

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»

Кафедра інженерії програмного забезпечення

КУРСОВА РОБОТА
(ПОЯСНЮВАЛЬНА ЗАПИСКА)

з дисципліни: «Backend-розробка»
на тему:
«Інтернет-магазин мотоциклів»

студента II курсу групи ВТ-21-1
спеціальності 121 «Інженерія
програмного забезпечення»
Шумського Олександра Вячеславовича
(прізвище, ім'я та по-батькові)

Керівник асис. каф. КН Фуріхата Д.В.

Дата захисту: " ____ " _____ 2023 р.
Національна шкала _____
Кількість балів: _____
Оцінка: ECTS _____

Члени комісії

_____	<u>О.Г. Чижмотря</u>
(підпис)	(прізвище та ініціали)
_____	<u>Д.В. Фуріхата</u>
(підпис)	(прізвище та ініціали)
_____	<u>І.А. Дмитренко</u>
(підпис)	(прізвище та ініціали)

Житомир – 2023

ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
Факультет інформаційно-комп'ютерних технологій
Кафедра інженерії програмного забезпечення
Освітній рівень: бакалавр
Спеціальність 121 «Інженерія програмного забезпечення»

«ЗАТВЕРДЖУЮ»
В.о. зав. кафедри ІПЗ
_____ А.В. Морозов
“ ” _____ 2022р.

ЗАВДАННЯ
НА КУРСОВУ РОБОТУ СТУДЕНТЦІ
Шумському Олександру Вячеславовичу

1. Тема роботи: Інтернет-магазин мотоциклів,
керівник роботи: асистент кафедри КН Фуріхата Д.В.
2. Строк подання студентом: “18”січня_2023р.
3. Вхідні дані до роботи: Розробити сайт продажу мотоциклів
4. Зміст розрахунково-пояснювальної записки(перелік питань. Які підлягають розробці)
 1. Постановка завдання
 2. Аналіз аналогічних розробок
 3. Алгоритми роботи програми.
 4. Опис роботи програми
 5. Програмне дослідження
5. Перелік графічного матеріалу(з точним зазначенням обов'язкових креслень)
 1. Посилання на Gitlab: <https://github.com/nonamecat19/MotorcycleShop>
6. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посади консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1,2,3	Фуріхата Д.В., асистент каф. КН		

7. Дата видачі завдання “ 15 ” вересня 2022 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів курсової роботи	Строк виконання етапів курсової роботи	Примітки
1	Постановка задачі	01.11.2022	виконано
2	Пошук, огляд та аналіз аналогічних розробок	07.11.2022	виконано
3	Формулювання технічного завдання	12.11.2022	виконано
4	Опрацювання літературних джерел	14.11.2022	виконано
5	Проектування структури	25.11.2022	виконано
6	Написання програмного коду	01.12.2022	виконано
7	Налагодження	05.01.2023	виконано
8	Написання пояснювальної записки	16.01.2023	виконано
9	Захист	18.01.2023	

Студент _____
(підпис)

Шумський О.В.
(прізвище та ініціали)

Керівник роботи _____
(підпис)

Фуріхата Д.В.
(прізвище та ініціали)

РЕФЕРАТ

Завданням на курсову роботу було розробка CMS-системи для сайту мотоциклів. Пояснювальна записка до курсової роботи на тему «Інтернет-магазин мотоциклів» складається з вступу, трьох розділів, висновків, списку використаної літератури та додатку.

Текстова частина викладена на 40 сторінках друкованого тексту.

Пояснювальна записка має 5 сторінок додатків. Список використаних джерел містить 10 найменувань і займає 1 сторінку. В роботі наведено 34 рисунків. Загальний обсяг роботи – 46 сторінок.

У першому розділі було обґрунтовано актуальність та причини створення програмного продукту.

У другому розділі проведено проектування і розробка програмного продукту.

У третьому розділі проведено тестування програмного продукту.

Висновок містить в собі результати виконаної роботи зі створення програмного продукту.

У додатку наведений лістинг розробленого програмного продукту.

Ключові слова: React, Flux, CMS, PHP, БАЗА ДАНИХ, ІНТЕРНЕТ МАГАЗИН, КОНТРОЛЛЕР, REST, API.

					Житомирська політехніка.23.121.14.000 – ПЗ			
Змн.	Арк.	№ докум.	Підпис	Дата	«Інтернет-магазин мотоциклів»	Літ.	Арк.	Аркушів
Розроб.		Шумський О.В.						
Керівник		Фуріхата Д.В.					4	47
Реценз.						ФІКТ, гр. ВТ-21-1[2]		
Н. Контр.								
Затверд.								

ЗМІСТ

ВСТУП	6
РОЗДІЛ 1. АНАЛІЗ ЗАДАЧІ.....	7
1.1 Аналіз задачі, засобів та методів її вирішення	7
1.2 Аналіз існуючого програмного забезпечення за тематикою курсового проекту	7
Висновки до першого розділу.....	13
РОЗДІЛ 2. ПРОЕКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	15
2.1 Проектування загального алгоритму роботи програми	15
2.2 Розробка функціональних алгоритмів роботи програми.....	16
2.3 Розробка програмного забезпечення:.....	22
Висновки до другого розділу	26
РОЗДІЛ 3. ОПИС РОБОТИ З ПРОГРАМНИМ ДОДАТКОМ ТА ЙОГО ТЕСТУВАННЯ	27
3.1 Опис роботи з програмним додатком	27
3.2 Тестування роботи програмного забезпечення.....	37
Висновки до третього розділу.....	38
ВИСНОВКИ.....	39
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	40
ДОДАТКИ.....	41
Додаток А.....	42
Додаток Б	43

ВСТУП

У курсовій роботі буде наведено процес створення CMS-системи інтернет-магазину продажу мотоциклів. На сьогоднішній день мотоцикл має статус дуже популярного транспорту, і є основним для багатьох людей через свою ціну а також зручність пересування у місті. Станом на 2002 рік кількість мотоциклів становила 205 млн.

Актуальність теми. обумовлена значним розвитком мотоциклів а також мотоспорту.

Метою розробки робота та створення програми у вигляді інтернет магазину мотоциклів

Об'єктом дослідження процес введення, обробки, зберігання, перевірки, покупки та пошук даних.

Мета роботи: розробка програми у вигляді сайту з використанням веб-технологій і концепції ООП у мові програмування PHP.

		Шумський О.В.			Житомирська політехніка.23.121.14.000 – ПЗ	Арк.
		Фуріхата Д.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		6

РОЗДІЛ 1. АНАЛІЗ ЗАДАЧІ

1.1 Аналіз задачі, засобів та методів її вирішення

Перед тим, як почати розробляти та писати програмний код, потрібно зробити аналіз та визначити порядок виконання завдання.

Задача на курсову роботу полягає у створенні сайту з продажу мотоциклів з використанням сучасної архітектури проекту. У процесі аналізу роботи було виділено наступні умови та функції, які має містити даний програмний продукт:

- Змодельовати загальний вигляд програми.
- Визначити, як саме користувач буде взаємодіяти з програмою.
- Розділити ролі користувачів сайту.
- Розробити можливість зміни даних сайту за допомогою інтерактивного інтерфейсу для адміністраторів та модераторів сайту.
- Розробити сторінки: головна, кошик, авторизація, реєстрація, перегляд товару разом з коментарями на ній, сторінки повідомлень про помилки, адміністраційна панель для редагування та панель з статистикою сайту.

Для розробки додатку були обрані WebStorm та PhpStorm. Проект розроблявся за допомогою HTML, CSS, PHP, TypeScript використовуючи фреймворк ReactJS. Також були використані додаткові бібліотеки, для полегшення процесу написання коду програмного забезпечення.

1.2 Аналіз існуючого програмного забезпечення за тематикою курсового проекту

При аналізі існуючого ПЗ за темою курсової роботи було виявлено багато різноманітних проектів. Кожен з них має унікальні принципи роботи та логіку. Кожен сайт має унікальний дизайн а також функціонал.

Першим сайтом для аналізу стане Mototek (рисунок 1.1.)[1]. Даний сайт зустрічає користувача приємним оку не надто яскравим дизайном. Зверху

		Шумський О.В.			Житомирська політехніка.23.121.14.000 – ПЗ	Арк.
		Фуріхата Д.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		7

розташовані посилання на категорії товарів. Можна одразу побачити товару на яку суму зараз міститься у кошику.

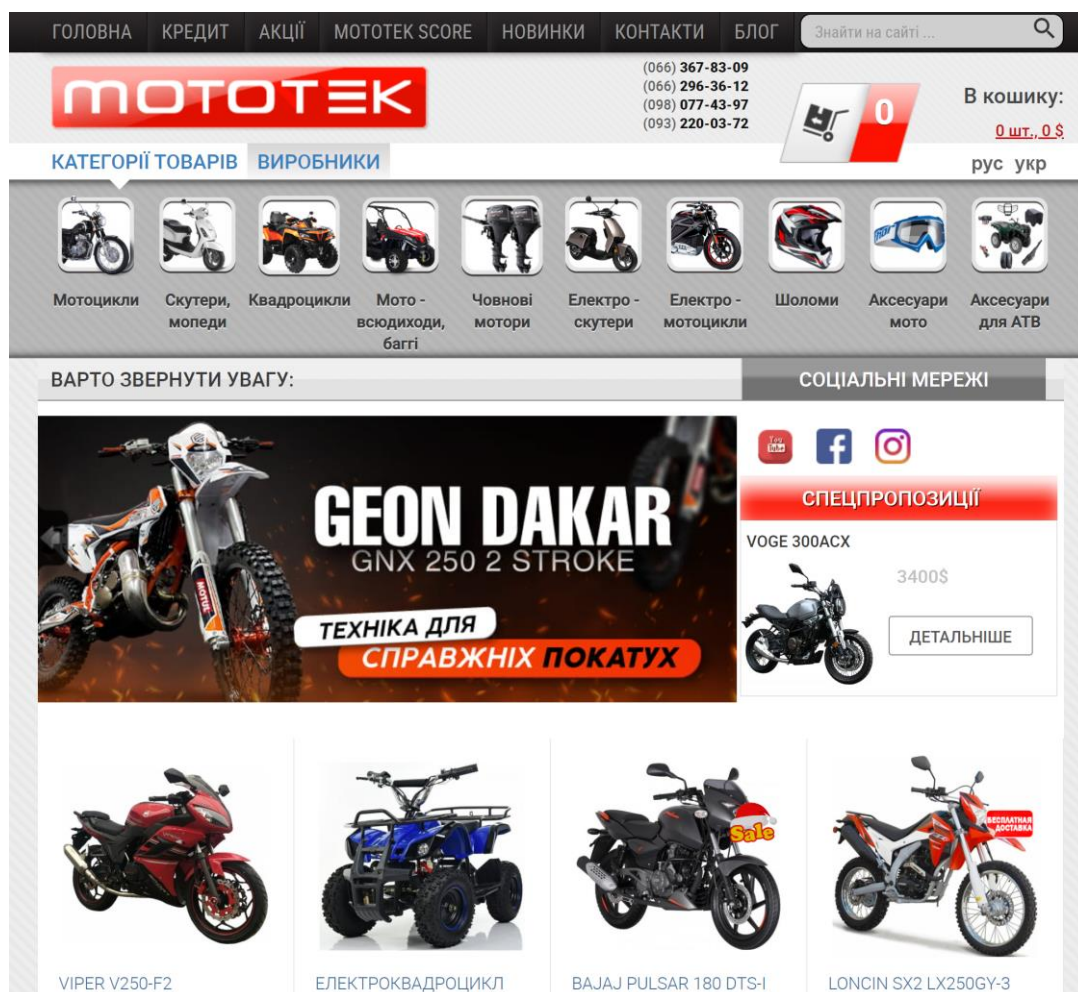


Рисунок 1.1. – Головна сторінка магазину Mototek

Прогорнувши нижче можна побачити асортимент товару (рисунок 1.2.). На найважливіших речах акцентується особлива увага за допомогою яскравого кольору.

		Шумський О.В.			Житомирська політехніка.23.121.14.000 – ПЗ	Арк.
		Фуріхата Д.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		8

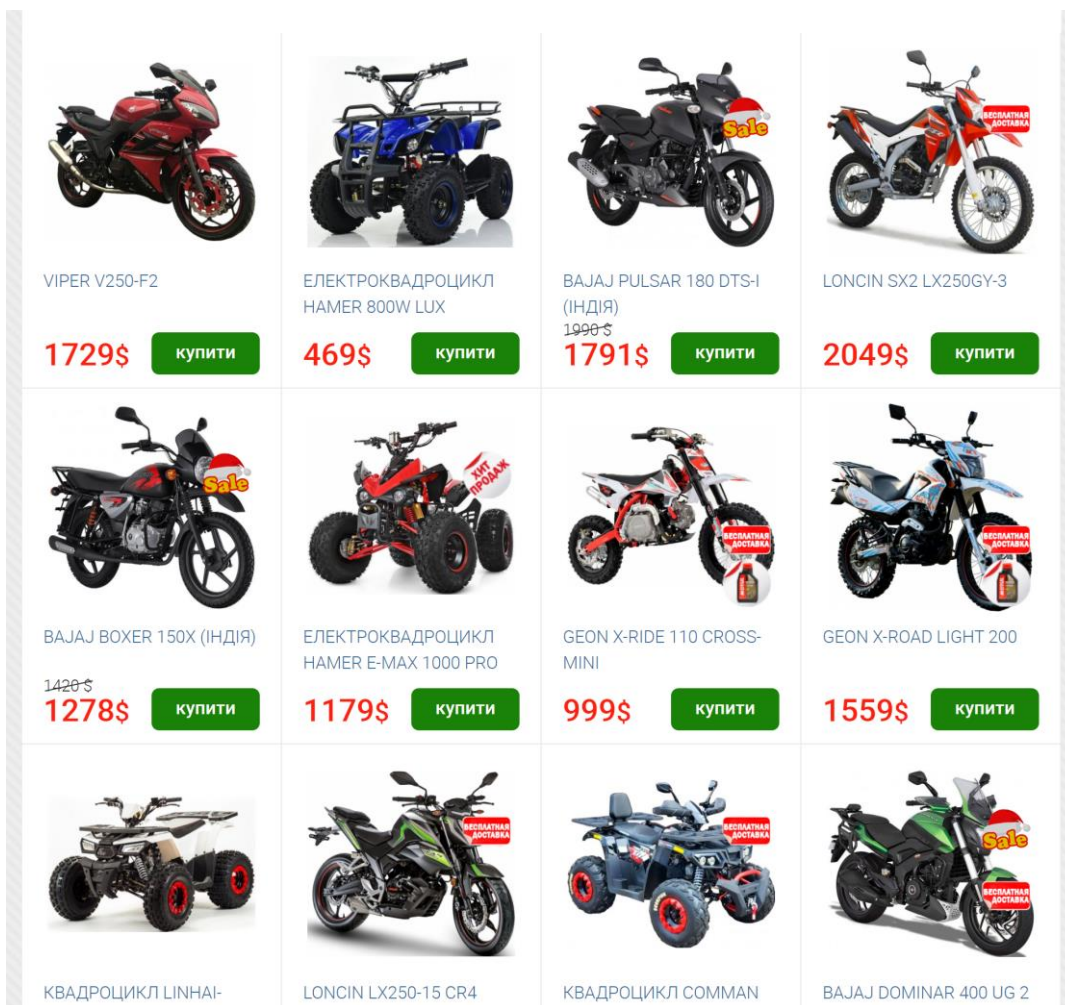


Рисунок 1.2. – Асортимент магазину Mototek

Таблиця 1.1

Плюси та мінуси магазину мотоциклів Mototek

Плюси	Мінуси
Акцент кольором на найважливіших речах	Оформлення сторінки товару має забагато деталей
Зручна навігація по сайту	Текс на сайті має різний шрифт що заважає нормальному сприйманню
Все оформлено в одному стилі	Деякі товари мають іконки поверх фото що перекривають основний контент картки

Наступним сайтом для аналізу буде сайт UA-Motors[2] (рисунок 1.3.). Головна сторінка сайту має приємний дизайн. Можна обрати необхідну категорію та перейти до вибору товару. На сайті можна сортувати товар за ціною, брендом та іншими даними. При наведенні на карту товару відображається більш розширена інформація про характеристики товару (рисунок 1.4.).

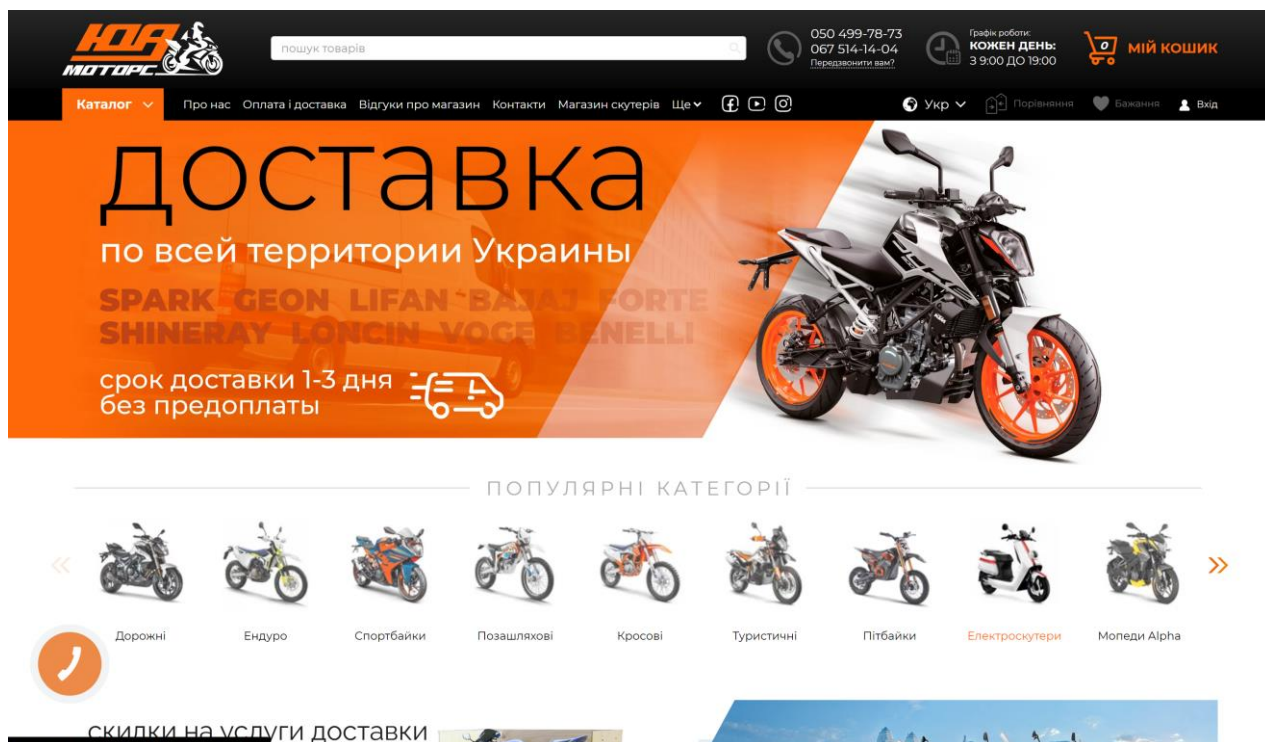


Рисунок 1.3. – Головна сторінка магазину UA-Motors

		Шумський О.В.			Житомирська політехніка.23.121.14.000 – ПЗ	Арк.
		Фуріхата Д.В.				10
Змн.	Арк.	№ докум.	Підпис	Дата		

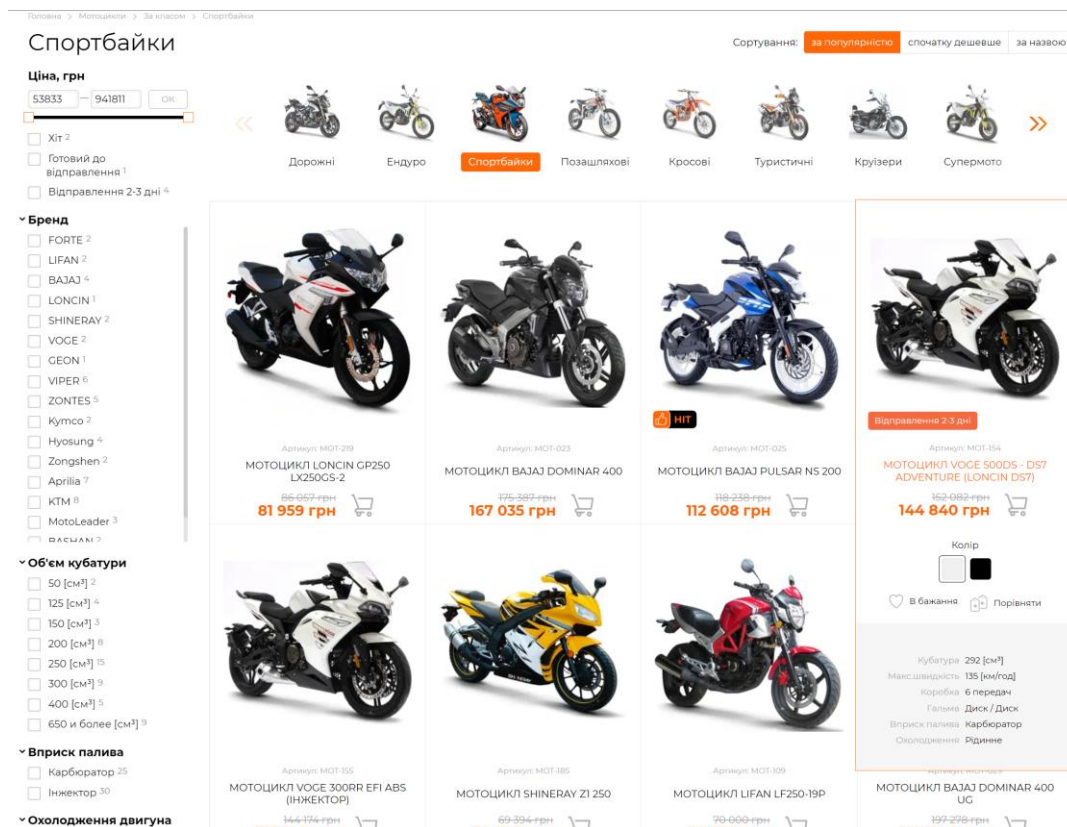


Рисунок 1.4. – Асортимент магазину UA-Motors

Таблиця 1.2

Плюси та мінуси магазину UA-Motors

Плюси	Мінуси
Красивий сучасний стиль оформлення	Не скрізь є українська локалізація
Багато сторінок з різним наповненням	Текст різним шрифтом
Красива головна сторінка	Деякі товари мають підписи в зоні фото що виділяється серед загальної стилістики сайту

Останнім сайтом для аналізу буде MOTODOM (рисунок 1.5.)[3]. Одразу при вході на сайт відкривається сторінка з вибором товару. Можна відфільтрувати товар за необхідними критеріями. На даному сайті найкраще

		Шумський О.В.			Житомирська політехніка.23.121.14.000 – ПЗ	Арк.
		Фуріхата Д.В.				11
Змн.	Арк.	№ докум.	Підпис	Дата		

серед вищезгаданих відображення карток товару: кольорова гама не ріже очі і, тим не менш, картка містить всю необхідну інформацію і навіть кнопку додавання в закладки та кнопку порівняння.

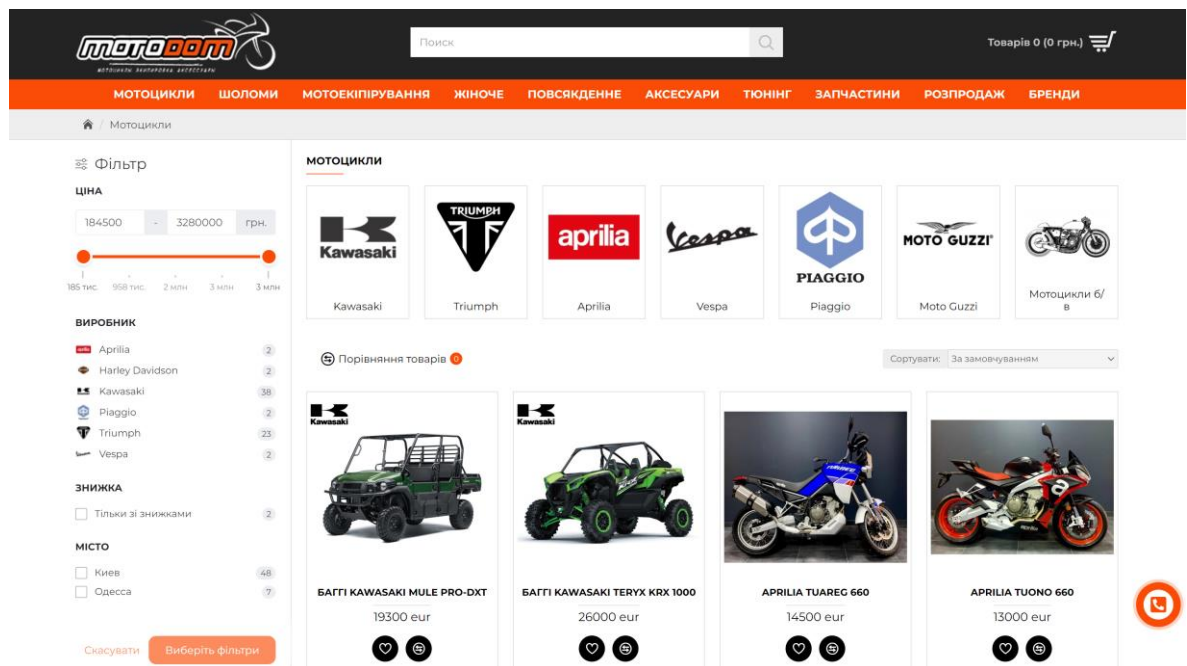


Рисунок 1.5. – Головна сторінка магазину MOTODOM

Контент на сторінці товару грамотно розміщений: довгий текст, такий як опис та характеристики товару відображається при натисканні на відповідний заголовок. Аналогічно до картки товару мотоцикл можна додати в закладки або порівняти з іншими мотоциклами (рисунок 1.6.).

		Шумський О.В.			Житомирська політехніка.23.121.14.000 – ПЗ	Арк.
		Фуріхата Д.В.				12
Змн.	Арк.	№ докум.	Підпис	Дата		

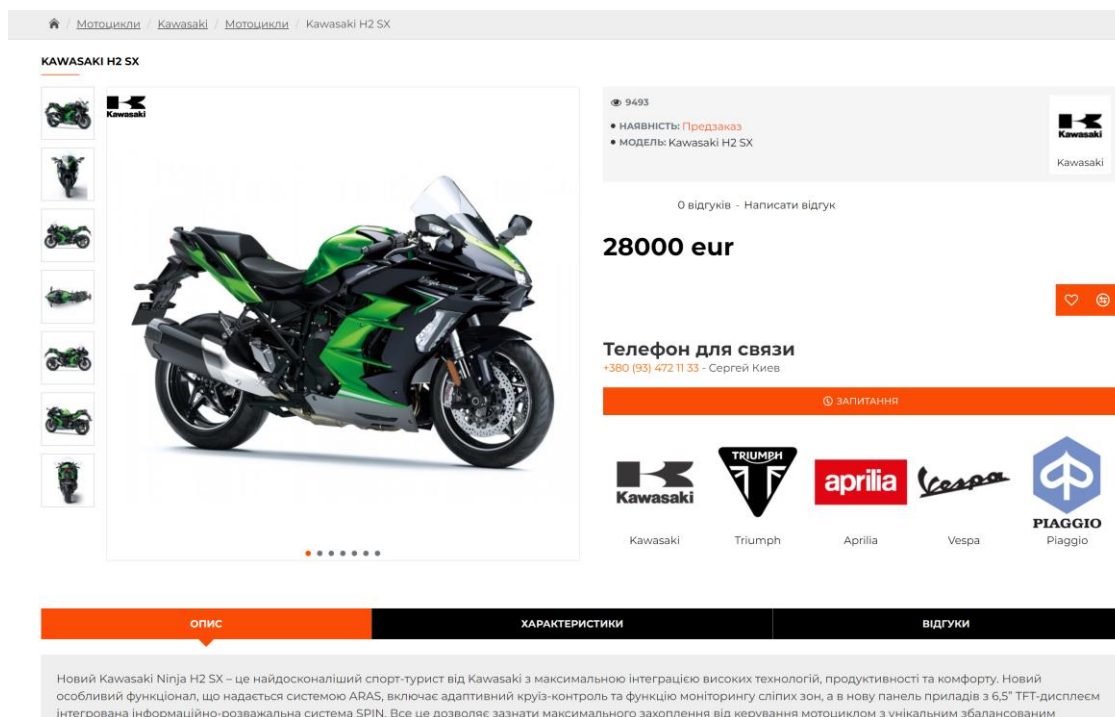


Рисунок 1.6. – Сторінка товару в магазині MOTODOM

Таблиця 1.3

Плюси та мінуси магазину MOTODOM

Плюси	Мінуси
Зручне розположення елементів на сторінках	Замало критеріїв для фільтрації
Приємний дизайн карток товару і загалом сайту	Не зрозуміло де саме на сайті розташована кнопка придбання товару
Зручне пересування сайтом	-

Висновки до першого розділу

В першому розділі було проаналізовано три різних варіанти реалізації інтерфейсу сайтів з продажу мотоциклів та вирішено питання способу реалізації. Було визначено переваги та недоліки схожих програмних продуктів. Визначено основні інструменти для написання програми, її функціонал та варіативність.

		Шумський О.В.			Житомирська політехніка.23.121.14.000 – ПЗ	Арк.
		Фуріхата Д.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		13

Таким чином, при розробці сайту слід урахувати і розробити зручну навігацію по сайту, приємний дизайн, та передбачити можливість підбору користувачем собі товару за визначеними ним критеріями.

		Шумський О.В.			Житомирська політехніка.23.121.14.000 – ПЗ	Арк.
		Фуріхата Д.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		14

РОЗДІЛ 2. ПРОЕКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Проектування загального алгоритму роботи програми

За результатами попередніх досліджень було визначено необхідну структуру та архітектуру програми.

Програма реалізує:

- CRUD функціонал Баз Даних: читання, створення, оновлення, видалення даних
- Різні рівні доступу до зміни даних сайту
- Фільтрування продуктів за незалежними фільтрами
- Можливість авторизації
- Функціонування покупок різних товарів з вказанням кількості штук.

При першому заході на сайт користувач бачить головну сторінку сайту, де розміщений асортимент продуктів. Також не авторизованим користувачам надається можливість додавати продукти в кошик та переглядати товари та їх коментарі, але вони не мають змоги надсилати коментарі та оформлювати замовлення.

Для того щоб мати повні привілеї користувачу потрібно перейти до форми авторизації для того щоб увійти в існуючий аккаунт або зареєструвати новий. Якщо користувач введе невірні дані то буде оповіщення про це. Також є можливість авторизуватись за допомогою Google OAuth 2.0. При вході в аккаунт кошик не очищається і користувач може побачити всі свої продукти в кошику так само, як це було до авторизації.

Для того щоб зайти в адміністративну панель потрібно перейти до форми авторизації і увійти в існуючий аккаунт. Сюди можуть зайти

		Шумський О.В.			Житомирська політехніка.23.121.14.000 – ПЗ	Арк.
		Фуріхата Д.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		15

користувачі з правами доступу “Адміністратор” та “Модератор”. Відмінності прав доступу наведено в таблиці 2.1.

Таблиця 2.1

Відмінності між адміністратором і модератором

	Адміністратор	Модератор
Кількість	1 на всю систему	Безліч
Доступ до редагування товарів	Так	Так
Доступ до редагування та додавання користувачів	Так	Так
Доступ до редагування та додавання модераторів	Так	Ні
Доступ до статистики сайту	Так	Так

2.2 Розробка функціональних алгоритмів роботи програми

Для представлення структури проекту була створена діаграма компонентів. На діаграмі відображуються основні зв'язки між класами та функціональними компонентами. Також на рисунку 2.1 показаний принцип роботи додатку на flux-архітектурі.

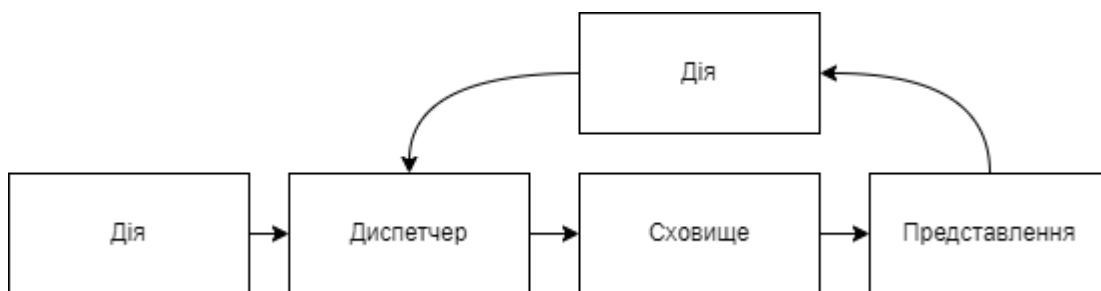


Рисунок 2.1 – Схема роботи flux

Index це початкова точка запуску представлення програми.

Представлення на діаграмі зображено блакитним фоном (рисунки 2.2 - 2.5.). Він запускає компонент App огорнутий в компонент MyProviders, який підключає всі потрібні провайдери для того щоб багато бібліотек мали змогу працювати у всьому проекті глобально. App – це компонент який представляє собою систему роутингу. Кожен шлях представляє собою компонент окремої сторінки та зберігається в папці pages. Вкладені компоненти, які використовуються на одній з сторінок, розташовуються у вигляді ієрархічного дерева всередині компонента сторінки. Для гнучких компонентів, які перевикористовуються у різних сторінках була створена спеціальна папка components.

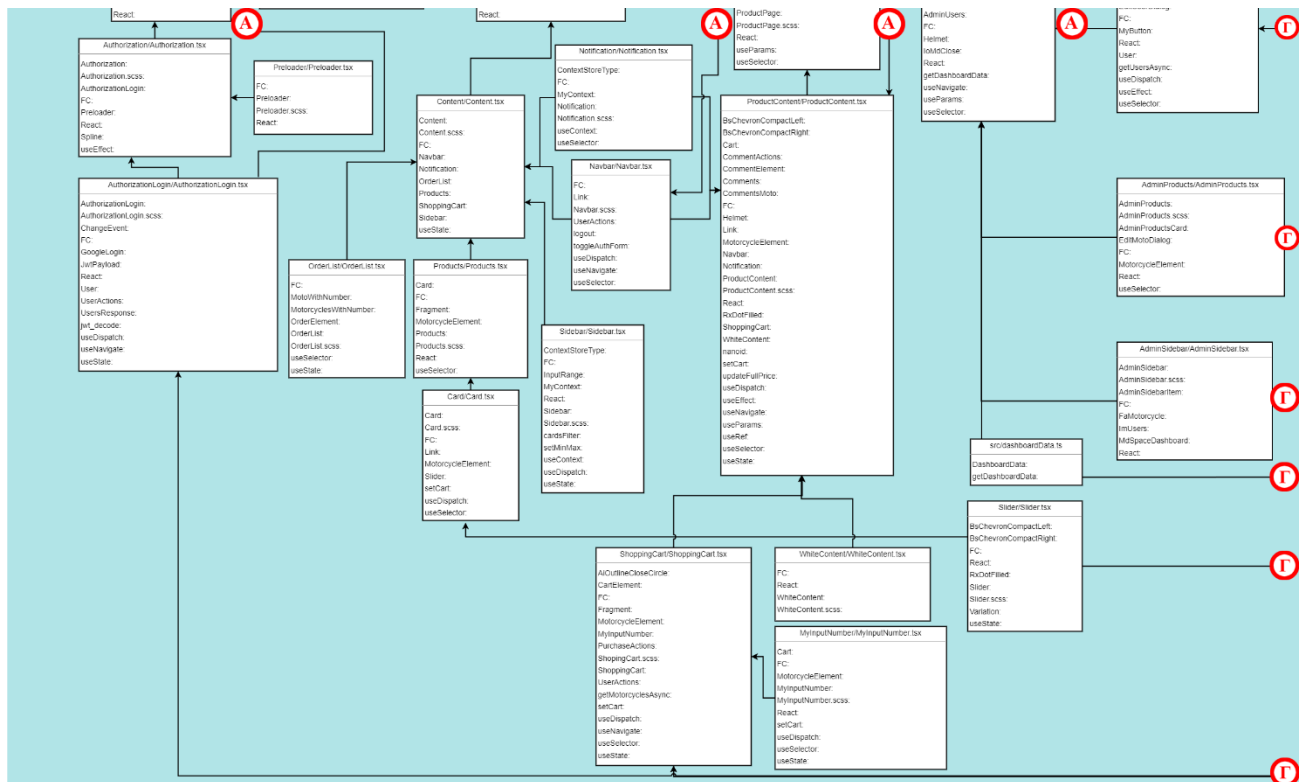


Рисунок 2.2 – Частина “В” клієнтської частини

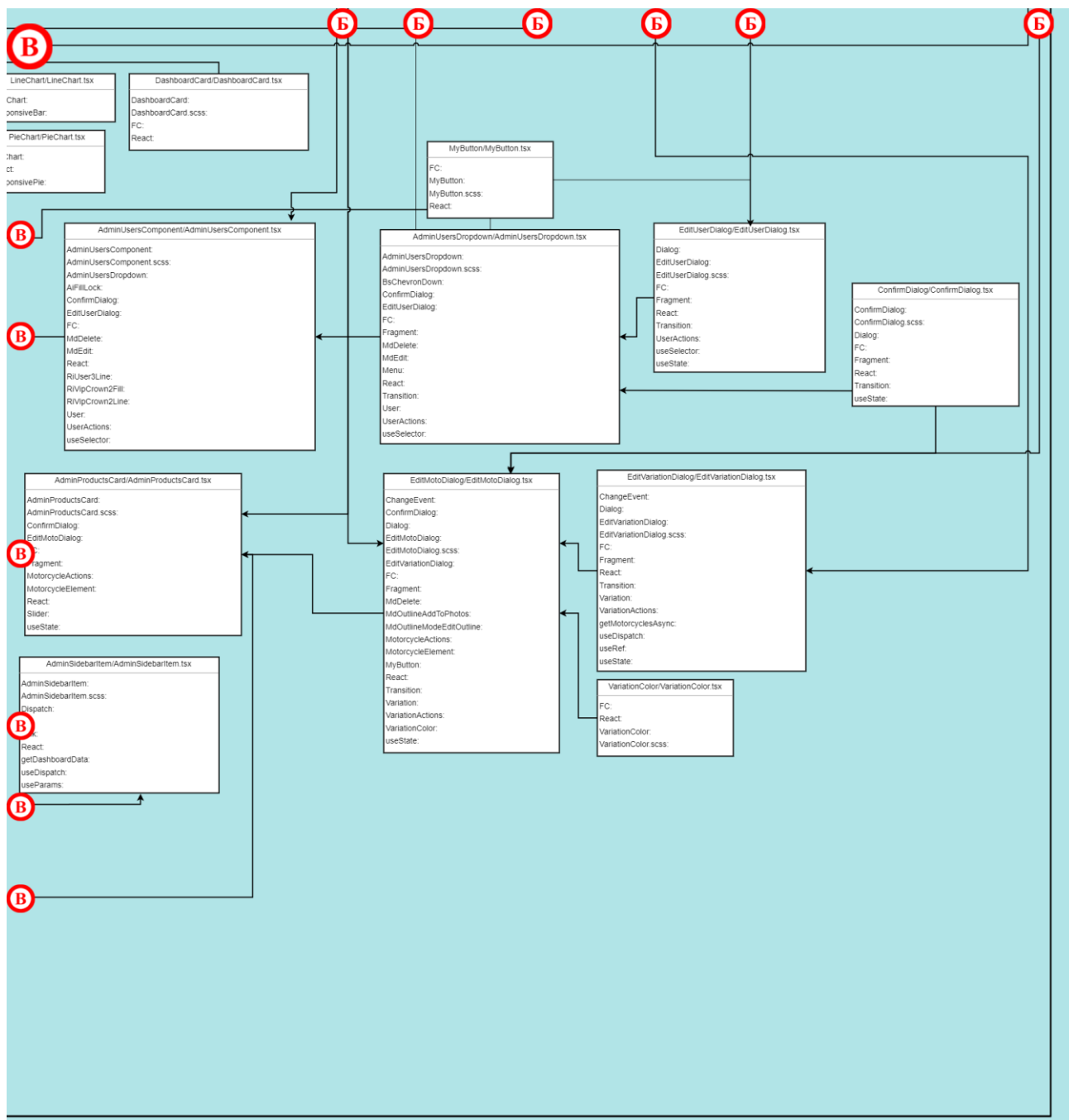


Рисунок 2.3 – Частина “Г” клієнтської частини

Для розташування дій була створена папка actions. На діаграмі вони зазначені червоним кольором (рисунок 2.4). Головна суть дій це обробка вхідних даних так відправка потрібних данхи на сервер а також повертання відповіді з сервера, яка знадобиться в інших компонентах.

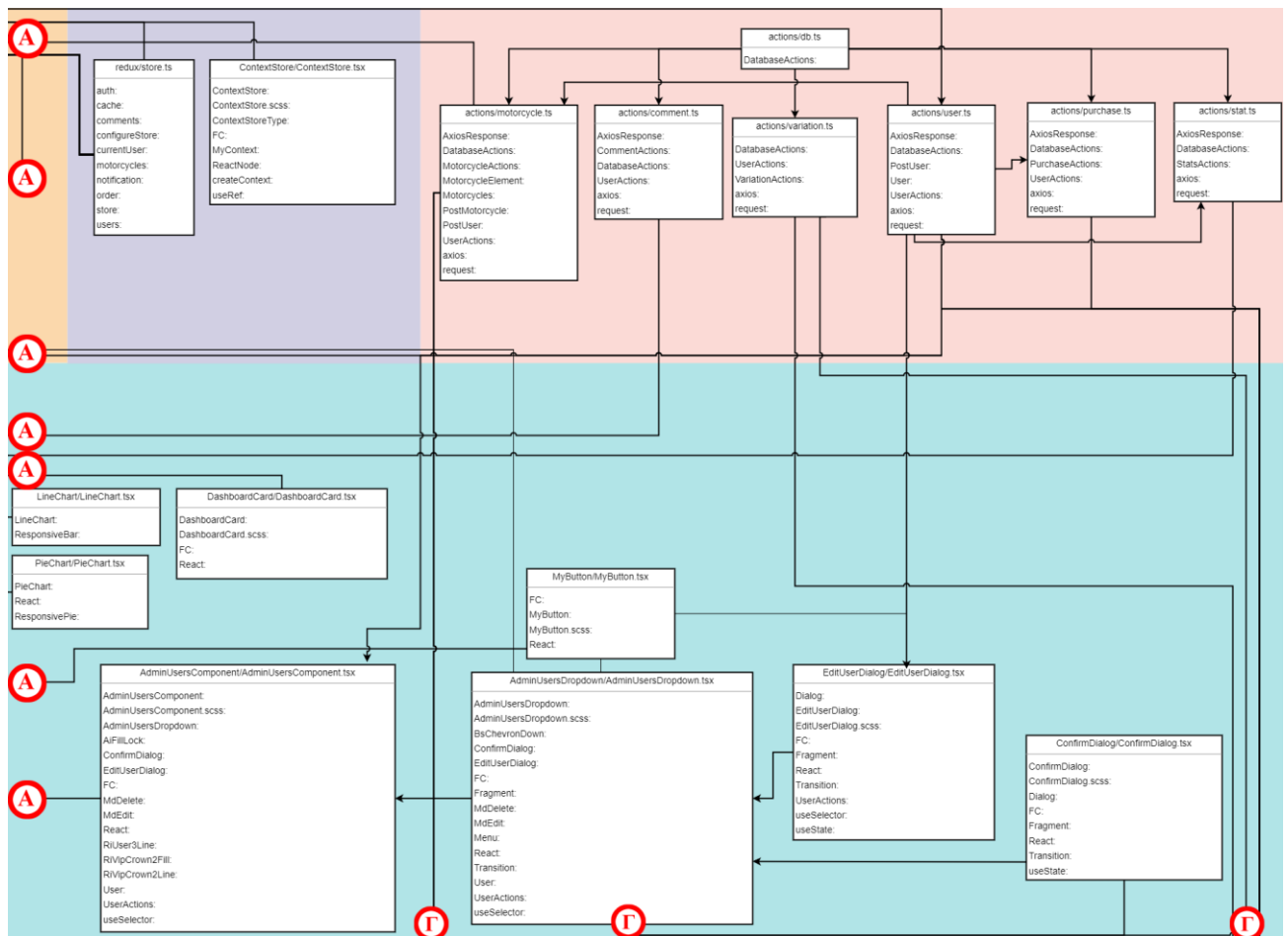


Рисунок 2.4 – Частина “Б” клієнтської частини

Для того щоб зберігати локальні дані додатку було створено одне глобальне сховище в папці redux. На діаграмі воно показано фіолетовим фоном (рисунок 2.4). Головна задача сховища полягає в зберіганні даних та віддаванні їх через підписку на зміну в представленні а також подальше перемальовування представлень при зміні даних в сховищі. Для стейт-менеджменту була використана бібліотека Redux Toolkit.

Диспетчери для додатку були зроблені за допомогою цієї ж бібліотеки. На діаграмі вони показані жовтим кольором (рисунок 2.5). Всі вони розташовані в папці slices а також мають закінчення “Slicer” в назві для того щоб зробити структуру проекту зрозумілішою. Їх робота полягає в тому щоб приймати дані з представлення, читати дані з сховища та змінювати дані в сховищі.

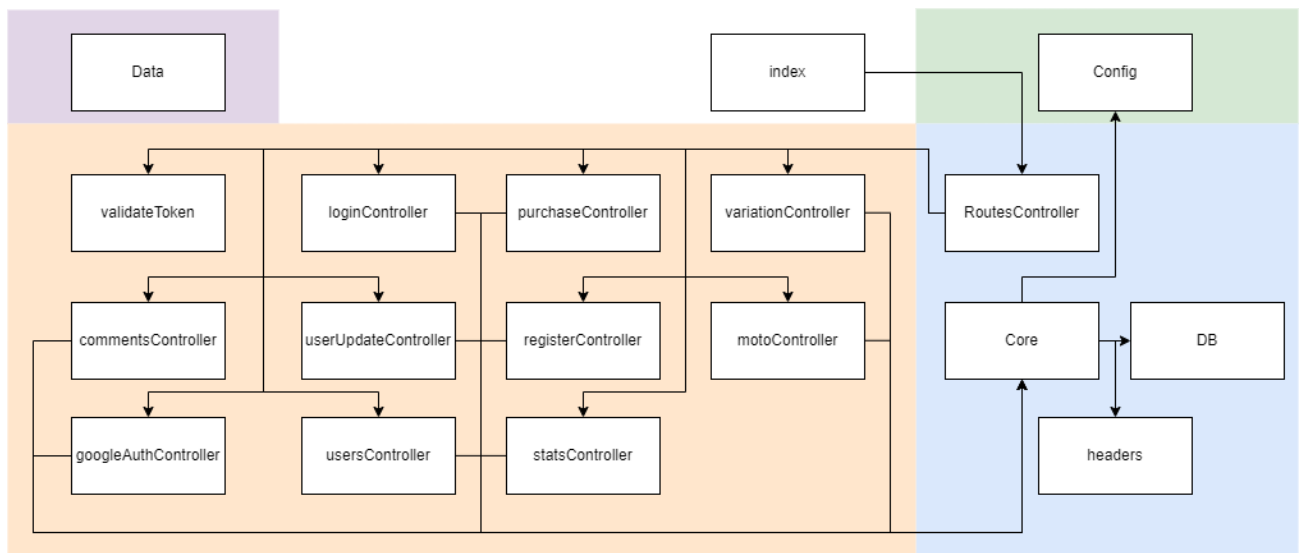


Рисунок 2.6 Схема серверної частини

Аналогічно з клієнтською частиною `index` є початковою точкою запуску серверу. Схема роботи серверної частини додатку показана на рисунку 2.6. Далі виконання коду дістається до класу `RoutesController`, який приймає всі дані і підключає потрібний контроллер для подальшої обробки. В контроллерах запити до бази даних працюють через викликання поля екземпляру класу `DB`, яке знаходиться в класі `Core`. В свою чергу в класі `DB` реалізовані всі методи для роботи з базою даних напрямку.

Також потрібно зазначити що для коректної роботи і уникання помилок зв'язаних з системою CORS клас `Core` використовує заголовки з файлу `headers` а також конфігураційні файли з папки `config`.

База даних складається з 6 потрібних таблиць, які зв'язані між собою (рисунок 2.7).

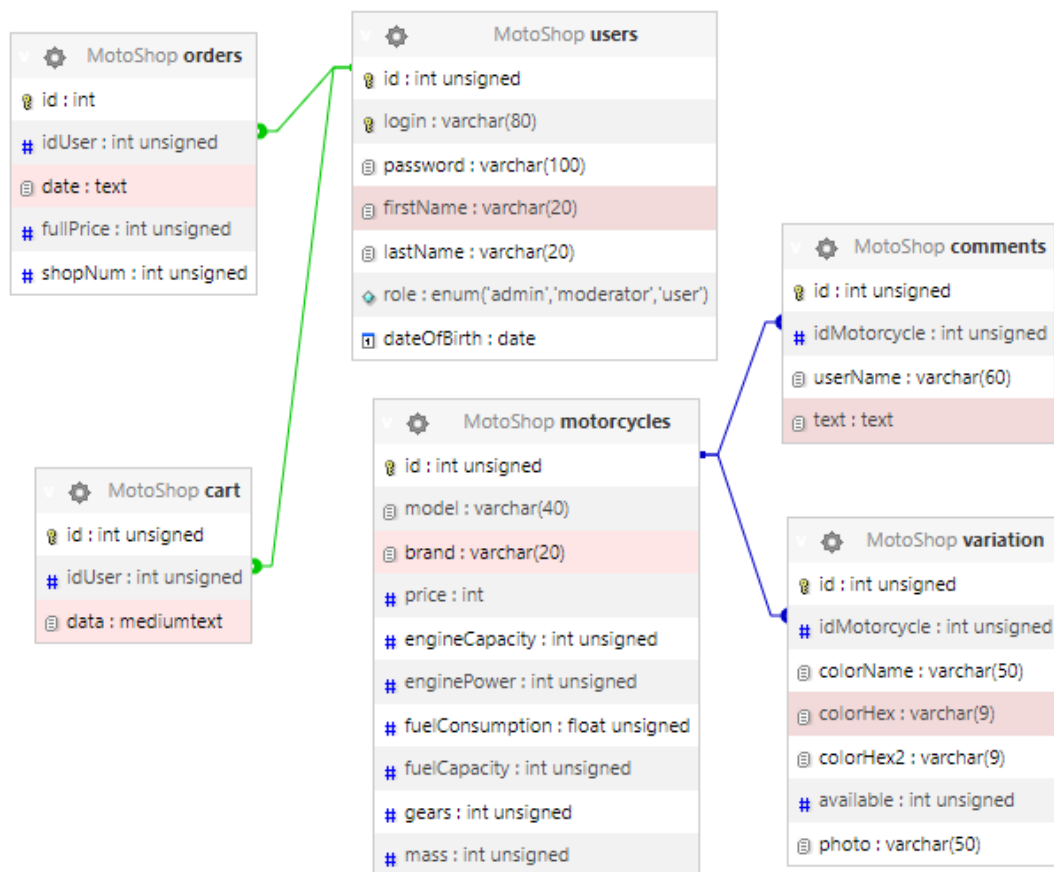


Рисунок 2.7 – Схема бази даних

2.3 Розробка програмного забезпечення:

Після проектування загального алгоритму роботи програми було розпочато її розробку.

Для початку було вирішено розробити основу Flux архітектури для подальшого розвитку та зручного розширення програмного забезпечення. Це стосується структури проекту, його логіки, набору класів та функціональних компонентів. Далі було вирішено розробити сторінку асортименту, сторінку налаштувань користувача, авторизацію та реєстрацію для використання цієї інформації у подальшому.

Всі дані, які надійшли чи надійдуть з клієнтської частини програми будуть перевірятися та потрапляти на сервер, який буде зберігати їх в базі даних у відповідну таблицю та діставати ці дані при запиті клієнта чи сервера.

На початку розробки була розроблена система клієнтського роутингу додатку за допомогою бібліотеки “react-router-dov-v6”. Приклад його роботи представлений далі.

```
export const App = () => {
  const dispatch = useDispatch()
  useEffect(() => {
    new MotorcycleActions().getMotorcycles()
      .then((res: Motorcycles | undefined) =>
        localStorage.setItem(
          'motorcycles',
          JSON.stringify(res ?? []))
      )
    dispatch(getMotorcyclesAsync())
    dispatch(getCurrentUserAsync())
  }, [])

  return (
    <div className="App">
      <Routes>
        <Route
          path="/"
          element={<ContentPage/>}
        />
        <Route
          path='auth'
          element={<AuthPage/>}
        />
        <Route
          path='products/:category/:id'
          element={<ProductPage/>}
        />
        <Route
          path='admin/:category'
          element={<AdminPage/>}
        />
        <Route
          path='*'
          element={<NotFoundPage/>}
        />
      </Routes>
    </div>
  )
}
```

Для того щоб сервер працював в режимі “запит-відповідь” була розроблена система роутингу на стороні серверу. Вся логіка роутингу полягає в розбиванні запиту на категорії і підключення потрібного контроллера для подальшої обробки даних.

```
<?php
namespace core;
require 'Core/Core.php';
class RoutesController
{
    public static function execute(): void
```

		Шумський О.В.			Житомирська політехніка.23.121.14.000 – ПЗ	Арк.
		Фуріхата Д.В.				23
Змн.	Арк.	№ докум.	Підпис	Дата		

```

{
    $route = explode(separator: '/', string: $_SERVER['REQUEST_URI']);
    array_shift($route);
    $category = $route[0];
    $categoryClean = explode('?', $category)[0];
    $method = $_SERVER['REQUEST_METHOD'];
    require "api/{$categoryClean}/{$categoryClean}Controller.php";
}
}

```

Для повноцінної роботи був розроблений клас Core, який має в собі метод і поле для роботи з базою даних.

```

<?php
namespace core;

include 'DB.php';
include 'headers.php';

class Core
{
    private static ?Core $instance = null;
    public static DB $db;
    public static function getInstance(): Core
    {
        if (empty(self::$instance)) {
            self::$instance = new self();
            self::$db = self::getDB();
        }
        return self::$instance;
    }
    public static function getDB(): DB
    {
        require 'config/config.php';
        return new DB(
            hostname: DATABASE_HOST,
            login: DATABASE_LOGIN,
            password: DATABASE_PASSWORD,
            database: DATABASE_BASENAME
        );
    }
}

```

Для переходу до сторінки мотоциклу був створений компонент ProductPage. Він містить в собі логіку пошуку даних та промальовування контенту сторінки якщо все нормально та перенаправлення на сторінку помилки якщо користувач зайшов за некоректним посиланням.

```

import React, {FC} from 'react'
import './ProductPage.scss'
import {useParams} from "react-router-dom"
import {useSelector} from 'react-redux'
import {MotorcycleElement} from "../../Types";
import {NotFoundPage} from "../NotFoundPage/NotFoundPage";
import {ProductContent} from "../ProductContent/ProductContent";

export interface ProductPageProps {
}

```

		Шумський О.В.			Житомирська політехніка.23.121.14.000 – ПЗ	Арк.
		Фуріхата Д.В.				24
Змн.	Арк.	№ докум.	Підпис	Дата		


```

export const ProductPage: FC<ProductPageProps> = ({}) => {
  const {motorcycles} = useSelector((state) => state.motorcycles)
  let pageId = useParams<{ id: string }>().id;
  if (!pageId)
    pageId = '-1'
  let currentMoto = motorcycles.find(({id}: MotorcycleElement) => id ===
parseInt(pageId as string))
  return (
    <>
      {
        (currentMoto?.variation?.length > 0) ?
          <ProductContent
            currentMoto={currentMoto}
            pageId={pageId}
            motorcycles={motorcycles}
          />
          :
          <NotFoundPage/>
      }
    </>
  )
}

```

Для реалізації карток продуктів був створений компонент Card. Основна його задача це приймання вхідних даних і промальовування інтерфейсу картки.

```

import {FC} from 'react'
import './Card.scss'
import {Link} from 'react-router-dom'
import {Slider} from "../../components/Slider/Slider";
import {MotorcycleElement} from "../../Types";

export interface CardProps extends MotorcycleElement {
  cartIcon: string
}

export const Card: FC<CardProps> = (
  {
    id,
    model,
    brand,
    price,
    fuelCapacity,
    fuelConsumption,
    gears,
    mass,
    variation,
    cartIcon
  }) => {

  return (
    <div className="wrap Card">
      <div className="card">
        <div
          className="img"
        >
          <Slider
            variation={variation}
            color={'#ffffff'}
            sliderStyle={'-mt-6'}

```

		Шумський О.В.			Житомирська політехніка.23.121.14.000 – ПЗ	Арк.
		Фуріхата Д.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		25

```

        />
      </div>
      <div className="card-body">
        <div className="flex justify-between">
          <h2 className="card-title">{brand}</h2>
          <div className="toCartWrapper relative">
            </div>
          </div>
          <div className="flex justify-between">
            <h4 className="truncate">{model}</h4>
            <h4>{price} грн</h4>
          </div>

          <div className="card-actions justify-end">
            <Link to={`products/motorcycles/${id}`}>
              <button
                className="toProduct"
              >
                Перейти до товару
              </button>
            </Link>
          </div>
        </div>
      </div>
    </div>
  )
}

```

Висновки до другого розділу

В цьому розділі було розглянуто структуру програми у вигляді діаграм компонентів, бази даних, її основні можливості, які необхідні для коректної роботи даної програми. Було розроблене програмне забезпечення та наведено лістинг деяких його найголовніших методів.

		Шумський О.В.			Житомирська політехніка.23.121.14.000 – ПЗ	Арк.
		Фуріхата Д.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		26

РОЗДІЛ 3. ОПИС РОБОТИ З ПРОГРАМНИМ ДОДАТКОМ ТА ЙОГО ТЕСТУВАННЯ

3.1 Опис роботи з програмним додатком

При першому заході на сайт користувач бачить головну сторінку сайту, де розміщений асортимент продуктів (рисунк 3.1). Є бокова панель завдяки якій можна відфільткувати товар за необхідними критеріями. Робота фільтрації зображена на рисунку 3.2.

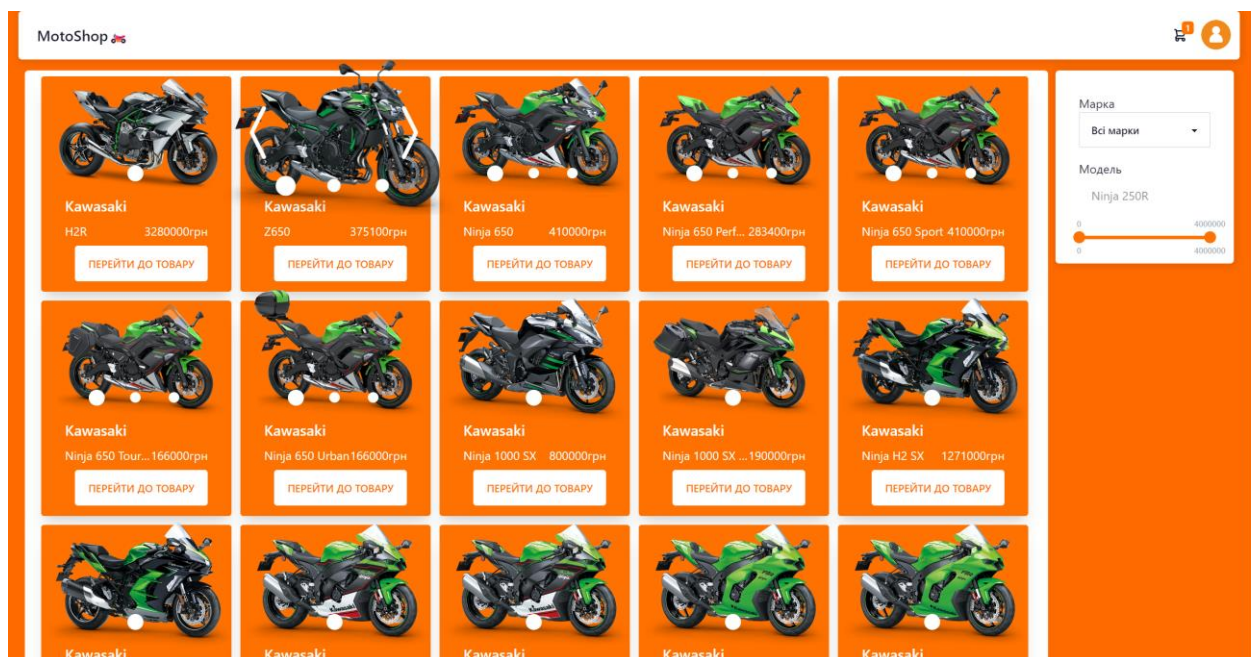


Рис. 3.1. Вигляд головної сторінки сайту

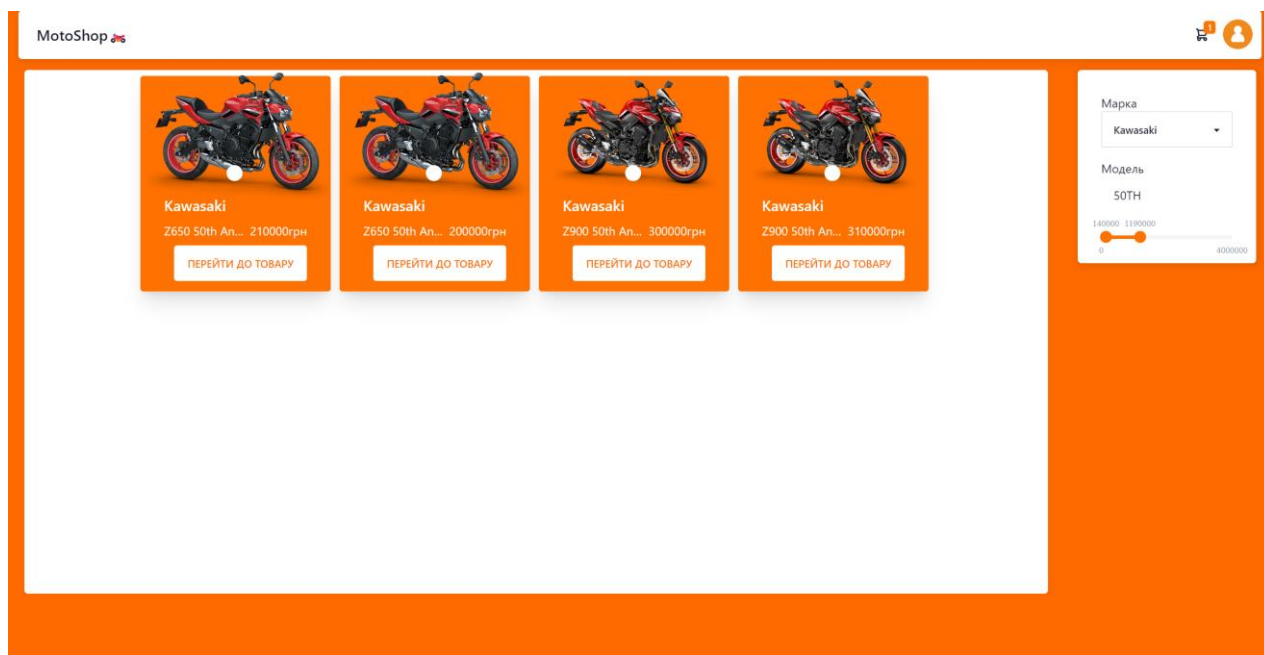


Рис. 3.2. Робота фільтрації

Користувач може додати товар до кошику навіть якщо він не авторизований. При додаванні товару до кошику іконка в навігаційній панелі змінює свій зовнішній вигляд і відображається кількість товару у кошику. При наведенні на іконку можна побачити товар на яку суму зараз міститься у кошику (рисунок 3.3). Зайшовши в кошик можна відредагувати кількість замовляемого товару, видалити товар з кошику та замовити товар. При замовленні товару користувач обирає за якою адресою він забиратиме замовлення. Зовнішній вигляд кошику зображено на рисунку 3.4.

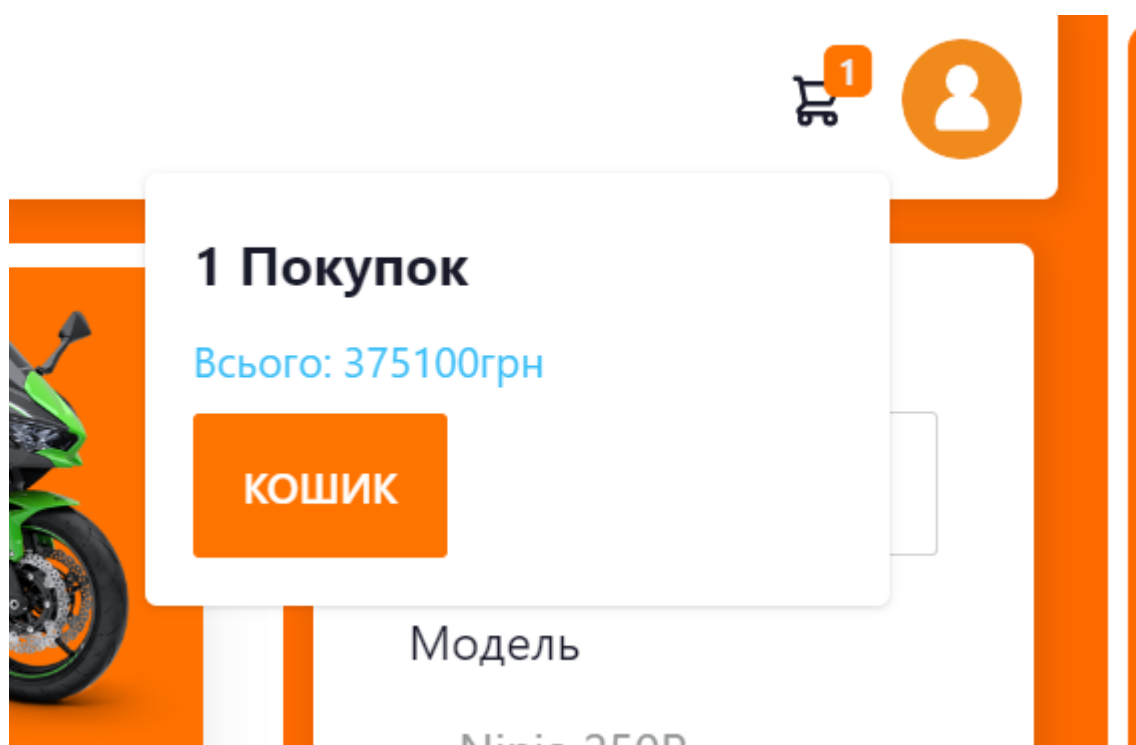


Рис. 3.3. Перегляд суми товарів у кошику

		Шумський О.В.			Житомирська політехніка.23.121.14.000 – ПЗ	Арк.
		Фуріхата Д.В.				28
Змн.	Арк.	№ докум.	Підпис	Дата		

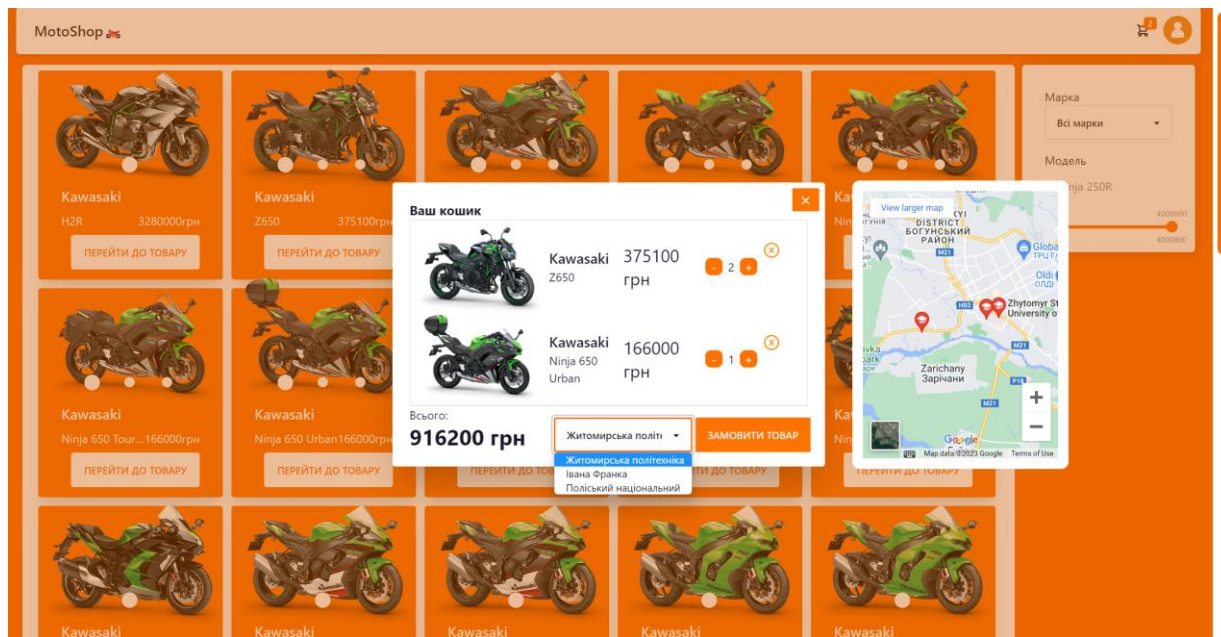


Рис. 3.4. Зовнішній вигляд кошику

Перейшовши на сторінку товару користувач може побачити більше інформації про товар, а також замовити його (рисунок 3.5). При натисканні на стрілки варіація мотоциклу змінюється. Кожна варіація відображається в замовленні окремим товаром, має свою кількість доступних одиниць на складі. Внизу сторінки є коментарі де користувач може залишити свій відгук (рисунки 3.6 – 3.7). Для неавторизованих користувачів додавання коментарів є забороненим.

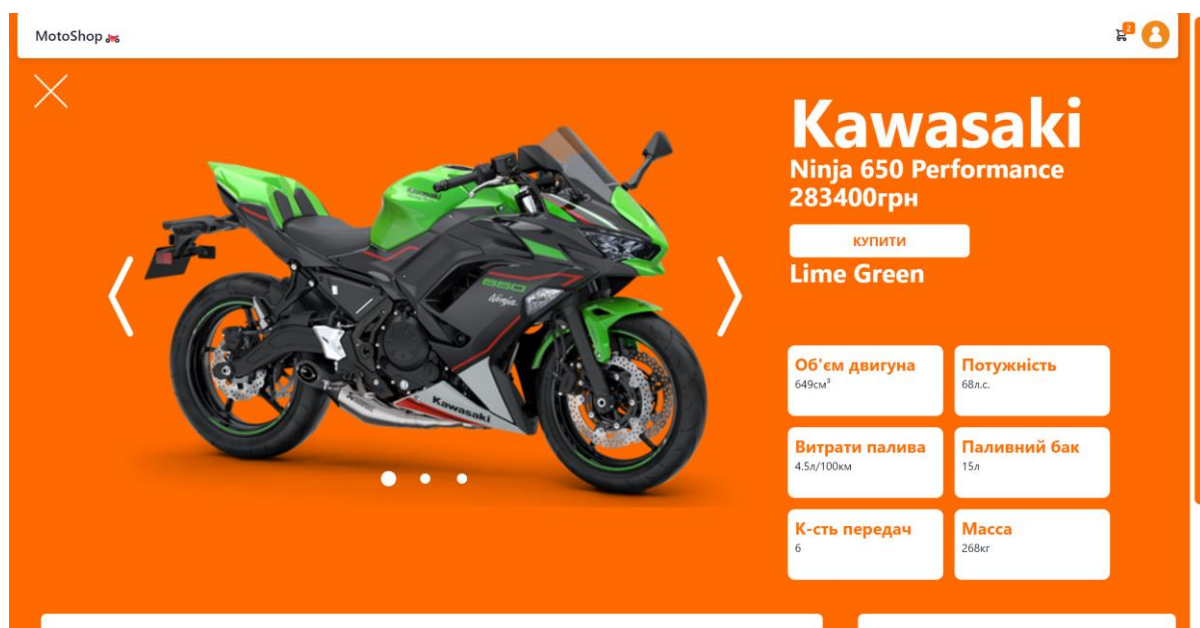


Рис. 3.5. Сторінка товару

		Шумський О.В.			Житомирська політехніка.23.121.14.000 – ПЗ	Арк.
		Фуріхата Д.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		29

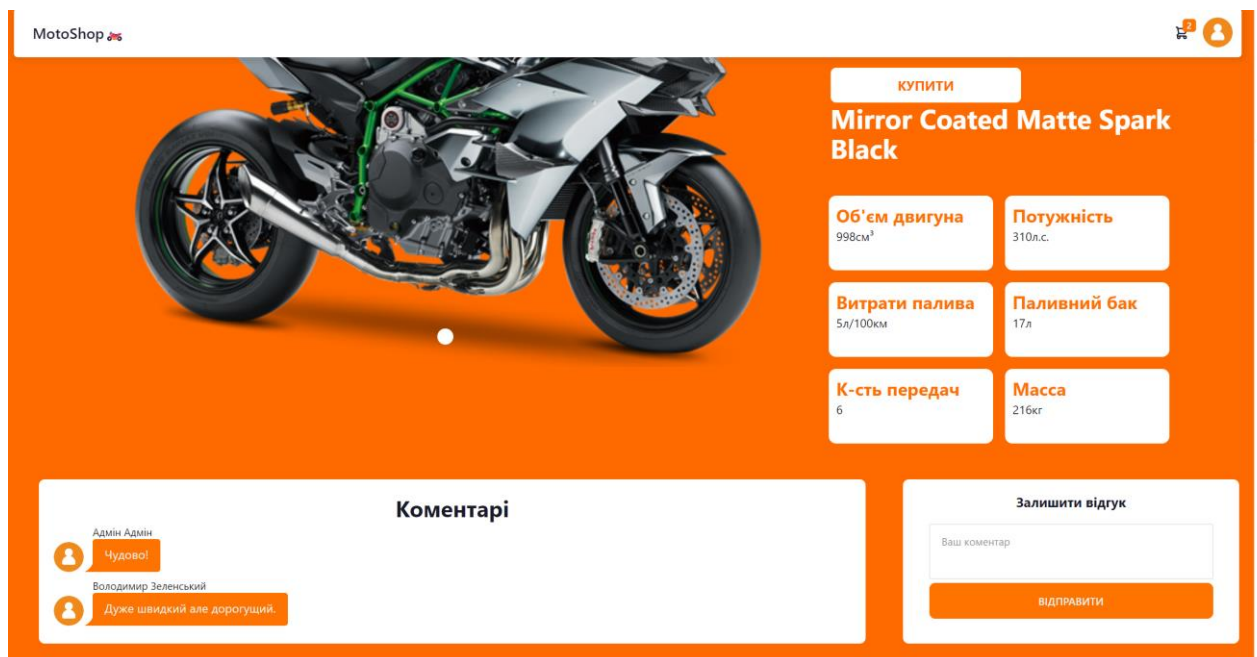


Рис. 3.6. Коментарі на сторінці товару

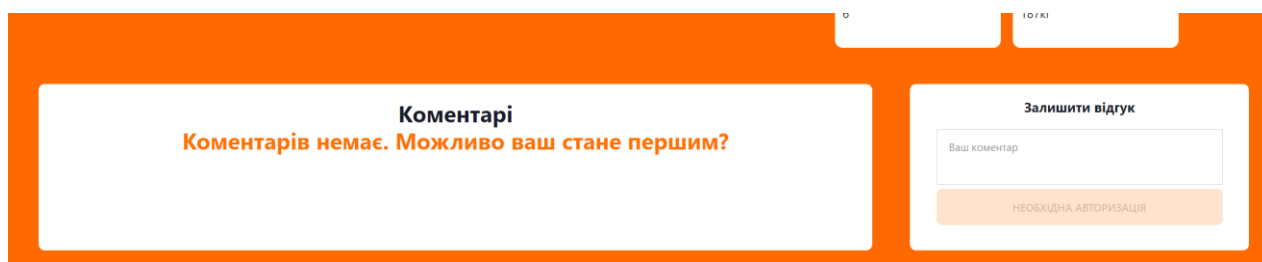


Рис. 3.7. Коментарі на сторінці товару

Увійти чи зареєструватися можна з усіх сторінок сайту. Авторизація знімає обмежування користувача на сайті (рисунок 3.8). Також користувач має можливість увійти на сайт через гугл акаунт (рисунок 3.9).

		Шумський О.В.			Житомирська політехніка.23.121.14.000 – ПЗ	Арк.
		Фуріхата Д.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		30

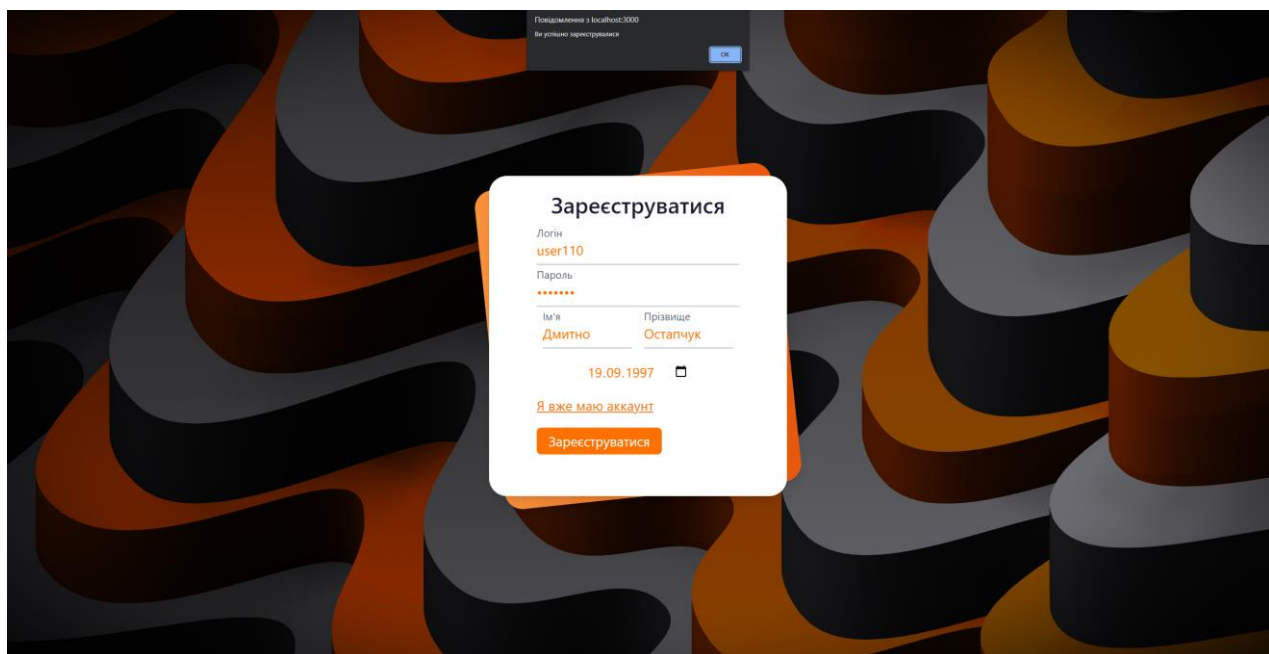


Рис. 3.8. Сторінка реєстрації користувача

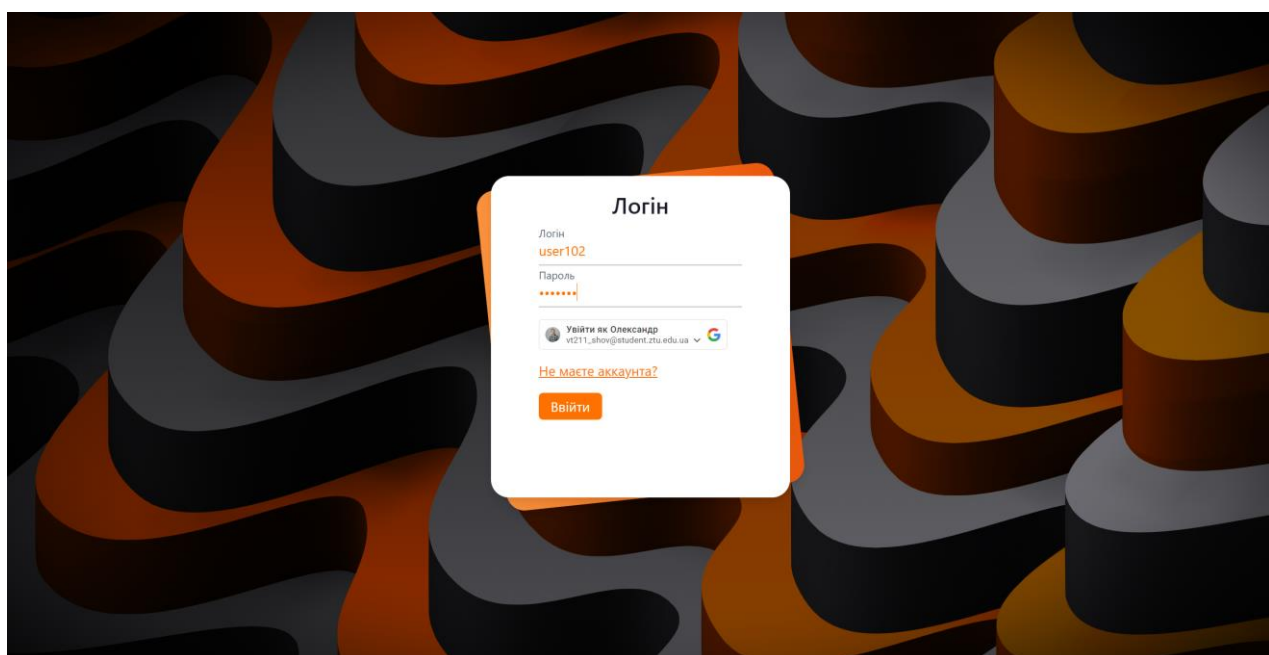


Рис. 3.9. Сторінка авторизації користувача

Користувачі що мають рівень доступу модератора можуть додавати, змінювати та видаляти товар з бази даних. Цю роль неможливо отримати ніяким іншим способом, окрім як, за допомогою адміністратора. При додаванні мотоцикла (рисунок 3.10) можна побачити тільки основні параметри товару. Після збереження він буде доданий в список, проте буде відображатись знак відсутності фото а також він не буде наявний в асортименті користувачів (рисунок 3.15). Після цього його знову можна буде знайти і відкрити меню з

		Шумський О.В.			Житомирська політехніка.23.121.14.000 – ПЗ	Арк.
		Фуріхата Д.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		31

роботою над варіаціями (рисунок 3.13). Варіації в кожного мотоцикла представлені вибором кількості одиниць на складі, індивідуального кольору а також фото (рисунок 3.11).

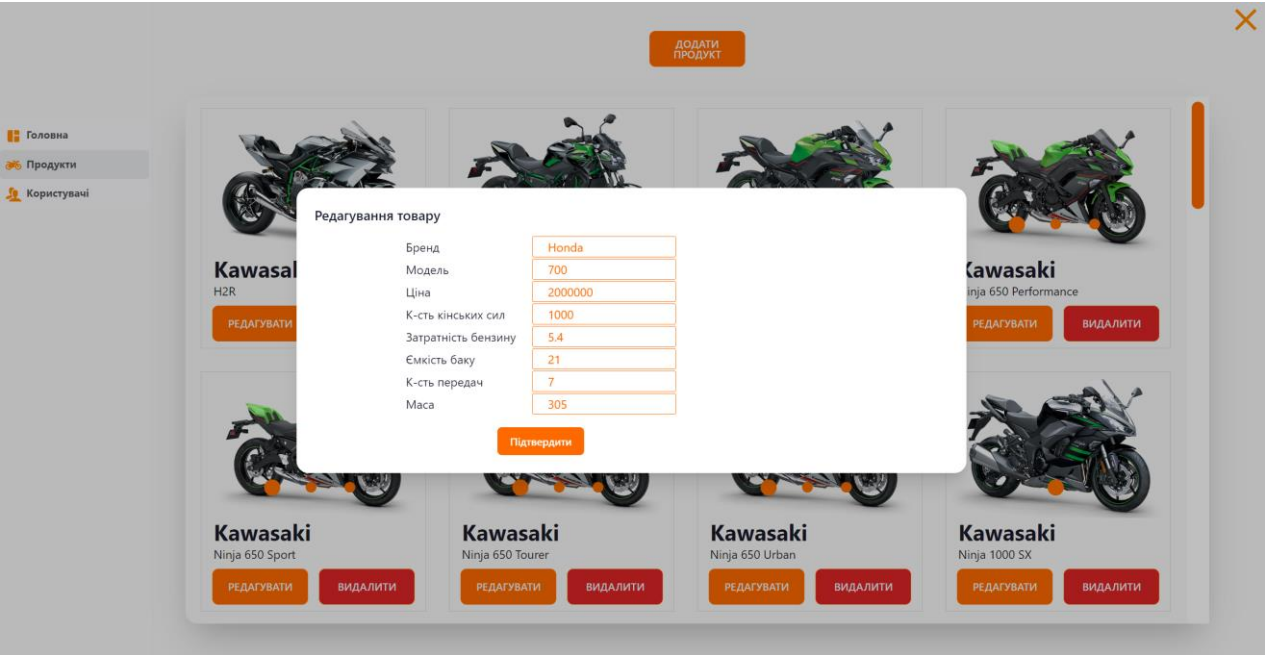


Рис. 3.10. Сторінка редагування товарів

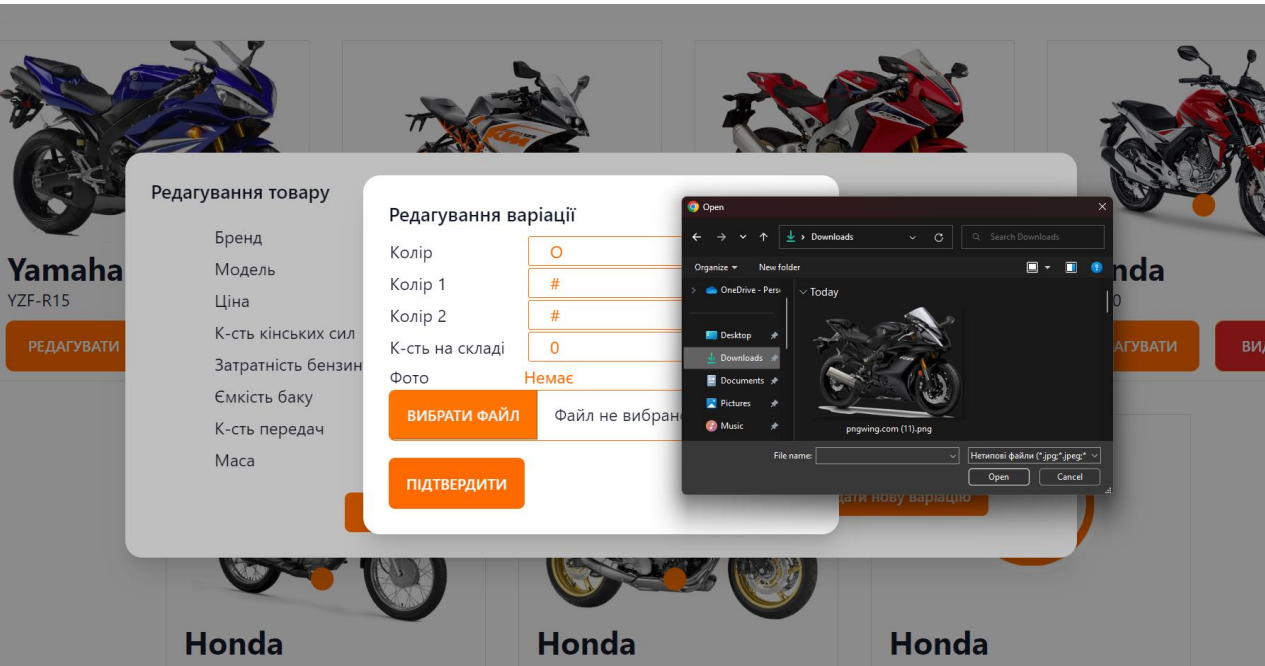


Рис. 3.11. Сторінка редагування товарів

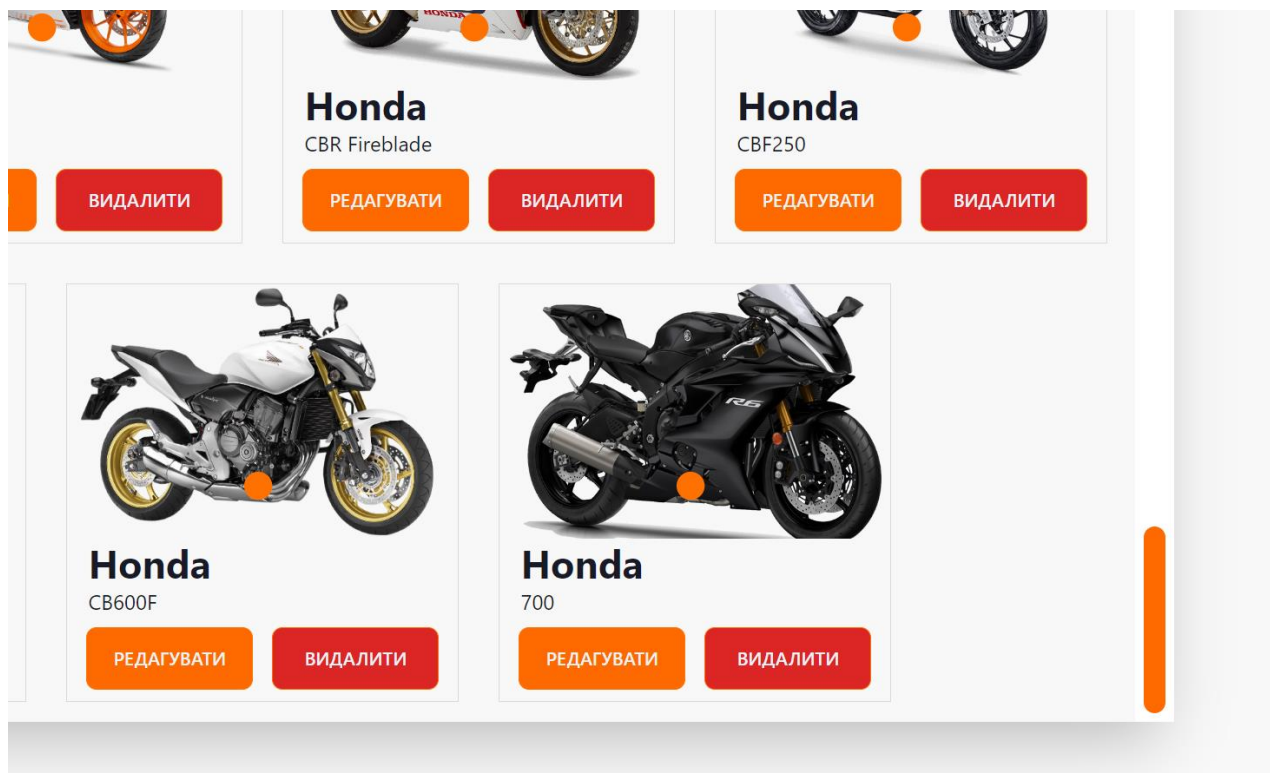


Рис. 3.12. Редагування товару

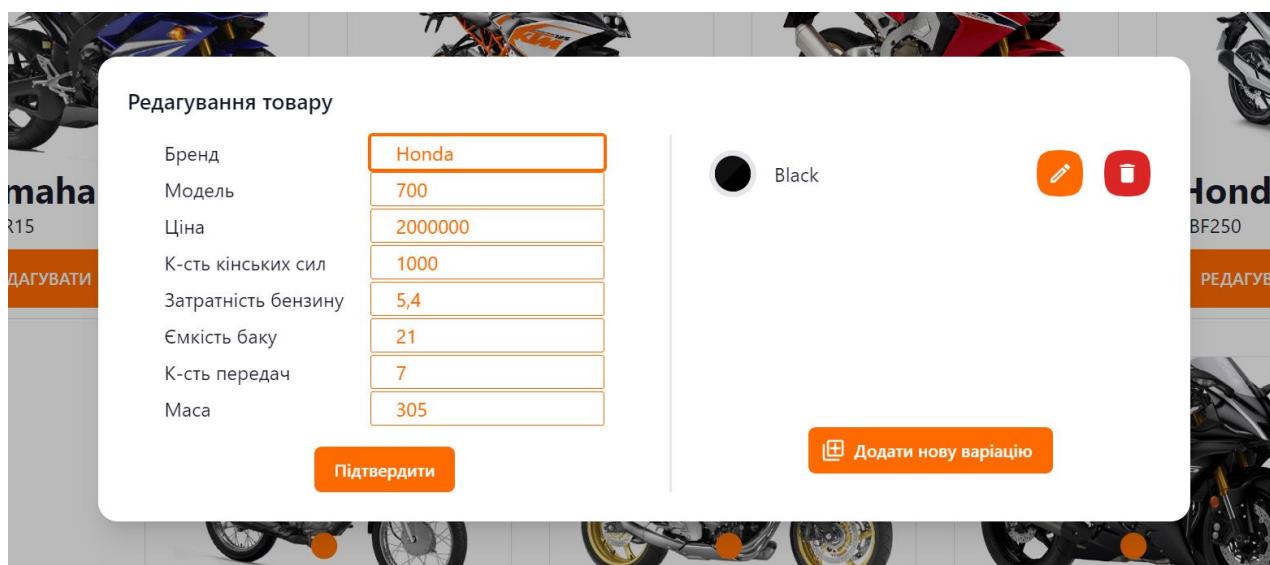


Рис. 3.13. Редагування товару

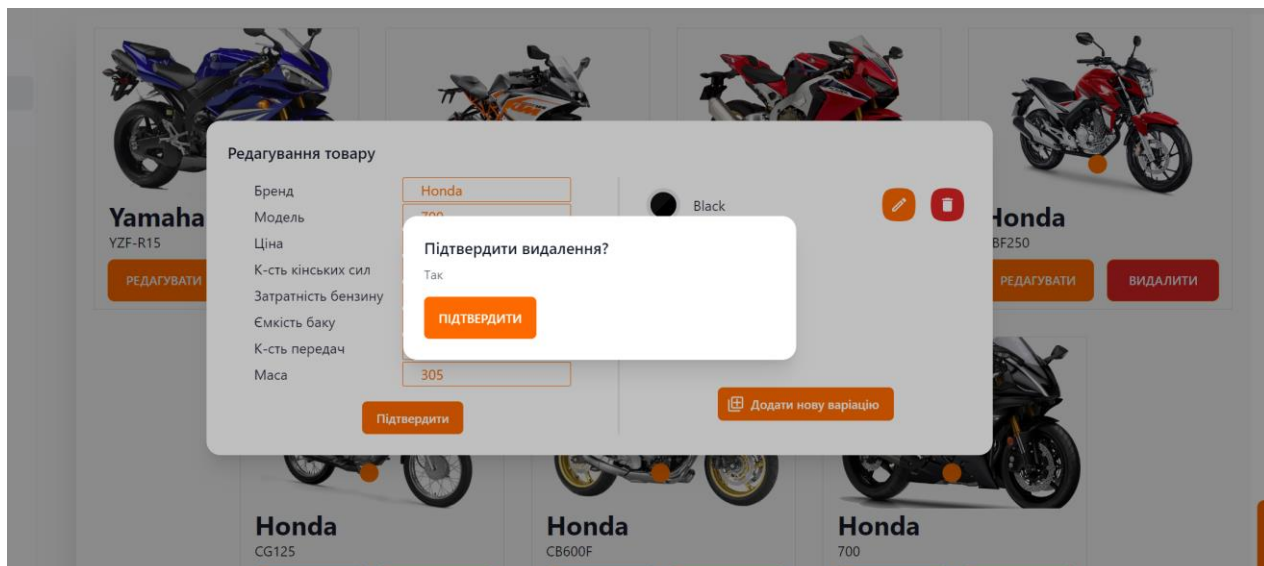


Рис. 3.14. Видалення варіації мотоциклу

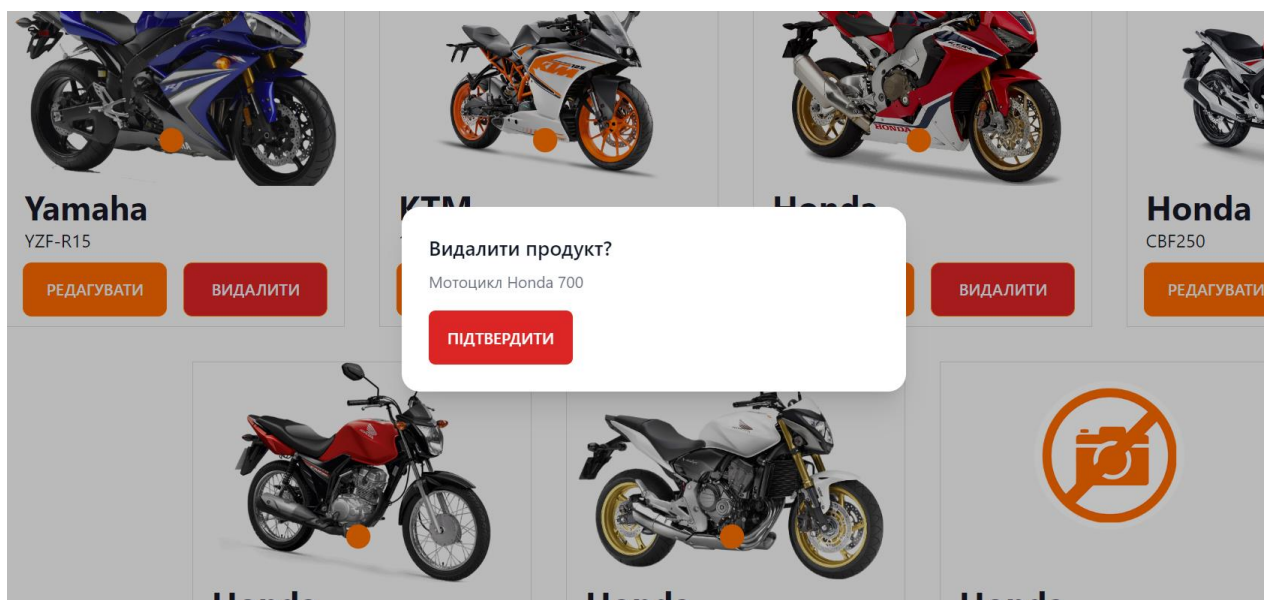


Рис. 3.15. Видалення товару

Для адміністратора та модераторів присутня сторінка для редагування користувачів (рисунок 3.16) з можливістю додати нового користувача (рисунок 3.18) та відредагувати або видалити вже існуючого (рисунок 3.17). Також в адміністратора є можливість назначити нову роль іншим обліковим записам (рисунки 3.18 – 3.19.). Слід відзначити що модератор має право редагувати тільки облікові записи зі статусом користувача (рисунок 3.20).

		Шумський О.В.			Житомирська політехніка.23.121.14.000 – ПЗ	Арк.
		Фуріхата Д.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		34

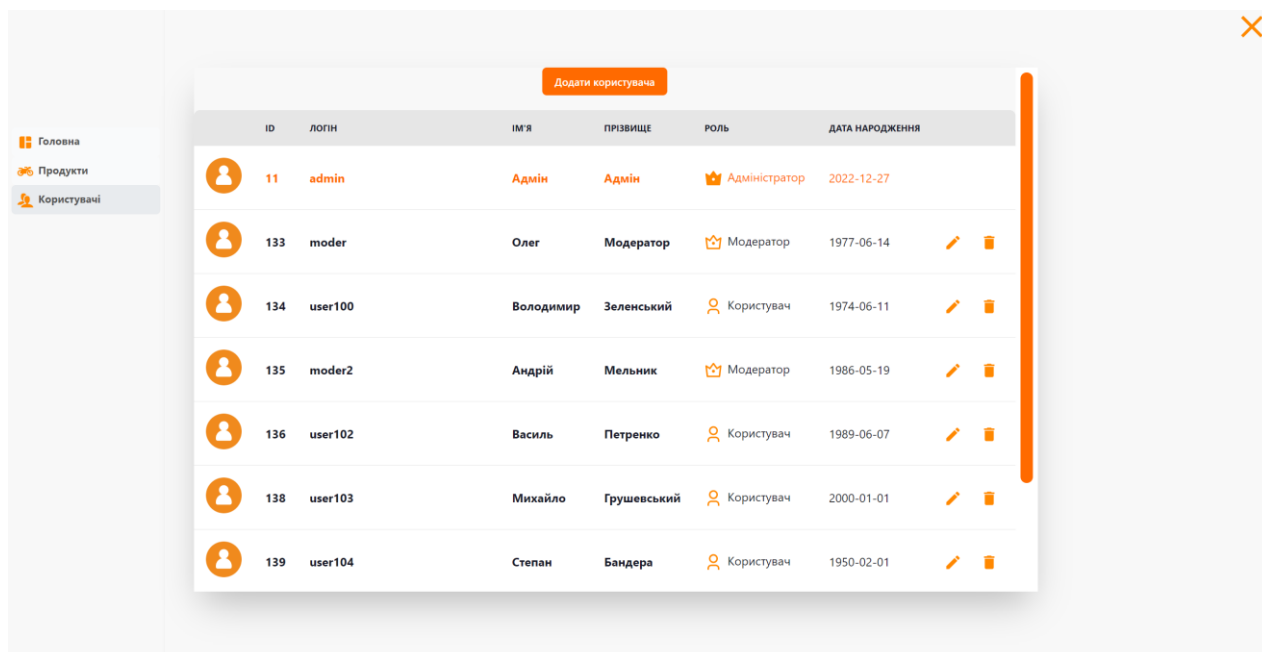


Рис. 3.16. Сторінка редагування користувачів

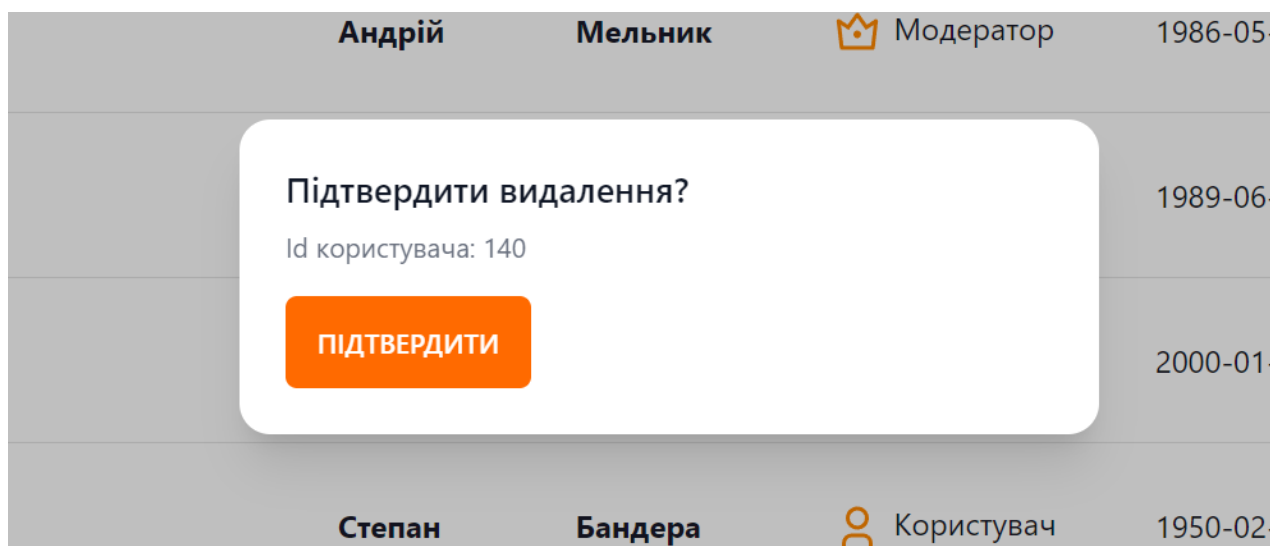


Рис. 3.17. Видалення користувача

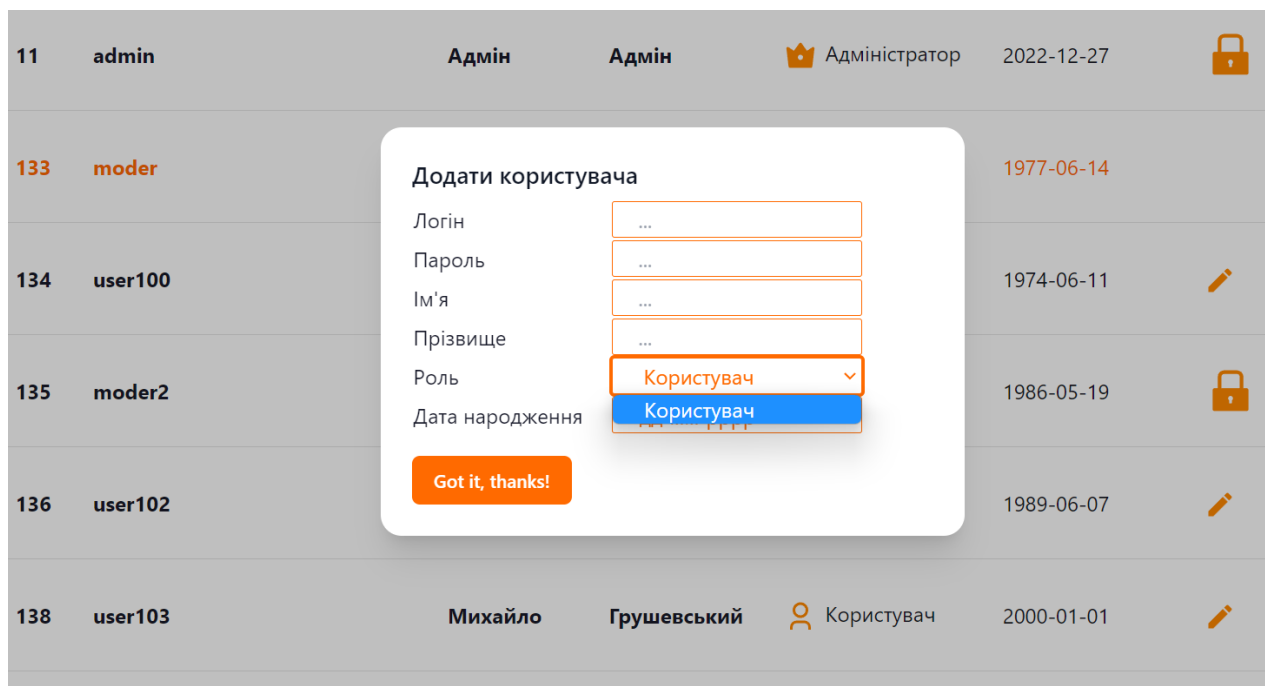


Рис. 3.18. Додавання користувача

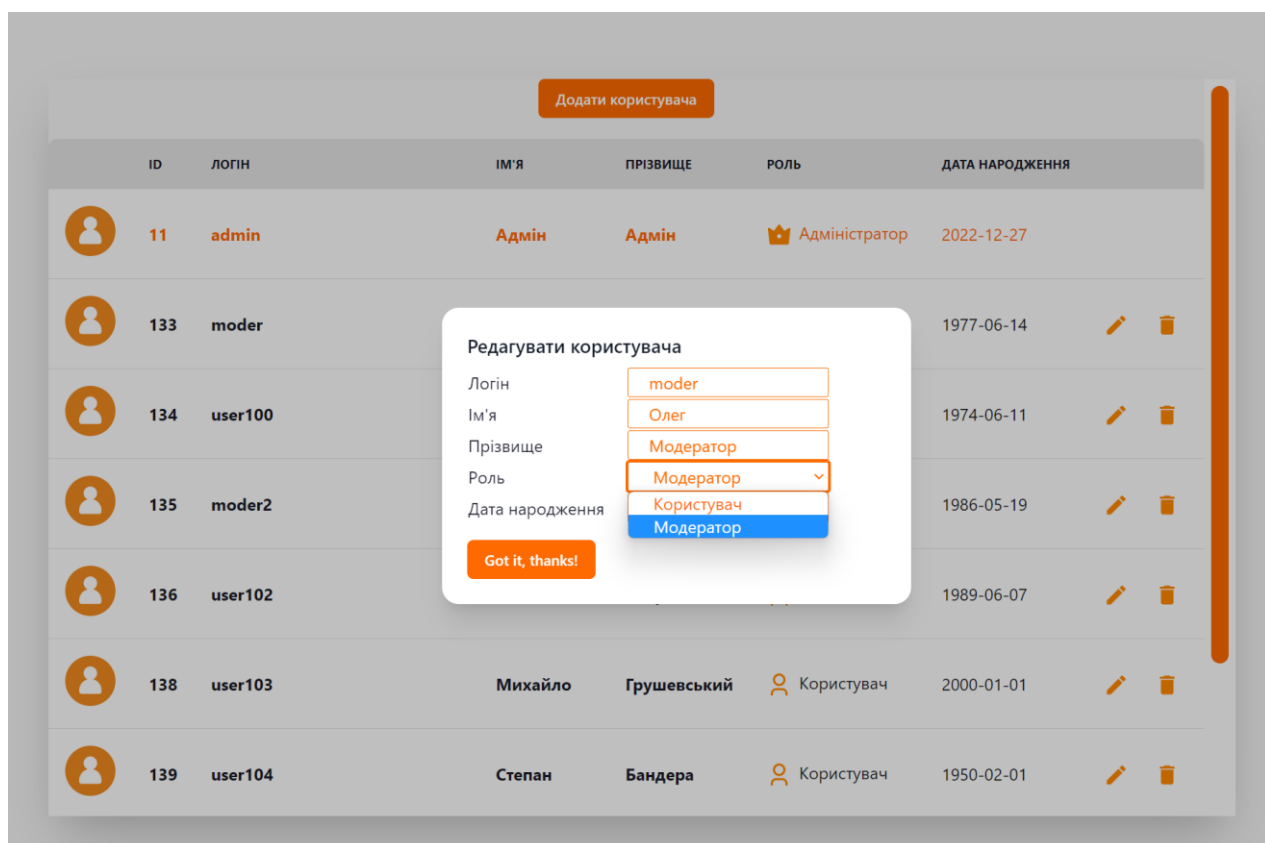


Рис. 3.19. Назначення ролі

ID	ЛОГІН	ІМ'Я	ПРИЗВИЩЕ	РОЛЬ	ДАТА НАРОДЖЕННЯ
11	admin	Адмін	Адмін	Адміністратор	2022-12-27
133	moder	Олег	Модератор	Модератор	1977-06-14
134	user100	Володимир	Зеленський	Користувач	1974-06-11
135	moder2	Андрій	Мельник	Модератор	1986-05-19
136	user102	Василь	Петренко	Користувач	1989-06-07
138	user103	Михайло	Грушевський	Користувач	2000-01-01
139	user104	Степан	Бандера	Користувач	1950-02-01

Рис. 3.20. Назначення ролі

Окрім того в адміністратора та модераторів є можливість переглядати статистику сайту на окремій для цього сторінці (рисунок 3.21.).

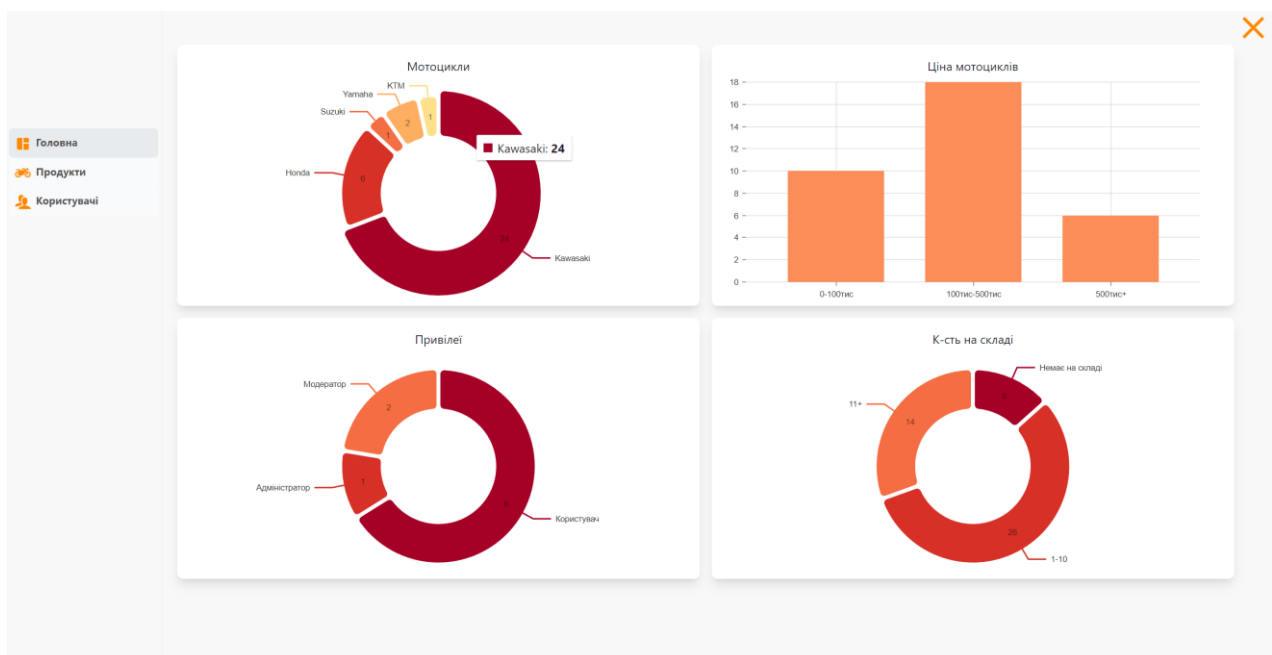


Рис. 3.21. Статистика сайту

3.2 Тестування роботи програмного забезпечення

При тестуванні програмного забезпечення було виявлено та усунено декілька проблем. Серед них:

- Некоректне відображення повної суми товару в кошику.
- Проблеми з авторизацією.

		Шумський О.В.			Житомирська політехніка.23.121.14.000 – ПЗ	Арк.
		Фуріхата Д.В.				37
Змн.	Арк.	№ докум.	Підпис	Дата		

- Неможливість додати користувача через панель адміністратора.
- Проблема з оформленням пустого замовлення.
- Проблема замовлення продуктів, які не наявні в базі даних.
- Некоректна робота фільтру.

Висновки до третього розділу

В цьому розділі було розглянуто роботу програми, зроблено повне її тестування, перевірено коректність взаємодії усіх введених даних та розглянуто основні можливості ПЗ, ознайомлено з основними виключеннями та усунено всі знайдені проблеми які заважали коректній роботі сайту.

		Шумський О.В.			Житомирська політехніка.23.121.14.000 – ПЗ	Арк.
		Фуріхата Д.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		38

ВИСНОВКИ

В першому розділі було проаналізовано поставлену задачу саме які дії потрібно виконати для повної розробки програми, а також конкурентів зі схожим функціоналом та інтерфейсом. Стало зрозуміло основні потреби при розробці програми та поставлено конкретні цілі на проект.

В другому розділі було описано логічну роботу програми, її потреби та проілюстровано за допомогою блок-схеми, діаграм класів та діаграми відносин БД. Цей етап роботи та розробки програмного забезпечення є одним з найважливіших, оскільки саме від проектування та логічної реалізації всіх функцій програмного додатку залежить надійність його роботи.

В третьому розділі було описано повну роботу готової програми, зроблено повне тестування та виправлено всі її недоліки. Підсумовуючи вище сказане, можемо сказати, що ця робота дала нам змогу закріпити раніше вивчений матеріал, написавши для цього реальний веб-сайт.

Під час виконання поставленого завдання на курсову роботу були отримані практичні навички з написання власної CMS системи за допомогою веб-технологій на мові програмування PHP і продемонстровано ці навички при написанні реального сайту.

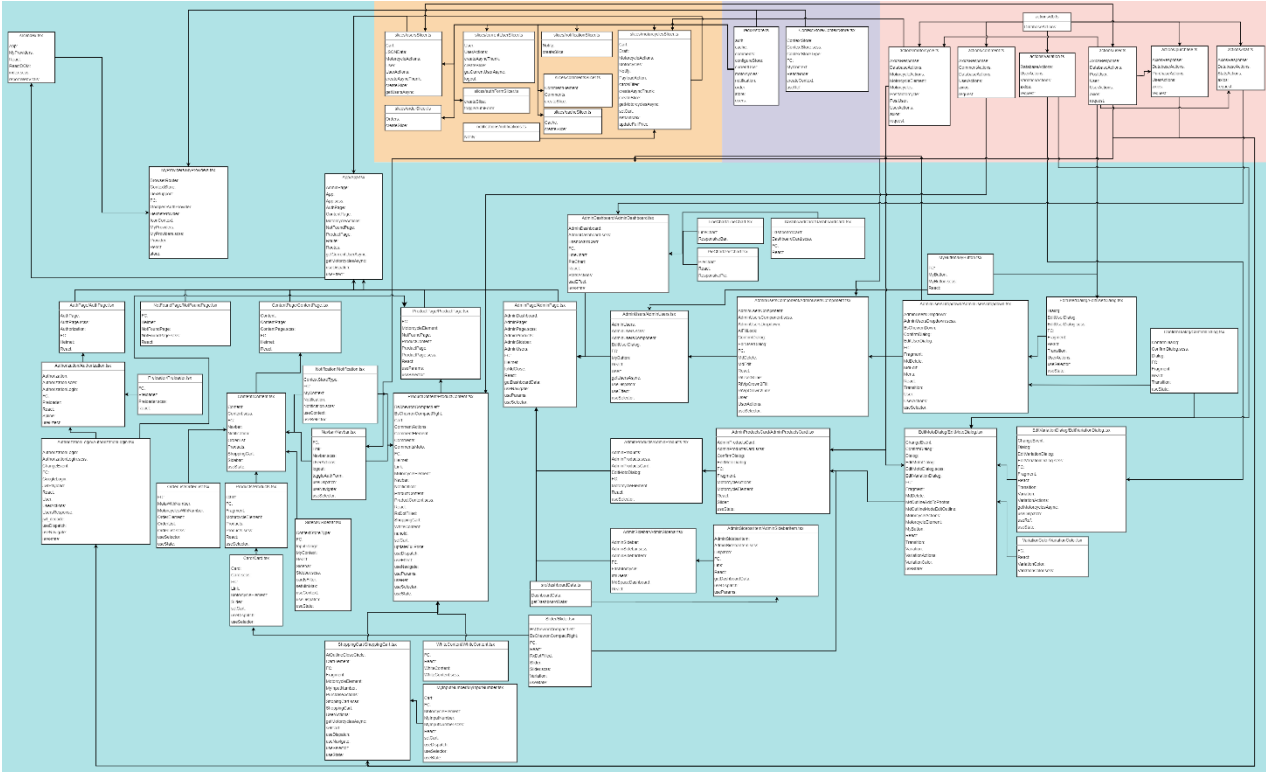
		Шумський О.В.			Житомирська політехніка.23.121.14.000 – ПЗ	Арк.
		Фуріхата Д.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		39

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. МОТОТЕК [Електронний ресурс] / admin. – 2011. – Режим доступу до ресурсу: <https://mototek.com.ua/uk/>
2. Мотомаркет [Електронний ресурс] / admin. – 2015. – Режим доступу до ресурсу: <https://ua-motors.com.ua/ua/>
3. Мотоцикли [Електронний ресурс] / admin. – 2012. – Режим доступу до ресурсу: <https://motodom.ua/ua/>
4. Документація по Daisy UI [Електронний ресурс] / admin. – 2020. – Режим доступу до ресурсу: <https://daisyui.com/>
5. Бібліотека React Icons [Електронний ресурс] / admin. – 2020. – Режим доступу до ресурсу: <https://react-icons.github.io/react-icons/>
6. Бібліотека react-helmet [Електронний ресурс] / admin. – 2019. – Режим доступу до ресурсу: <https://www.npmjs.com/package/react-helmet>
7. Headless UI [Електронний ресурс] / admin. – 2019. – Режим доступу до ресурсу: <https://headlessui.com/>
8. Redux Fundamentals, Part 8: Modern Redux with Redux Toolkit [Електронний ресурс] / admin. – 2019. – Режим доступу до ресурсу: <https://redux.js.org/tutorials/fundamentals/part-8-modern-redux>
9. Home | nivo [Електронний ресурс] / admin. – 2021. – Режим доступу до ресурсу: <https://nivo.rocks/>
10. The Clean Architecture на TypeScript и React. Часть 1: Основы [Електронний ресурс] / RostislavDugin. – 2020. – Режим доступу до ресурсу: <https://habr.com/ru/post/499078/>

		Шумський О.В.			Житомирська політехніка.23.121.14.000 – ПЗ	Арк.
		Фуріхата Д.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		40

ДОДАТКИ



Вихідний код програмного продукту

motorcyclesSlicer.ts

```
import {createAsyncThunk, createSlice, Draft, PayloadAction} from
"@reduxjs/toolkit";
import {Cart, Motorcycles} from "../../Types";
import {MotorcycleActions} from "../../actions/motorcycle";
import {Notify} from "../../notifications/notifications";

interface State {
  motorcycles: Motorcycles
  filtered: Motorcycles
  min: number
  max: number
  model: string
  brand: string
  cart: Cart
  fullPrice: string
}

const getInitialCart = () => {
  let Stored = localStorage.getItem('cart')
  return JSON.parse(Stored ?? '[]')
}

const getFullPrice = (): string => {
  let motorcycles = JSON.parse(localStorage.getItem('motorcycles') ?? '[]')
  let cartData = getInitialCart()
  let sum: number = 0
  for (let [id, variation, number] of cartData)
    for (let item of motorcycles)
      if (item.id === id)
        sum += item.price * number
  return sum.toString()
}

const initialState: State = {
  motorcycles: [],
  filtered: [],
  min: 0,
  max: 10_000_000,
  model: '',
  brand: 'All',
  cart: getInitialCart(),
  fullPrice: getFullPrice()
}

const filterFunc = (brand: string, motorcycles: Motorcycles, model: string,
min: number, max: number) => {
  let data =
    brand === 'All'
      ?
        motorcycles
      :
        motorcycles
        .filter((item: any) =>
          item.brand.toLowerCase()
            .includes(
              brand.toLowerCase()
            )
        )
}
```

```

    )
    return data.filter((item: any) =>
      item.model
        .toLowerCase()
        .includes(model.toLowerCase())
        && item.price >= min
        && item.price <= max)
  }
}

export const getMotorcyclesAsync = createAsyncThunk(
  'motorcycles/getMotorcyclesAsync',
  async () => {
    let motoAction = await new MotorcycleActions()
    return await motoAction.getMotorcycles()
  }
)

export const motorcyclesSlicer = createSlice({
  name: 'motorcycles',
  initialState,
  reducers: {
    setMotorcycles: (state: Draft<State>, action:
PayloadAction<Motorcycles>) => {
      state.motorcycles = action.payload
      state.fullPrice = getFullPrice()
    },
    setFiltered: (state: Draft<State>, action: PayloadAction<Motorcycles>)
=> {
      state.filtered = action.payload
    },
    cardsFilter: (state: Draft<State>, action: PayloadAction<{model:
string, brand: string}>) => {
      let {model, brand} = action.payload
      state.model = model
      state.brand = brand
      state.filtered = filterFunc(brand, state.motorcycles, model,
state.min, state.max)
    },
    setMinMax: (state: Draft<State>, action: PayloadAction<{min: number,
max: number}>) => {
      let {min, max} = action.payload
      state.min = min
      state.max = max
      state.filtered = filterFunc(state.brand, state.motorcycles,
state.model, min, max)
    },
    setCart: (state: Draft<State>, action: PayloadAction<Cart>) => {
      state.cart = action.payload
      localStorage.setItem('cart', JSON.stringify(action.payload))
      state.fullPrice = getFullPrice()
    },
    updateFullPrice: (state: Draft<State>) => {
      state.fullPrice = getFullPrice()
    }
  },
  extraReducers: (builder) => {
    builder.addCase(getMotorcyclesAsync.fulfilled, (state, action) => {
      state.motorcycles = action.payload
      state.filtered = action.payload
    })
    builder.addCase(getMotorcyclesAsync.rejected, () => {
      new Notify().DbError()
    })
  }
})

```

```

))

export const {cardsFilter, setMinMax, setCart, updateFullPrice} =
motorcyclesSlicer.actions
export default motorcyclesSlicer.reducer

```

validateTokenController.php

```

<?php
use \Firebase\JWT\JWT;
use \Firebase\JWT\Key;

$jwt = $_GET['jwt'];

if ($jwt)
{
    try
    {
        $decoded = JWT::decode($jwt, new Key($key, algorithm: 'HS256'));
        http_response_code(response_code: 200);
        echo json_encode($decoded->data);
    }
    catch (Exception $e)
    {
        http_response_code(response_code: 401);
    }
}
else
{
    http_response_code(response_code: 401);
}

```

usersController.php

```

<?php
use core\Core;
$tableName = 'users';

use \Firebase\JWT\JWT;
use \Firebase\JWT\Key;

$jwt = $_GET['jwt'];

switch ($_SERVER['REQUEST_METHOD']) {
    case 'GET':
        $fields = ['id', 'login', 'firstName', 'lastName', 'role',
'dateOfBirth'];
        $data = Core::getInstance()::$db->select(tableName: $tableName,
fieldsList: $fields);
        echo json_encode($data, JSON_UNESCAPED_UNICODE);
        break;

    case 'POST':
        $decoded = JWT::decode($jwt, new Key($key, algorithm: 'HS256'));
        if ($decoded->data->role === 'admin' || $decoded->data->role ===
'moderator') {
            $data = [
                'login' => $_GET['login'],
                'password' => md5($_GET['password']),
                'firstName' => $_GET['firstName'],
                'lastName' => $_GET['lastName'],
                'role' => $_GET['role'],
                'dateOfBirth' => $_GET['dateOfBirth']

```

```

];
Core::getInstance()::$db->insert(tableName: $tableName,
newRowArray: $data);
} else {
    http_response_code(response_code: 403);
}

break;

case 'DELETE':
    $decoded = JWT::decode($jwt, new Key($key, algorithm: 'HS256'));
    if ($decoded->data->role === 'admin' || $decoded->data->role ===
'moderator') {
        $data = ['id' => $_GET['id']];
        Core::getInstance()::$db->delete(tableName: $tableName,
conditionArray: $data);
    } else {
        http_response_code(response_code: 403);
    }
    break;

case 'PUT':
    $decoded = JWT::decode($jwt, new Key($key, algorithm: 'HS256'));
    if ($decoded->data->role === 'admin' || $decoded->data->role ===
'moderator') {
        $id = [
            'id' => $_GET['id']
        ];
        $data = [
            'login' => $_GET['login'],
            'firstName' => $_GET['firstName'],
            'lastName' => $_GET['lastName'],
            'role' => $_GET['role'],
            'dateOfBirth' => $_GET['dateOfBirth']
        ];
        Core::getInstance()::$db->update(tableName: $tableName,
newValuesArray: $data, conditionArray: $id);
    } else {
        http_response_code(response_code: 403);
    }
    break;
}
}

```