

Міністерство освіти і науки України
Національний університет «Львівська політехніка»
Інститут комп'ютерних наук та інформаційних технологій
Кафедра автоматизованих систем управління



Звіт
до лабораторної роботи №3
з дисципліни
“Веб-технології та розробка веб-застосунків”
на тему
“Фронтенд-фреймворки”

Виконав:
студент групи ОІ-26
Чабанов Павло

Прийняла:
Троян О. А.

Львів – 2025

Тема: Фронтенд-фреймворки.

Мета: Отримати практичний досвід роботи з одним із сучасних фронтенд-фреймворків. Навчитись розробляти веб-сторінки за допомогою компонентів, керувати станами компонент та працювати з маршрутизацією для переходу між різними відображеннями (сторінками).

Хід виконання роботи

Варіант 23

Веб-сайт платформи для оренди спортивного обладнання

1. Створити компоненти для картки обладнання, списку оренд та форми оплати.
2. У компоненті "Обладнання" реалізувати фільтрацію за типами спорту (футбол, теніс, лижі тощо).
3. Реалізувати маршрутизацію між сторінками без перезавантаження сторінки браузера.

Для того, щоб виконати дане завдання, потрібно вибрати та встановити один із запропонованих фреймворків. Для себе та свого сайту я обрав саме React, тому що він швидкий, гнучкий, популярний та доволі простий. Для встановлення серверу я використав Vite. Він швидший за create-react-app. Основною відмінністю React є написання Html коду прямо в js файлі. Для наочності, такі файли мають розширення jsx. Для його установки в проект я скористався консольною командною:

```
npm create vite@latest sport-equipment-rental -- --template react
```

Потім я почав створювати файли, починаючи з App.jsx. Файл App.jsx у React-додатку виконує кілька ключових функцій:

1. **Головний компонент додатку** - це кореневий компонент, який відповідає за загальну структуру інтерфейсу користувача.
2. **Налаштування маршрутизації** - у нашому додатку "SportRent" саме в App.jsx налаштовані маршрути, які визначають, яка сторінка буде показана користувачу залежно від URL (сторінка обладнання, мої оренди, оплата).

3. **Об'єднання компонентів** - включає спільні для всіх

сторінок компоненти, такі як навігаційна панель (Navbar) та футер.

4. **Точка входу** - це перший React-компонент, який рендериться в DOM після запуску додатку.

У моєму проєкті "SportRent" файл App.jsx відповідає за підключення React Router, який забезпечує навігацію між різними сторінками без перезавантаження, а також за відображення спільних елементів інтерфейсу на всіх сторінках. Без цього файлу не було б можливості перемикатися між сторінками "Обладнання", "Мої оренди" та "Оплата".

Для запуску потребується Node.js, також потрібно ввести команду:

npm run dev

Це запустить сервер, та дасть нам посилання на нього

```
1  import React from 'react';
2  import './EquipmentCard.css';
3
4  function EquipmentCard({ id, name, price, image, onRent }) {
5    return (
6      <div className="equipment-card">
7        <img src={image} alt={name} />
8        <div className="info">
9          <h3>{name}</h3>
10         <p>Ціна: <span className="price">{price} грн/день</span></p>
11         <button
12           className="rent-button"
13           onClick={() => onRent({ id, name, price, image })}
14         >
15           Орендувати
16         </button>
17       </div>
18     </div>
19   );
20 }
21
22 export default EquipmentCard;
```

Рис. 1. Текст файлу EquipmentCard.jsx

На Рис. 1 зображено текст файлу EquipmentCard.jsx. Це файл для відображення картки обладнання. Вона складається з:

1. Імпорту реакту та стилів для картки
2. Функція
3. Експорт

Функція – в даному випадку, після проведення певних дій, повертає html вираз.

Експорт - робить компонент EquipmentCard доступним для використання в інших файлах проекту.

```
1  import React from 'react';
2  import './EquipmentFilter.css';
3
4  function EquipmentFilter({ onFilterChange }) {
5      const sportTypes = [
6          'Всі',
7          'Футбол',
8          'Баскетбол',
9          'Теніс',
10         'Лижі',
11         'Сноуборд',
12         'Велоспорт'
13     ];
14
15     return (
16         <div className="filter-container">
17             <h3>Фільтр за видом спорту</h3>
18             <div className="filter-buttons">
19                 {sportTypes.map(type => (
20                     <button
21                         key={type}
22                         onClick={() => onFilterChange(type)}
23                         className="filter-button"
24                     >
25                         {type}
26                     </button>
27                 ))}
28             </div>
29         </div>
30     );
31 }
32
33 export default EquipmentFilter;
```

Рис. 2. Код файлу EquipmentFilter.jsx

На Рис. 2 ми можемо бачити код файлу EquipmentFilter.jsx. В даному випадку функція ініціалізує масив типів спорту, а потім повертає мапу з кнопок, які мають певні назви, ключи і т.д. В результаті отримуємо результат, який показано на Рис. 3.

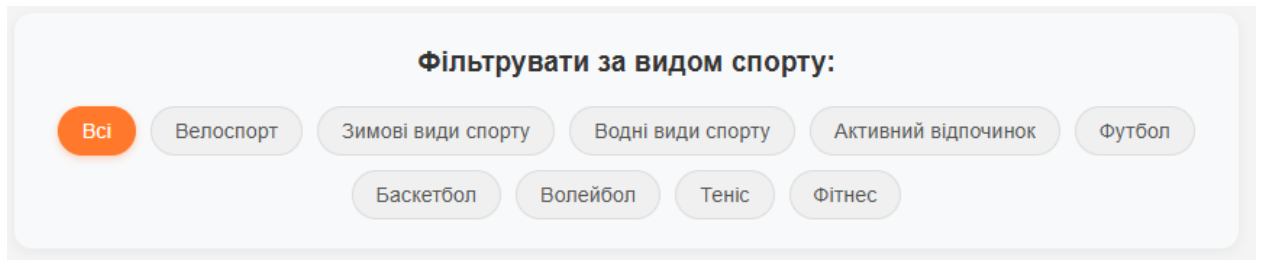


Рис. 3. Меню фільтрації

Також я змінив вигляд на сторінці Мої оренди. Я додав картинки та більше кнопок. Для видалення зробив окрему кнопку іншого кольору. Стан оплати я виніс в верхній правий кут, для кращого розуміння, а при оплаті вона змінює колір на зелений, для наочності результату. Результат показано на Рис. 4. Також додав кнопку оплатити все.

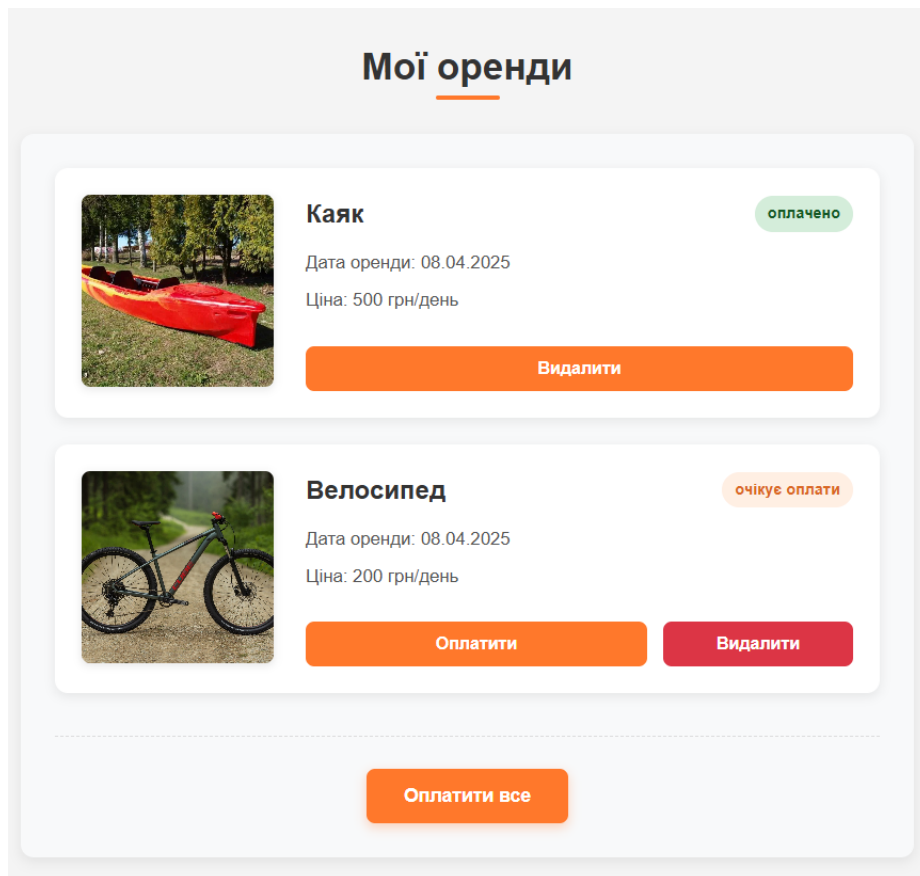


Рис. 4. Список моїх оренд

Також я змінив сторінку оплати. Тепер можна оплачувати кілька різних обладнань, та сам процес оплати став швидшим. Результат можна побачити в додатку В.

У межах цієї лабораторної роботи я реалізував покращене інтелектуальне редагування полів введення даних платіжної картки. Зокрема, для поля введення номера картки встановлено обмеження — користувач не зможе ввести більше ніж 16 символів. Крім того, для зручності читання та перевірки, після кожних чотирьох цифр автоматично додається пробіл. Аналогічний підхід застосовано і до поля введення дати - після кожних двох символів автоматично додається роздільник у вигляді символу /. Це дозволяє користувачеві вводити дані у звичному, стандартизованому форматі, зменшуючи ризик помилок.

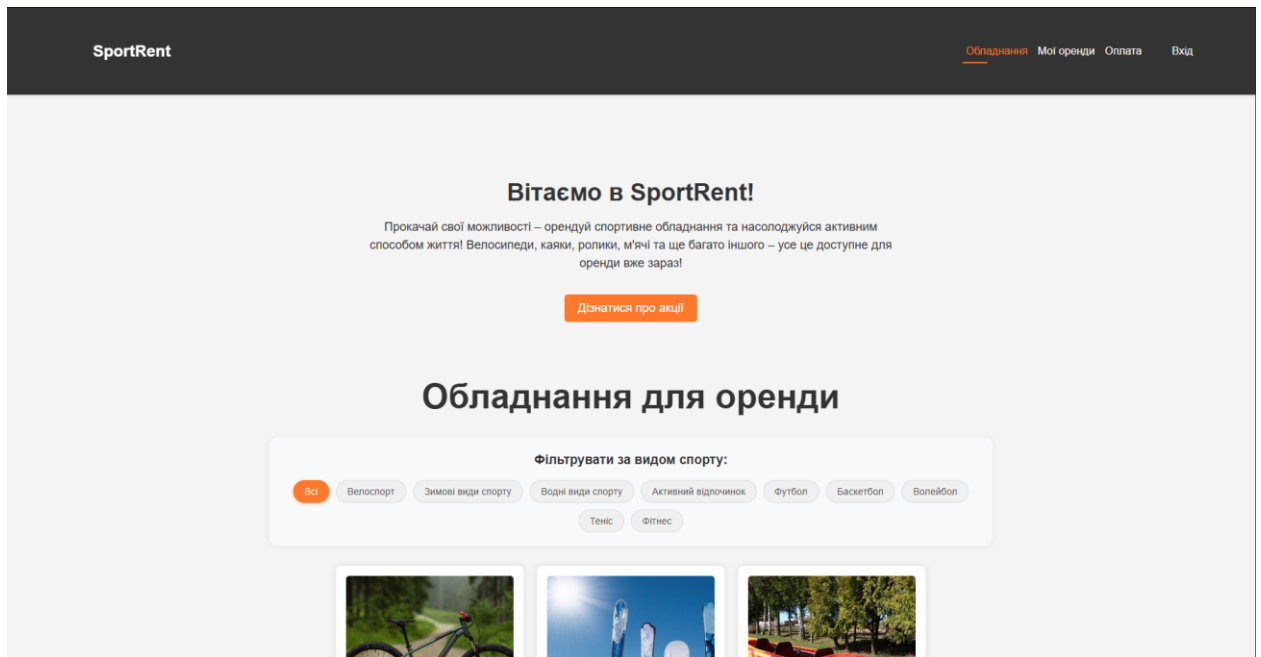
Таке вдосконалення не лише покращує користувацький досвід, а й підвищує рівень безпеки. Автоматичне форматування допомагає уникнути введення небажаних або шкідливих значень, що можуть призвести до SQL-ін'єкцій або інших типів кіберзагроз, пов'язаних із передачею даних до бази даних.

Висновок

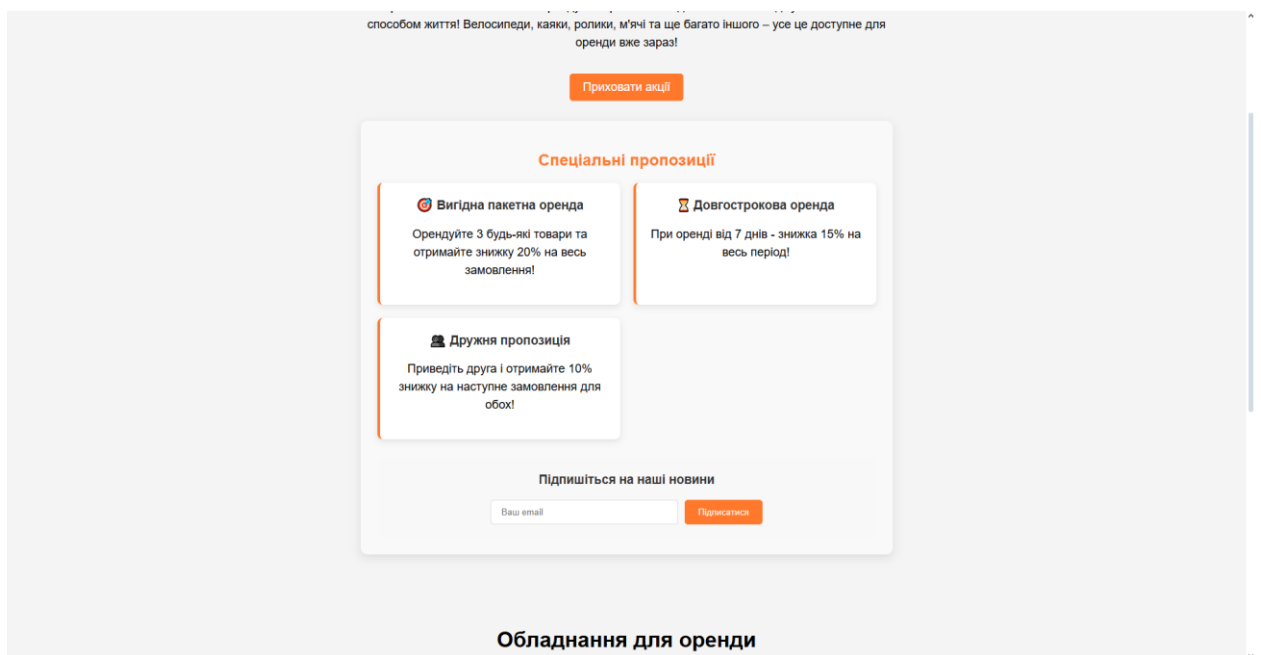
Отже, на цій лабораторній роботі я навчився застосовувати основні функціональні можливості мови програмування JavaScript, такі як маніпуляції з елементами веб-сторінки, обробка подій, цикли та умови, для реалізації інтерактивності та динамічності веб-сторінок. Я оновив свій сайт з минулої лабораторної роботи додавши багато взаємодії з JavaScript, через що він став виглядати ще красивіше та інтерактивніше.

Додатки

Додаток А (скріншот головної сторінки)



Додаток Б (скріншот прихованого меню на головній сторінці)




Додаток В (скріншот сторінки оплати)

Оплата оренди

Деталі оплати

Ваше замовлення:

☒



Футбольний м'яч


50 грн/день

Кількість днів:

1

Загалом: 50 грн

☒



Лижі

300 грн/день

Кількість днів:

1

Загалом: 300 грн

Загальна сума:

350 грн

Ім'я на картці

Номер картки

XXXX XXXX XXXX XXXX

16 цифр, автоматичне форматування

Термін дії

MM/YY

Формат: MM/YY

CVV

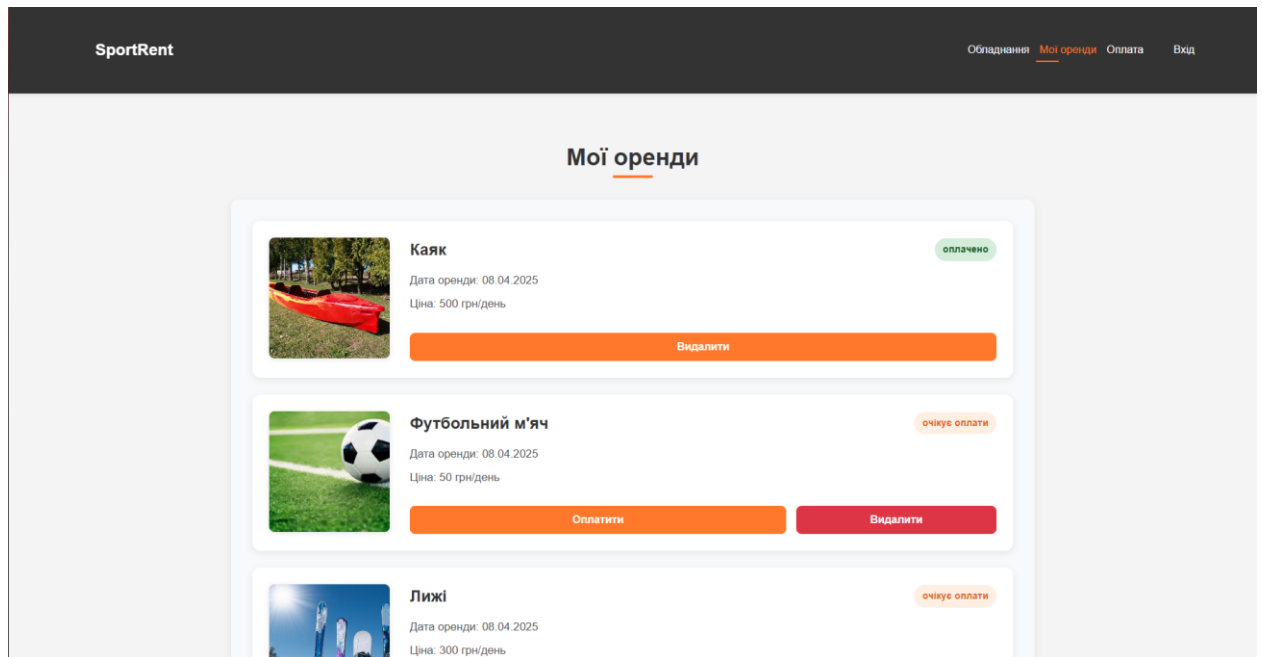
XXX

3-4 цифри

Скасувати

Сплатити 350 грн

Додаток Г (скріншот сторінки “Мої оренди”)



Додаток Г (скріншот подробиць до обладнання)

