиланням у формі на другу форму – реєстрації і навпаки.

Тож на панелі інструментів є клавiша «Autorisation» (рис. 3.1), клікнувши на неї користувач попадає на сторінку авторизації й авторизується, або створює обліковий запис (рис. 3.2).

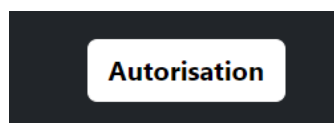


Рисунок 3.1 – Демонстрація клавiші авторизації на навігаційній панелі

Рисунок 3.2 – Демонстрація форми авторизації

Верстка елементів має багато функцій що в залежності до форми відображає елементи для логіну й реєстрації, приклад наведено у лістингу 3.18, там наведений код для клавiші підтвердження входу в аккаунт або реєстрації. Після потрібно створити панель адміністратора й можливість додавати й видаляти товари, їх бренди й типи. Адмін-панель буде складатись з кнопок, що будуть визивати модальні вікна, де адміністратор буде записувати параметри тих-чи-інших даних.

Лістинг 3.18 – Код файлу «Auth.js» мовою програмування JavaScript

```
const click = async () => {
  try{
    let data;
    if (isLogin) {
      data = await login(email, password)
    } else {data = await registration(email, password)}
    user.setRole(data.role)
    user.setUser(data)
    user.setIsAuth(true)
    history(MAIN_ROUTE)
  } catch (e) {
    alert(e.response.data.message)
  }
}
```

кінець лістингу 3.18

Приклад модального вікна створення я покажу на товарах (зброї) на лістингах 3.19 – 3.22.

Лістинг 3.19 – Код файлу «CreateWeapon.js» мовою програмування JavaScript

```
const {weapons} = useContext(Context)
const [name, setName] = useState("")
const [price, setPrice] = useState(0)
const [file, setFile] = useState(null)
const [info, setInfo] = useState([])
```

кінець лістингу 3.19

Для початку з допомогою `useState` я присвоюю статичний стан даних для змінних в які ми будемо записувати дані, що присвояться продукту [11].

Далі виводимо усі типи й бренди, з допомогою `UseEffect` для того, щоб вибрати в комбобоксі, який присвоювати продукту, а також функції для інформації (трейтів): додавання, видалення та зміни, приклад наведений у лістингу 3.20 [12].

Лістинг 3.20 – Код файлу «CreateWeapon.js» мовою програмування JavaScript

```
useEffect(() => {
    fetchTypes().then(data => weapons.setTypes(data))
    fetchBrands().then(data => weapons.setBrands(data))
}, [])

const addInfo = () => {
    setInfo([...info, {title: "", description: "", number: Date.now()}])
}
const removeInfo =(number) =>{
    setInfo(info.filter(i => i.number !== number))
}
const changeInfo = (key, value, number) => {
    setInfo(info.map(i => i.number === number ? {...i, [key]: value} : i ))
}
```

кінець лістингу 3.20

Тепер функція для формування даних продукту, приклад у лістингу 3.21.

Далі йде лістинг візуальної частини, який я також ну буду включати через його громісткість.

Модальні вікна для створення брендів та типів написані подібним образом і навіть простіше, так як там непотрібно вибирати інші дані, а просто записати одну строк в базу даних.

Лістинг 3.21 – Код файлу «CreateWeapon.js» мовою програмування JavaScript

```
const addWeapon = () => {
  const formData = new FormData()
  formData.append('name', name)
  formData.append('price', `${price}`)
  formData.append('img', file)
  formData.append('brandId', weapons.selectedBrand.id)
  formData.append('typeId', weapons.selectedType.id)
  formData.append('info', JSON.stringify(info))
  createWeapon(formData).then(data => onHide())
}
```

кінець лістингу 3.21

Видалення для усіх даних однакове – з початку з бази даних витягуються усі дані й видаляється одна, певна, яку вказав адміністратор, приклад наведено у лістингу 3.22.

Лістинг 3.22 – Код файлу «DeleteBrand.js» мовою програмування JavaScript

```
useEffect(() => { fetchBrands().then(data => weapons.setBrands(data))
}, [])
const delBrand = () => {
  deleteBrand(weapons.selectedBrand.id).then(data => {
    onHide()
  })
}
```

кінець лістингу 3.22

3.2 Тестування та налагодження інформаційно-комп'ютерної системи

У даному розділі буде проводитись тестування готової системи. Будуть перевірені такі елементи:

- логін та реєстрація;
- навігація;
- пагінація;
- коректна робота панелі адміністратора.

Перш за все перейшовши на головну сторінку тисну кнопку авторизації (рис. 3.3).

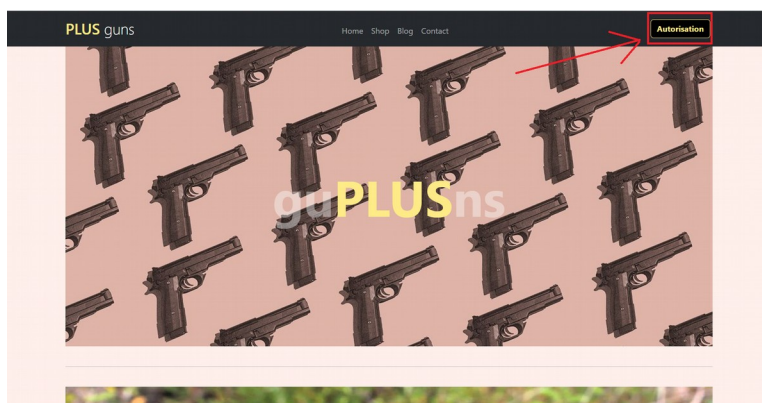


Рисунок 3.3 – Кнопка авторизації

При переході на сторінку логіну я можу вибрати вже зареєстрованого користувача (рис. 3.4). Це саме користувачі, що були зареєстровані цим комп'ютером з даного браузера, тобто інші люди цього не побачать.

Далі вибравши потрібний мені обліковий запис із роллю «user» я логінюсь. Готово – кнопка для авторизації змінилась кнопками кошика й виходу з аккаунту. (рис. 3.5).

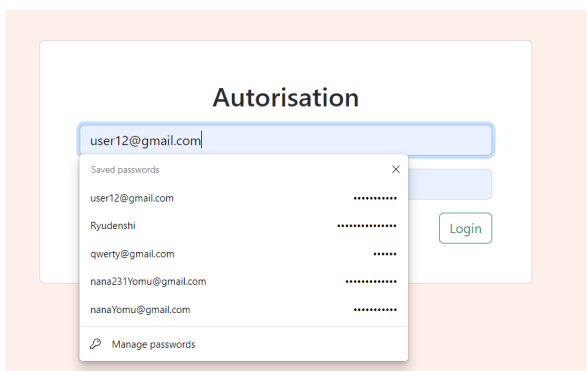


Рисунок 3.4 – Список зареєстрованих користувачів при логіні

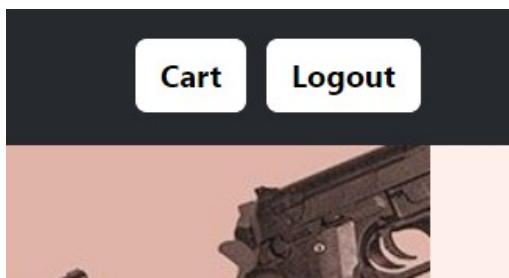


Рисунок 3.5 – Клавiшi кошику й логауту

Тепер я можу додати якийсь товар собі в кошик перейшовши на сторінку магазину, вибравши товар та натиснувши клавiшу «Add to cart» (рис. 3.6).

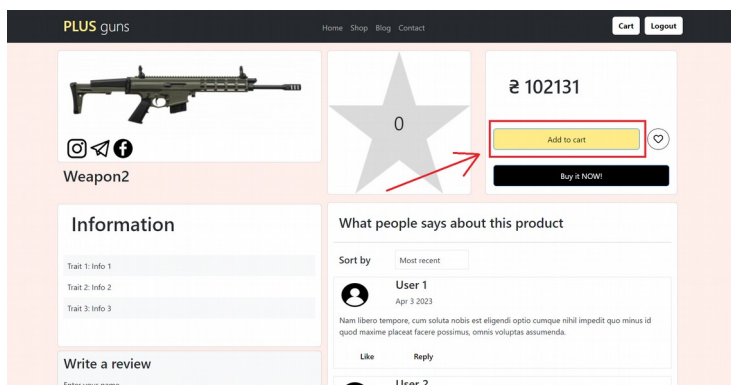


Рисунок 3.6 – Клавiша «Додати в кошик»

Після цього з'явиться модальне вікно з назвою товару та статусом, який підтвердить, що товар був доданий до корзини.

У кошику вже можна побачити список обраного товару та купити його, або очистити корзину (рис. 3.7).

Наступним кроком буде спроба створити новий бренд, тип та товар для магазину. Для цього потрібно вийти з облікового запису та зайти в аккаунт зі роллю адміністратора.

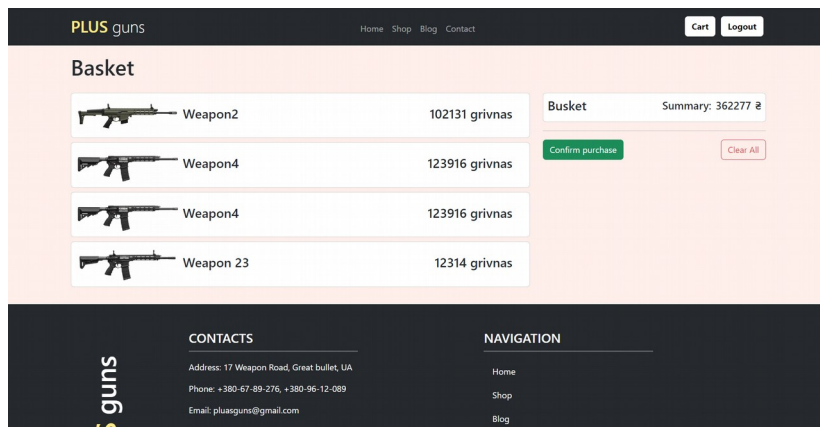


Рисунок 3.7 – Кошик користувача

Адміністратор має доступ до панелі, в якій він може оперувати товарами, брендами та типами. Тобто видаляти й додавати їх. Кнопка для переходу знаходиться на місці кошика. Перейшовши на панель клікаю «Add weapon» і з'являється модульне вікно (рис. 3.8).

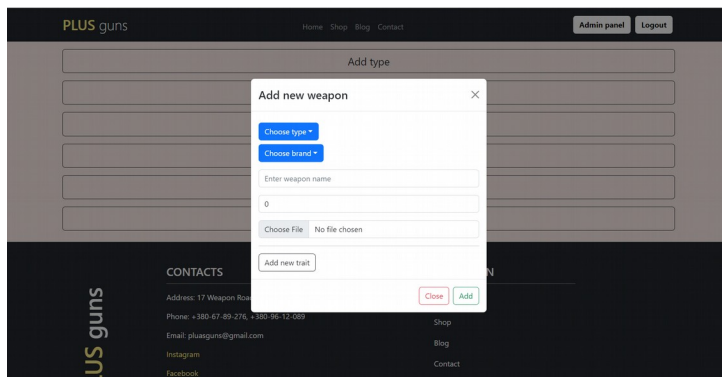


Рисунок 3.8 – Модульне вікно для додавання товару в панелі адміністратора

Обираю тип та бренд, вказую назву й ціну, додаю зображення та характеристики (рис. 3.9).

Товар створено, тепер можна перейти на сторінку магазину й знайти додану зброю, приклад у додатку Г.

The screenshot shows a web application interface for adding a new item. The background is a dark-themed page with a 'CONTACTS' section visible on the left. Overlaid on this is a light-colored form for adding a new firearm. The form contains the following elements:

- Two blue dropdown menus: 'Firearms' and 'Adams arms'.
- A text input field containing 'DDM4 V7 PRO K.223REM 18" M-LOK MFR 15'.
- A text input field containing '161480'.
- A file upload section with a 'Choose File' button and a filename 'tJHdDzK8FRrsNwUcpc...C7Pye1gxK5.jpeg.webp'.
- An 'Add new trait' button.
- A table of traits:

Weight	3.35kg	Delete
Muzzle thread	1 / 2-28 TPI	Delete
Barrel length, mi	457mm	Delete
Caliber	223Rem	Delete
Total length, mn	882mm	Delete
Barrel twist	1: 7	Delete
- At the bottom right, 'Close' and 'Add' buttons.

Рисунок 3.9 – Заповнена інформація про товар

Загалом усі потрібні функції працюють справно. Ніяких дефектів не виявлено.

3.3 Розробка бази даних

У даному підпункті буде описано структуру бази даних у СКБД postgresSQL, що використовується сервером для зберігання даних з сайту.

При налаштуванні зв'язків для бази даних я не використовував SQL синтаксис, тобто сам нічого не писав. База даних була структурована та налаштована через IDE, в файлі «models.js», про що я говорив вище, приклад лістингу файлу в додатку Е. Після того як я описав усі зв'язки в «models.js» через термінал я мігрував структуру й усі потрібні таблиці в СКБД були створені й налаштовані.

На рисунку 3.10 продемонстровані зв'язки між таблицями в базі даних виконана в онлайн застосунку для побудови діаграм «Diagrams.net», у «entity relationship» діаграмі [13].

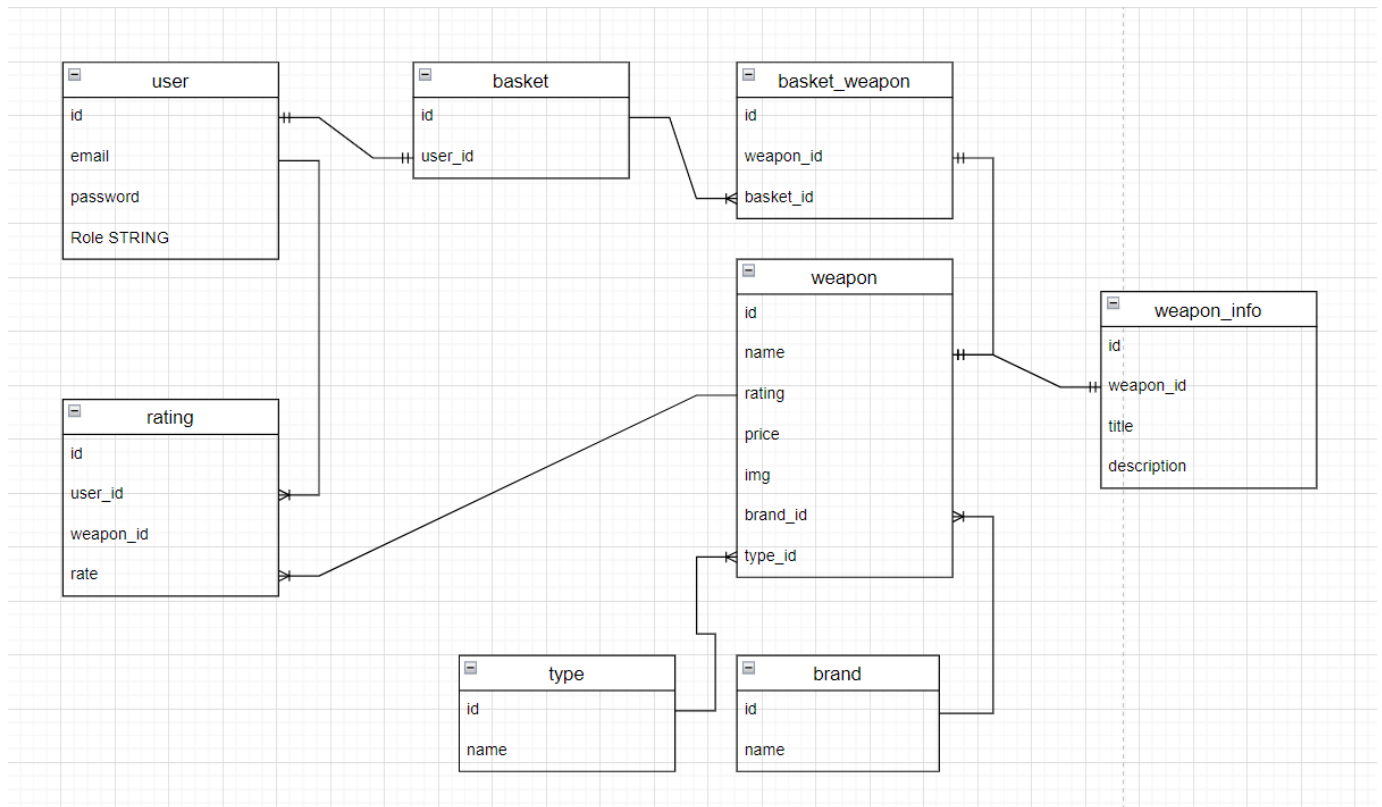


Рисунок 3.10 – Зв'язки між таблицями бази даних

Після подання запиту сервером на міграцію, javascript код вже в СКБД, перетворюється у SQL код. Коли відбулася міграція можна зайти в СКБД та перевірити код якоїсь таблиці, наприклад «weapons», приклад у лістингу 3.23.

Лістинг 3.23 – Код таблиці «weapons» мовою програмування SQL

```
-- Table: public.weapons

-- DROP TABLE IF EXISTS public.weapons;

CREATE TABLE IF NOT EXISTS public.weapons
(
    id integer NOT NULL DEFAULT nextval('weapons_id_seq'::regclass),
    name character varying(255) COLLATE pg_catalog."default" NOT NULL,
    price integer NOT NULL,
    rating integer DEFAULT 0,
    img character varying(255) COLLATE pg_catalog."default" NOT NULL,
    "createdAt" timestamp with time zone NOT NULL,
    "updatedAt" timestamp with time zone NOT NULL,
    "typeId" integer,
    "brandId" integer,
    CONSTRAINT weapons_pkey PRIMARY KEY (id),
    CONSTRAINT weapons_name_key UNIQUE (name),
    CONSTRAINT "weapons_brandId_fkey" FOREIGN KEY ("brandId")
        REFERENCES public.brands (id) MATCH SIMPLE
        ON UPDATE CASCADE
        ON DELETE SET NULL,
    CONSTRAINT "weapons_typeId_fkey" FOREIGN KEY ("typeId")
        REFERENCES public.types (id) MATCH SIMPLE
        ON UPDATE CASCADE
        ON DELETE SET NULL
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS public.weapons
    OWNER to postgres;
```

кінець лістингу 3.23

У лістингу 3.23 можна побачити, що усі мігровані зв'язки утворені в таблиці. Таким же способом було утворено кожную таблицю в базі даних та зв'язки між ними.

Висновки до розділу 3

У даному розділі було проведено тестування та налагодження інформаційно-комп'ютерної системи, що включало перевірку різних її елементів, таких як логін та реєстрація, навігація, пагінація та коректна робота панелі адміністратора. Під час тестування стало зрозуміло, що всі необхідні функції працюють справно і жодних дефектів не було виявлено.

Починаючи з авторизації, система дозволяє користувачам з легкістю входити у свої облікові записи та перехід на основну сторінку магазину. Крім того, користувачам надається можливість додавати товари до кошика та переглядати їх перед покупкою. Також відмінність ролей адміністратора була успішно створена, звання адміністратора дає можливість додавати нові бренди, типи та товари до магазину, а користувача – досліджувати сайт та оперувати бажаними товарами для покупки.

Загальним висновком після проведення тестування і налагодження інформаційно-комп'ютерної системи можна стверджувати, що вона працює стабільно та задовольняє всі вимоги, поставлені перед нею. Результати тестування свідчать про відсутність дефектів та підтверджують готовність системи до подальшого використання та удосконалення.

ВИСНОВКИ

Результати кваліфікаційної роботи бакалавра підтверджують повноту виконання поставленого завдання. Розроблений веб-сайт «Онлайн магазин для огляду та оформлення зброї» з використанням стеку PERN відповідає всім вимогам, включаючи аналіз проблеми поводження зі зброєю в Україні, розробку інтерфейсу та функціоналу, а також тестування та налагодження системи.

Робота привнесла нові результати, які знайшли відображення в статтях, винаходах та інших джерелах. Аналіз сучасного стану проблеми поводження зі зброєю в Україні та розробка інтернет-магазину сприяють покращенню ефективності та безпеки обігу зброї серед цивільних осіб. Рекомендації щодо подальшої роботи в даному напрямку включають поширення інформації про розроблену систему та співпрацю з відповідними органами та експертами для покращення законодавства.

Згідно з поставленими задачами було реалізовано систему реєстрації та логіну користувачів на сайті, типізацію товарів для фільтрованого пошуку, сторінка товару з зображенням загальної та точної інформації про продукт кошик для збереження та відображення списку обраних товарів для покупки.

Матеріали дослідження та розробки інформаційно-комп'ютерної системи мають практичне застосування в галузі обігу зброї та інтернет-торгівлі. Веб-сайт «Онлайн магазин для огляду та оформлення зброї» надає зручну та безпечну платформу для покупки та огляду зброї, сприяючи підвищенню доступності та контролю над обігом зброї.

Для подальшого вдосконалення розроблених досліджень та рішень рекомендується звернути увагу на можливості розширення функціоналу веб-сайту, такі як впровадження системи рейтингування товарів та вдосконалення системи адміністрування. Також важливо продовжувати вивчати та враховувати зміни в законодавстві щодо обігу зброї та вносити відповідні оновлення до системи.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Прийнято за основу проект Закону «Про право на цивільну вогнепальну зброю» // Верховна Рада України. – 2023. URL: <https://www.rada.gov.ua/news/Novyny/219896.html> (дата звернення 06.02.2023).
2. IBIS // Ibis. – 2023. URL: <https://ibis.net.ua/> (дата звернення 10.02.2023).
3. Stvol URL // Stvol. – 2023. URL: <https://stvol.ua/> (дата звернення 10.02.2023).
4. Все про професію UI/UX дизайнера // DAN.IT Education – 2023. URL: <https://dan-it.com.ua/uk/blog/vse-pro-profesiju-ui-ux-dizajnera/> (дата звернення 17.02.2023).
5. PLUS guns shop // Figma. – 2023. – URL: <https://www.figma.com/file/cW0EnansQSgKa6jjuWrD7Q/PLUS-guns-shop?type=design> (дата звернення 15.03.2023).
6. PostgreSQL URL // PostgreSQL – 2023. URL: <https://www.postgresql.org/> (дата звернення 25.02.2023).
7. Express.js // Express – 2023. URL: <http://expressjs.com/> (дата звернення 28.02.2023).
8. React URL // React – 2023. URL: <https://uk.legacy.reactjs.org/> (дата звернення 12.03.2023).
9. Node.js v20.2.0 documentation // Node – 2023. URL: <https://nodejs.org/docs/latest-v20.x/api/> (дата звернення 28.02.2023).
10. Built-in React APIs // React. – 2023. URL: <https://react.dev/reference/react/apis> (дата звернення 18.03.2023).
11. useState // React. – 2023. – URL: <https://react.dev/reference/react/useState> (дата звернення 16.03.2023).
12. useEffect // React. – 2023. – URL: <https://react.dev/reference/react/useEffect> (дата звернення 17.03.2023).
13. Diagram.net // Diagram.net. – 2023. URL: <https://app.diagrams.net/> (дата звернення 30.02.2023).

ДОДАТОК А

Головна сторінка (Home page)





ДОДАТОК В

Сторінка продукту (Product page)

Product page

PLUSguns

Home Shop Blog Contact

Cart

Logout

Choose your partner



Weapon 1

Calibre: 308 Win
Barrel length: 508 mm
Weight: 4.97 kg
Ammo capacity: 20
Weight: 4.97 kg
Country: USA





0

€ 74999.99

Add to Cart

Buy it NOW!

What people says about this product

Sort by: Most recent

User
Apr 3 2023

Nam libero tempore, cum soluta nobis est eligendi optio cumque nihil impedit quo minus id quod maxime placeat facere possimus, omnis voluptas assumenda.

Like Reply

User
June 28 2023

Nam libero tempore, cum soluta nobis est eligendi optio cumque nihil impedit quo minus id quod maxime placeat facere possimus, omnis voluptas assumenda.

Like Reply

User
Feb 12 2023

Nam libero tempore, cum soluta nobis est eligendi optio cumque nihil impedit quo.

Like Reply

Write a review

Write your name

ex: Evgen Lets

Your email

ex: evgenL@gmail.com

Rating

Choose Rating

Reviews

Write your review

PLUSguns

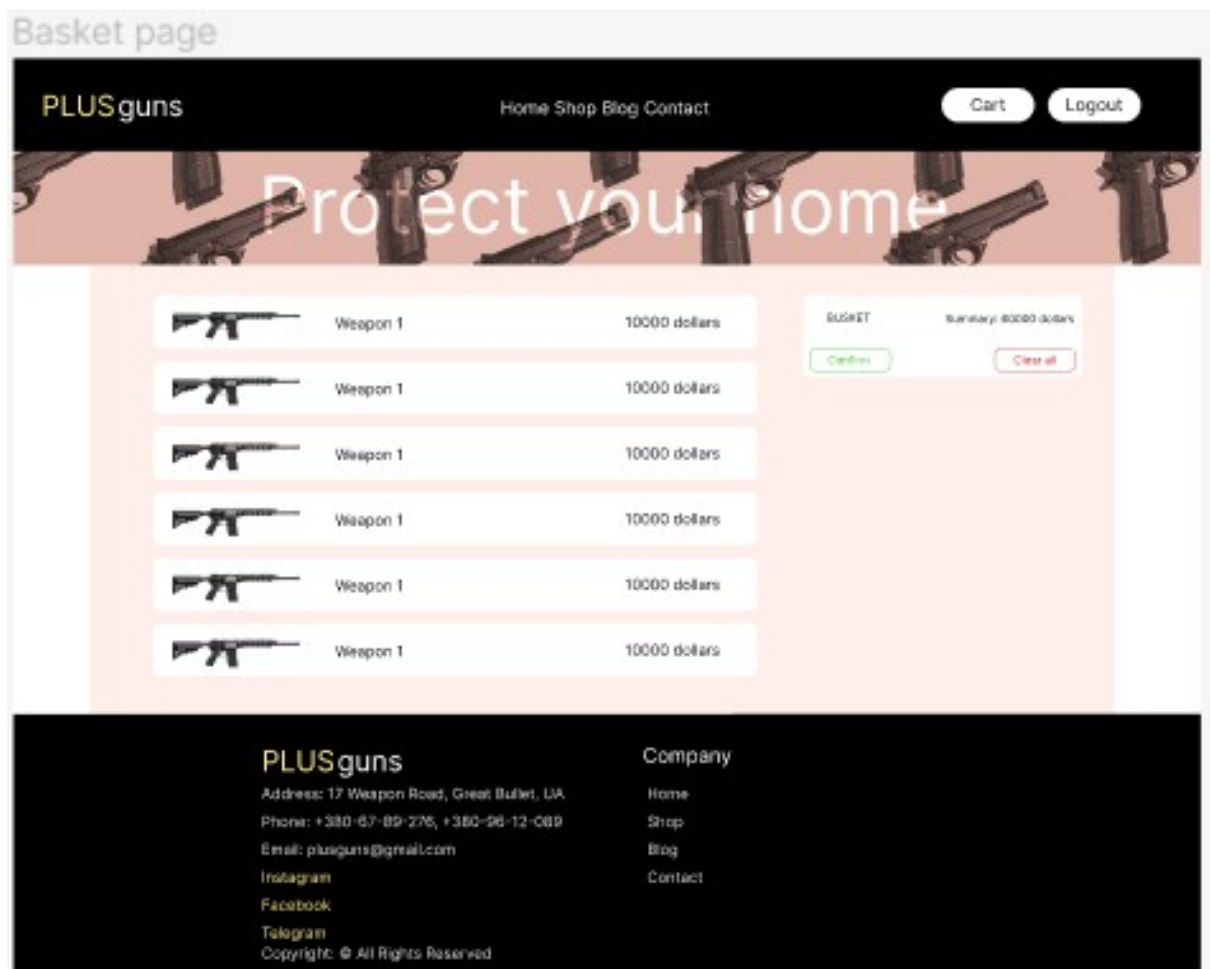
Address: 17 Weapon Road, Great Bullet, UA
Phone: +380-67-89-276, +380-96-12-089
Email: plusguns@gmail.com
Instagram
Facebook
Telegram
Copyright: © All Rights Reserved

Company

Home
Shop
Blog
Contact

ДОДАТОК Г

Кошик (Busket page)



ДОДАТОК Г

Додана зброя



**DDM4 V7 PRO K.223REM 18" M-
LOK MFR 15'**

Information

Weight: 3.35kg

Muzzle thread: 1 / 2-28 TPI

Barrel length, mm: 457mm

Caliber: 223Rem

ДОДАТОК Д

Лістинг Д.1 – Код файлу «models.js» (таблиці) мовою програмування JavaScript

```
const User = sequelize.define( 'user', {
  id: {type: DataTypes.INTEGER, primaryKey: true, autoIncrement: true},
  email: {type: DataTypes.STRING, unique: true},
  password: {type: DataTypes.STRING},
  role: {type: DataTypes.STRING, defaultValue: "USER"},
})

const Type = sequelize.define( 'type', {
  id: {type: DataTypes.INTEGER, primaryKey: true, autoIncrement: true},
  name: {type: DataTypes.STRING, unique: true, allowNull: false},
})

const Weapon = sequelize.define( 'weapon', {
  id: {type: DataTypes.INTEGER, primaryKey: true, autoIncrement: true},
  name: {type: DataTypes.STRING, unique: true, allowNull: false},
  price: {type: DataTypes.INTEGER, allowNull: false},
  rating: {type: DataTypes.INTEGER, defaultValue: 0},
  img: {type: DataTypes.STRING, allowNull: false},
})

const Brand = sequelize.define( 'brand', {
  id: {type: DataTypes.INTEGER, primaryKey: true, autoIncrement: true},
  name: {type: DataTypes.STRING, unique: true, allowNull: false},
})

const TypeBrand = sequelize.define( 'type_brand', {
  id: {type: DataTypes.INTEGER, primaryKey: true, autoIncrement: true},
})

const Rating = sequelize.define( 'rating', {
  id: {type: DataTypes.INTEGER, primaryKey: true, autoIncrement: true},
  rate: {type: DataTypes.INTEGER, allowNull: false},
})

const WeaponInfo = sequelize.define( 'weapon_info', {
  id: {type: DataTypes.INTEGER, primaryKey: true, autoIncrement: true},
  title: {type: DataTypes.STRING, allowNull: false},
  description: {type: DataTypes.STRING, allowNull: false},
})
```

```
const Basket = sequelize.define( 'basket', {  
  id: {type: DataTypes.INTEGER, primaryKey: true, autoIncrement: true},  
})  
  
const BasketWeapon = sequelize.define( 'basket_weapon', {  
  id: {type: DataTypes.INTEGER, primaryKey: true, autoIncrement: true},  
})
```

кінець лістингу

ДОДАТОК Е

Лістинг Е.1 – Код файлу «models.js» (зв'язки) мовою програмування JavaScript

User.hasOne(Basket)

Basket.belongsTo(User)

User.hasMany(Rating)

Rating.belongsTo(User)

Type.hasMany(Weapon)

Weapon.belongsTo(Type)

Brand.hasMany(Weapon)

Weapon.belongsTo(Brand)

Weapon.hasMany(Rating)

Rating.belongsTo(Weapon)

Weapon.hasMany(WeaponInfo, {as: 'info'})

WeaponInfo.belongsTo(Weapon)

Type.belongsToMany(Brand, {through: TypeBrand })

Brand.belongsToMany(Type, {through: TypeBrand })

Basket.hasMany(BasketWeapon)

BasketWeapon.belongsTo(Basket)

Weapon.hasMany(BasketWeapon)

BasketWeapon.belongsTo(Weapon)

кінець лістингу

ДОДАТОК Є

Лістинг Є.1 – Код файлу «routes.js» мовою програмування JavaScript

```
export const authRoutes = [  
  {  
    path: ADMIN_ROUTE,  
    Component: Admin  
  },  
  {  
    path: BASKET_ROUTE,  
    Component: Basket  
  }  
]  
export const publicRoutes = [  
  {  
    path: SHOP_ROUTE,  
    Component: Shop  
  },  
  {  
    path: LOGIN_ROUTE,  
    Component: Auth  
  },  
  {  
    path: REGISTRATION_ROUTE,  
    Component: Auth  
  },  
  {  
    path: WEAPON_ROUTE +('/:id'),  
    Component: WeaponPage  
  },  
  {  
    path: MAIN_ROUTE,  
    Component: Main  
  }  
]
```

кінець лісингу

ДОДАТОК Ж

Лістинг Ж.1 – Код файлу «userAPI.js» мовою програмування JavaScript

```
export const registration = async (email, password) => {
  try {
    const { data } = await $host.post('api/user/registration', {
      email,
      password,
      role: 'user',
    });
    localStorage.setItem('token', data.token);
    return jwt_decode(data.token);
  } catch (error) {
    throw new Error(error.response.data.message);
  }
};

export const login = async (email, password) => {
  try {
    const { data } = await $host.post('api/user/login', { email, password });
    localStorage.setItem('token', data.token);
    return jwt_decode(data.token);
  } catch (error) {
    throw new Error(error.response.data.message);
  }
};

export const check = async () => {
  try {
    const { data } = await $authHost.get('api/user/auth');
    localStorage.setItem('token', data.token);
    return jwt_decode(data.token);
  } catch (error) {
    throw new Error(error.response.data.message);
  }
};
```

кінець лістингу

ДОДАТОК 3

Лістинг 3.1 – Код файлу «weaponsAPI.js» мовою програмування JavaScript

```
export const createType = async (type) => {
  try {
    const { data } = await $authHost.post('api/type', type);
    return data;
  } catch (error) {
    console.error('Error creating type:', error);
    throw error;
  }
};

export const fetchTypes = async () => {
  try {
    const { data } = await $host.get('api/type');
    return data;
  } catch (error) {
    console.error('Error fetching types:', error);
    throw error;
  }
};

export const deleteType = async (id) => {
  try {
    const { data } = await $host.delete(`api/type/${id}`);
    return data;
  } catch (error) {
    console.error('Error deleting type:', error);
    throw error;
  }
};

export const createBrand = async (brand) => {
  try {
    const { data } = await $authHost.post('api/brand', brand);
    return data;
  } catch (error) {
    console.error('Error creating brand:', error);
    throw error;
  }
};
```

```

};
export const fetchBrands = async () => {
  try {
    const { data } = await $host.get('api/brand');
    return data;
  } catch (error) {
    console.error('Error fetching brands:', error);
    throw error;
  }
};
export const deleteBrand = async (id) => {
  try {
    const { data } = await $host.delete(`api/brand/${id}`);
    return data;
  } catch (error) {
    console.error('Error deleting brand:', error);
    throw error;
  }
};

export const createWeapon = async (weapon) => {
  try {
    const { data } = await $authHost.post('api/weapon', weapon);
    return data;
  } catch (error) {
    console.error('Error creating weapon:', error);
    throw error;
  }
};

export const fetchWeapons = async (typeId, brandId, page, limit = 5) => {
  try {
    const { data } = await $host.get('api/weapon', {
      params: { typeId, brandId, page, limit },
    });
    return data;
  } catch (error) {
    console.error('Error fetching weapons:', error);
    throw error;
  }
};
export const fetchOneWeapon = async (id) => {

```

```

    try {
      const { data } = await $host.get(`api/weapon/${id}`);
      return data;
    } catch (error) {
      console.error('Error fetching weapon:', error);
      throw error;
    }
  };

export const deleteWeapon = async (id) => {
  try {
    const { data } = await $authHost.delete(`api/weapon/${id}`);
    return data;
  } catch (error) {
    console.error('Error deleting weapon:', error);
    throw error;
  }
};

export const addToBasket = async (weaponId) => {
  try {
    const { response } = await $authHost.post('api/basket', weaponId);
    return response;
  } catch (error) {
    console.error('Error adding to basket:', error);
    throw error;
  }
};

etc.

```

кінець лістингу