МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ІВАНА ФРАНКА

Факультет прикладної математики та інформатики

(повне найменування назва факультету)

Кафедра інформаційних систем

(повна назва кафедри)

ДИПЛОМНА РОБОТА

Розробка інтернет магазину музичних інструментів

Виконав: студент групи ПМІ-44

спеціальності 122 – комп'ютерні науки

(шифр і назва спеціальності)

Яцола А. В.

(підпис) (прізвище та ініціали)

Керівник доц. Горлач В. М.

(підпис) (прізвище та ініціали)

Рецензент

(підпис) (прізвище та ініціали)

2023

Зміст

ВСТУП

Актуальність теми. В сучасному світі все більше людей використовують Інтернет для покупок, електронна торгівля стала надзвичайно популярною. Електронна торгівля, або е-комерція, є процесом купівлі та продажу товарів та послуг через Інтернет. Завдяки електронній торгівлі, покупці можуть зробити покупки з будь-якого місця у світі та отримати товари безпосередньо до свого дому або офісу. Водночас, продавці можуть знизити витрати на збут, зокрема на складські приміщення та зарплату персоналу, а також розширити свій ринок збуту та залучити нових клієнтів через ефективнішу рекламу та маркетинг. У зв'язку з розвитком технологій та зростанням популярності онлайн-торгівлі, електронна торгівля стала невід'ємною частиною бізнесу у багатьох галузях економіки.

Інтернет-магазин музичних інструментів - це актуальний вид електронної комерції, який стає все більш популярним серед покупців, які шукають якісні музичні інструменти за доступною ціною. Таким чином, розробка сайту для магазину музичних інструментів стає все більш важливою задачею для підприємців, що бажають розширити свій бізнес. Спроектований сайт має задовольняти потреби клієнтів та забезпечувати оптимальний спосіб продажу товарів, що не тільки підвищує конкурентоспроможність магазину, але й дозволяє збільшити прибуток. Таким чином, розробка інтернет-магазину музичних інструментів має великий потенціал, що може бути реалізовано за допомогою використання сучасних технологій веб-розробки та дизайну.

Мета дипломної роботи. Метою даної роботи є ознайомлення зі способами розробки інтернет магазину музичних інструментів, вибір оптимальних технологій та інструментів для реалізації проекту. При розробці сайту для магазину необхідно враховувати потреби користувачів, їхні переваги та

очікування, а також забезпечити максимальну зручність та безпеку під час проведення операцій з покупкою та продажем товарів. В результаті цієї роботи буде обрано оптимальний підхід до розробки інтернет магазину музичних інструментів, що дозволить створити ефективний та привабливий сайт для клієнтів та забезпечити успішну реалізацію проекту.

СУТЬ ДИПЛОМНОЇ РОБОТИ

Розділ 1. Вибір та обґрунтування засобів розробки інтернет магазину музичних інструментів.

При проектуванні будь-якої системи важливо обрати правильне рішення з точки зору програмної реалізації. Обране рішення має відповідати певним вимогам, основними з яких є: функціональність обраного методу, практичність, простота реалізації. На сьогоднішній день є дуже багато засобів та інструментів, які можуть допомогти з програмною реалізацією тієї чи іншої задачі і при виборі інструментів розв'язку потрібно перш за все визначити особливості, які має та чи інша система. В нашому випадку інтернет магазин музичних інструментів є вебсайтом. Також, необхідно спроектувати базу даних, в якій зберігатимуться дані про товари, замовлення користувачів та інформація про самих користувачів. Тому при виборі засобів реалізації необхідно звертати увагу на ті, які найкраще підходять для розробки вебсайтів та обрати модель бази даних для проектованої системи. Проаналізувавши стек технологій, які найбільше підходять для реалізації даної системи були обрані наступні технології для її реалізації:

- React;
- Redux;
- Firebase.

Опишемо детальніше переваги кожної з даних технологій, обраних для створення інтернет магазину.

React – це одна з найпопулярніших бібліотек Javascript, що була розроблена в 2013 році. Дана бібліотека використовується для розробки веб – додатків, допомагає у розробці односторінкових та багатосторінкових програм[1]. Бібліотека в програмуванні – це додаткові можливості, які полегшують розробку. Це незалежний код, який можна підключити до проекту як незалежний модуль.

Через бібліотеки можна додати проекту готовий додатковий функціонал. В свою чергу Javascript – це одна з найпопулярніших мов програмування, призначена для веб – розробки. Основними перевагами Javascript вважають:

- швидкість – Javascript є “інтерпретованою” мовою, тому час компіляції програм є швидшим ніж у багатьох інших мов програмування;
- простота у вивченні;
- широка бібліотека – надає багато додаткових можливостей при розробці програм;
- функціональність;
- універсальність[11].

Водночас, серед недоліків основними є підтримка браузерів (кожен браузер може по різному інтерпретувати JavaScript та безпека даних користувача. Як уже було сказано раніше, в проектованій системі буде використовуватися одна з бібліотек Javascript – React. Тож варто детальніше проаналізувати плюси та мінуси даної бібліотеки. До переваг даної технології відносять:

- Простота у вивченні;
- Можливість повторного використання компонентів;
- Можливість створення односторінкових додатків – так званих ”Single Page Application (SPA)” – тобто при переході на нову сторінку у розробленому додатку не відбувається перезагрузки сторінки у браузері;
- Незмінність стану компонента після його встановлення;
- React Developer Tools – набір інструментів, що допомагають при розробці коду[1].

Серед недоліків можна виділити:

- Погана документація.
- Швидкі темпи розвитку.
- Охоплює не так багато технологій розробки[2].

Найбільшим конкурентом для React є Angular.

Angular – це фреймворк, написаний на Typescript, для створення веб-додатків. Angular має багато бібліотек, які в свою чергу містять багато функцій, в тому числі керування формами, маршрутизацію та багато іншого.

Основними перевагами даної технології є:

- Підтримка відкладеного завантаження – з самого початку завантажуються лише ті модулі , які потрібні для роботи в даний момент.
- Наявність багатьох пакетів, які полегшують процес розробки.
- Двостороння прив’язка даних.
- Використання Typescript[3].

Серед недоліків виділяють:

- Важкий у вивченні.
- Швидкі темпи розвитку.
- Займає багато пам’яті.
- Наявність ієрархічної деревоподібної структури[4].

Однак, React все таки має деякі переваги перед Angular, такі як: HTML-подібний синтаксис, який дозволяє використовувати шаблони та обширну документацію. Зручно створювати Single Page Application. Окрім того, подібний синтаксис дає можливість приділяти більше уваги самому процесу написання коду не витрачаючи час на специфічний синтаксис як в Angular. Також перевагою є одностороння прив’язка даних та наявність більшої кількості допоміжних інструментів, які допомагатимуть в розробці[5].

Також при розробці також використовуватиметься *Redux*.

Redux – це менеджер станів, який на відмінну від локального стейту в класових компонентах, та хуках *useState* у функціональних компонентах, дозволяє зберігати стани всього додатку в одному місці. Це місце називається сховищем(store). Redux зручно використовувати коли мова йде про односторінкові додатки, в яких досить складне управління станами. І хоча

першочергово Redux розроблявся для використання спільно з React , зараз його можна використовувати також спільно з jQuery та Angular. Основне призначення Redux – управління станами в додатку. Всі змінні стану компоненти

відправляють безпосередньо до сховища, яке, як уже було зазначено, зберігає всі стани, які є в додатку. В Redux компоненти не зв'язані одне з одним безпосередньо, всі зміни в компонентах та взаємодія між ними проходять через сховище. Також стан кожного компонента встановлюється за допомогою сховища. Інформацію про зміну стану компонент відправляє у сховище, лише ініціюючи зміну стану[12].

Дешо схожим на *redux* є *context,* а саме, в можливості обміну даними між компонентами. Зазвичай, у додатку React дані передаються зверху вниз (від батьківського компонента – *App* до дочірнього). Передача даних відбувається через *props.* Але такий спосіб передачі даних може бути доволі громіздким, особливо якщо компонент знаходиться на нижньому рівні дерева. Контекст надає можливість обміну даними між компонентами без необхідності передачі параметрів через кожен рівень дерева. Контекст призначений для обміну даними, які можна вважати глобальними для дерева компонентів React. Наприклад, користувач, автентифікований в системі на даний момент. Контекст переважно застосовується коли деякі дані повинні бути доступними багатьом компонентам, які знаходяться на різних рівнях дерева React[12].

Незвачаючи на всі переваги, які мають контексти, при розробці інтернет магазину музичних інструментів використовуватиметься Redux, тому що він є зручнішим у використанні.

Ще одним інструментом для створення інтернет магазину музичних інструментів є Firebase.

Firebase надає можливість використовувати багато функцій та можливостей для розробки веб-додатків. Firebase – це BaaS(Backend as a Service) і

класифікується як програма, що містить нереляційні бази даних (NoSQL), які зберігають дані у документах. Документ – це сукупність пар типу ключ-значення, а декілька пар документів в свою чергу утворюють колекції[6].

Основні можливості, які Firebase надає для користувача, це:

- Можливість автентифікації користувача.
- База даних.
- Хостинг.
- Аналітика даних[6].

Основними перевагами Firebase вважають:

- Можливості бази даних.
- Прості налаштування.
- Багато функцій та можливостей.
- Коротка, зрозуміла та проста документація[7].

Серед недоліків варто виділити наступні:

- Можливості запитів є обмеженими.
- Слабка підтримка IOS.
- Міграція даних також є обмеженою[7].

Подібним аналогом до Firebase можна вважати AWS.

Amazon Web Services (AWS) – це популярна платформа хмарних обчислень, яка надає користувачам безліч корисних функцій та можливостей. Основними перевагами AWS є:

- Широкий вибір інструментів для розробки.
- Наявність інструментів управління.
- Наявність бази даних.
- Швидкість.
- Обчислювальний сервіс[8].

Водночас, AWS має багато схожих функцій з Firebase таких як:

- Наявність бази даних.
- Наявність хостингу.
- Безпека.
- Аналітика[9].

Проте, при розробці системи обрано саме Firebase, так як він має багато необхідних інструментів, які будуть використані в проектованій системі (авторизація, різні можливості бази даних).

При розробці інтернет магазину буде використано деякий функціонал з *Firebase,* а саме, базу даних *Cloud Firestore* та *Firebase Authentication* для можливості авторизації користувача.

Cloud Firestore - це гнучка, масштабована NoSQL(нереляційна) база даних, для мобільних та веб-розробок від Firebase та GoogleCloud. *Cloud Firestore* синхронізує дані, забезпечуючи їх оновлення в реальному часі, дозволяє створювати адаптивні програми, які працюватимуть незалежно від підключення до інтернету. Також є можливість інтеграції Cloud Firestore з продуктами Firebase та Google Cloud. Можна виділити такі основні переваги, які надає застосування Cloud Firestore:

- Дані зберігаються у документах, документи складають колекції.
- База даних є гнучкою.
- Забезпечує синхронізацію даних в реальному часі.
- Забезпечує інтеграцію з Google Cloud та Firebase
- Гарантує безпеку даних[10]

В інтернет магазині Cloud Firestore міститме колекцію з каталогом товарів, та колекцію з історією замовлень.

Firebase Authentication – спеціальна служба, яка має можливість автентифікувати користувачів. Firebase Authentication надає бекенд і готові бібліотеки для реалізації автентифікації користувача. Автентифікація здійснює перевірку на принадлежність ідентифікатора певному користувачеві, допомагає

ідентифікувати його. А ідентифікація користувача є дуже важливою в інтернет магазині, адже лише після ідентифікації в системі можна здійснити замовлення. Firebase Authentication підтримує різні варіанти автентифікації користувача:

- Автентифікація з використанням пароля та електронної пошти.
- Автентифікація з використанням номера телефону.
- Автентифікація з використанням Google акаунту.
- Автентифікація з використанням акаунту на GitHub.
- Автентифікація з використанням акаунту в соціальних мережах (Facebook, Twiter).

Також необхідно обрати редактор для написання коду. Вибір стояв між двома найбільш популярними та ефективними – *Visual Studio Code* та *JetBrains WebStorm*. Обидва мають чимало можливостей:

- *emmet* (автодоповнення коду);
- багатий функціонал та кросплатформенність;
- підсвітка синтаксису;
- влаштована інтеграція з *Git*;
- налагодження (*Debug*).

Для розробки системи, в якості редактора коду було обрано *Visual Studio Code*.

Розділ 2. Розробка інтернет магазину музичних інструментів

В першу чергу, створюємо новий React-додаток, використовуючи команду npx create-react-app online-store, де ‘online-store’ – назва папки проєкту. Ми отримуємо початковий набір реакту, який далі будемо редагувати та доповнювати власними елементами.

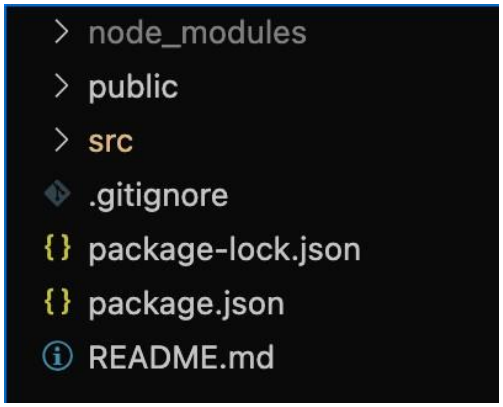

Рисунок 2.1

В папці src я створюю папку pages, в якій розміщую файли сторінок інтернет магазину.

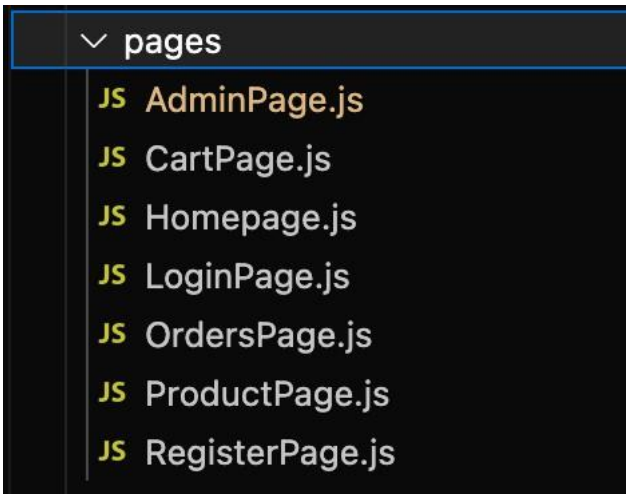

Рисунок 2.2

Також налаштував роутинг до всіх сторінок

```
<Routes>
  <Route path='/login' exact element={<LoginPage />} />
  <Route path='/register' exact element={<RegisterPage />} />
  <Route path='/' exact element={<ProtectedRoutes><Homepage /></ProtectedRoutes>} />
  <Route path='/product/:productid' exact element={<ProtectedRoutes><ProductPage /></ProtectedRoutes>} />
  <Route path='/cart' exact element={<ProtectedRoutes><CartPage /></ProtectedRoutes>} />
  <Route path='/orders' exact element={<ProtectedRoutes><OrdersPage /></ProtectedRoutes>} />
  <Route path='/admin' exact element={<ProtectedRoutes><AdminPage /></ProtectedRoutes>} />
</Routes>
```

Рисунок 2.3

На рисунку 2.3 можна побачити, що всі сторінки, окрім сторінок логіну та реєстрації є захищеними, тобто тільки користувач, який авторизувався може переглядати вміст магазину та робити замовлення.

Наступним кроком є підключення до firestore database. Спочатку на самому сервісі firestore database я заповнив колекцію products майбутніми продуктами. Кожен з продуктів це об’єкт, який містить наступні ключі:

- name – назва продукту
- price – ціна продукту
- category – категорія, до якої відноситься продут
- description – опис продукту
- imageURL – посилання на зображення продукту

На рисунку 2.4 можна побачити, як це виглядає на самому сервісі firestore database.

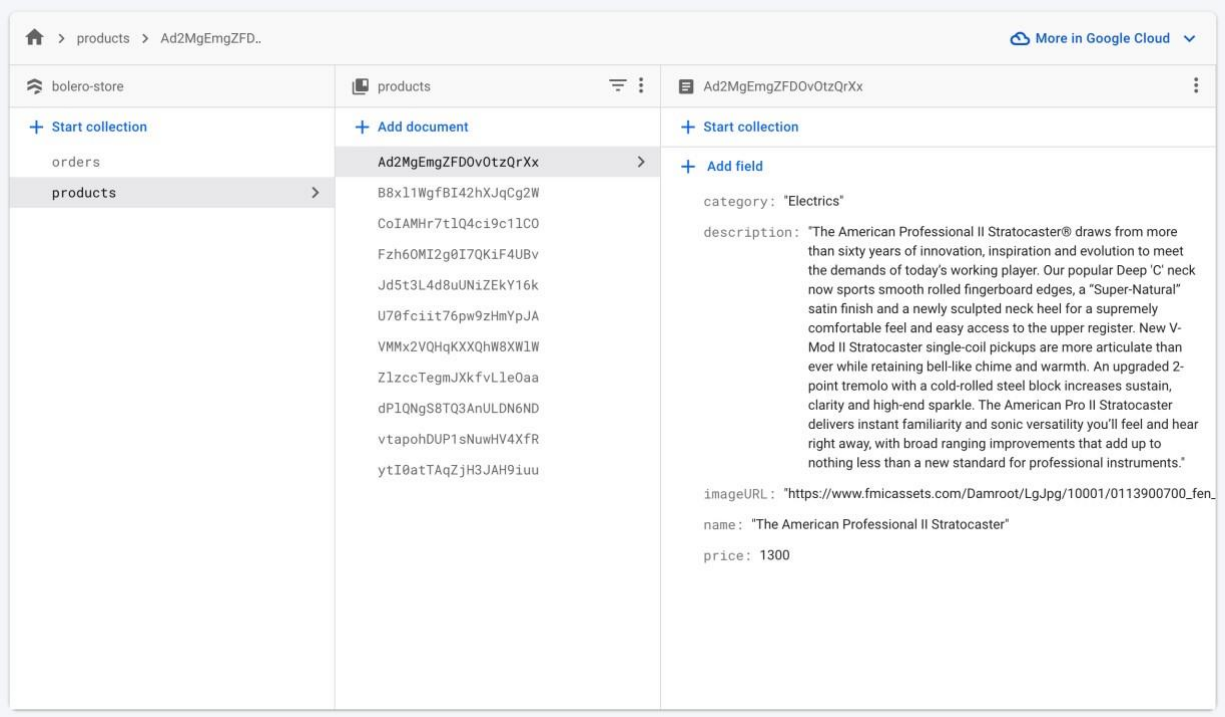

Рисунок 2.4

Далі мені потрібно отримати дані з firestore database, для цього я використав функцію getDocs().

```
async function getData() {
    try {
        setLoading(true);
        const products = await getDocs(collection(fireDB, "products"))
        const productsArray = []

        products.forEach((doc) => {
            const obj = {
                id: doc.id,
                ...doc.data()
            }

            productsArray.push(obj);
            setLoading(false)
        });

        setProducts(productsArray);
    } catch (error) {
        console.log(error)
        setLoading(false)
    }
}
```

Рисунок 2.5

Отримавши дані з бази даних, я можу заповнити головну сторінку інтернет магазину продуктами. Застилізувавши сторінку, я отримав наступний результат:

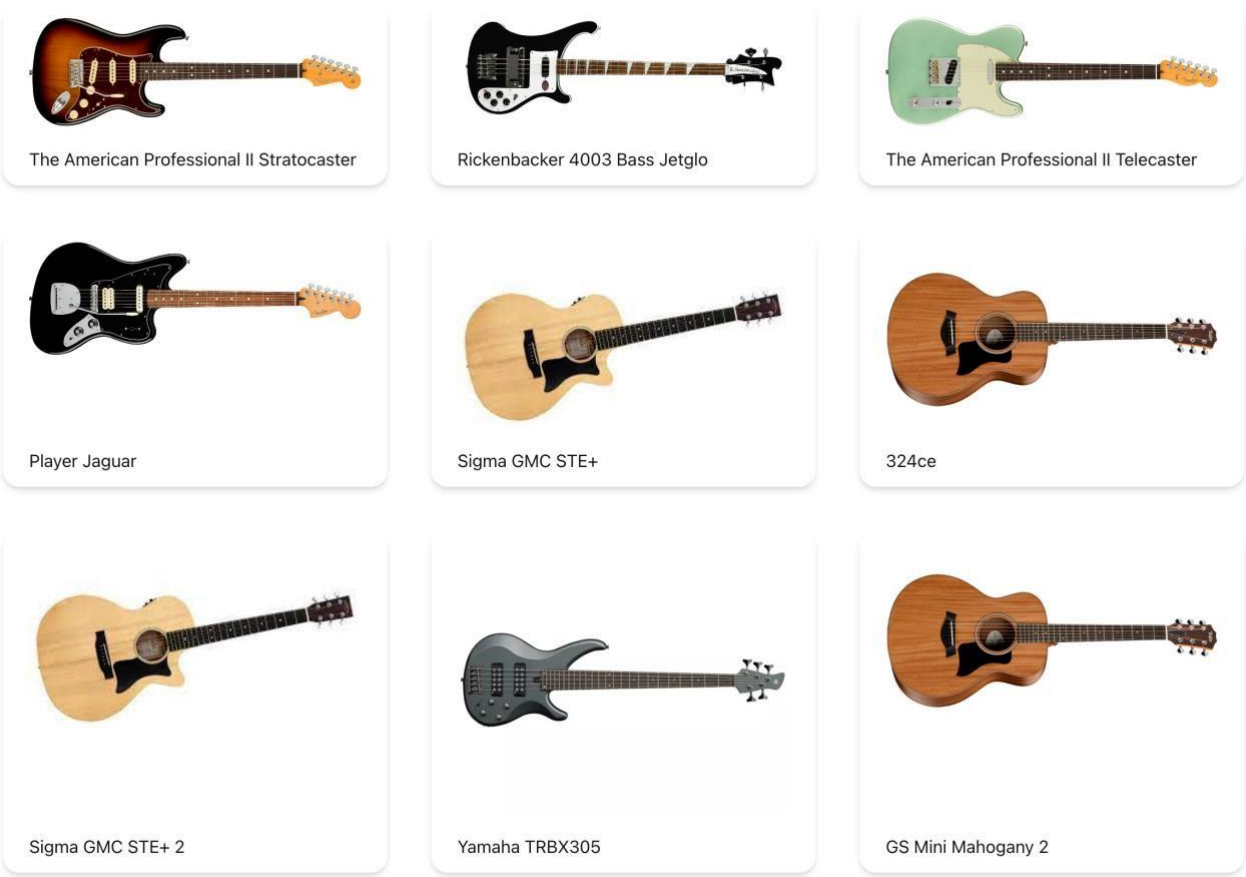

Рисунок 2.6

При наведенні на продукт з’являється додаткова інформація.

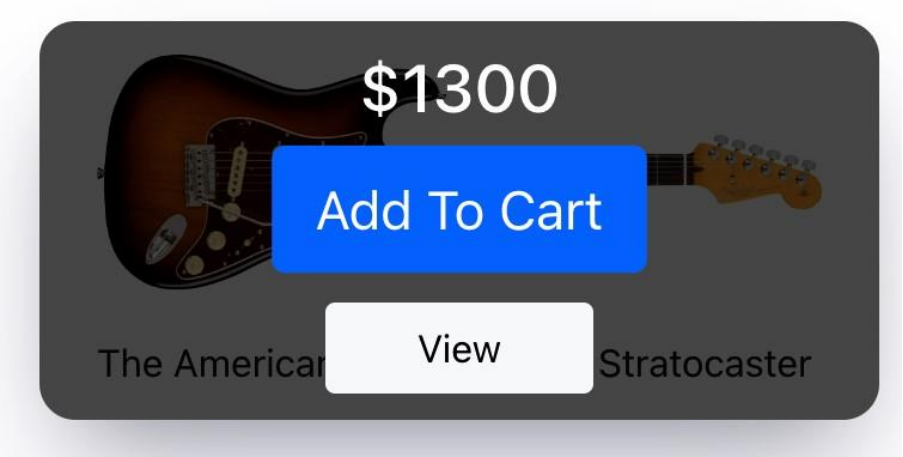

Рисунок 2.7

На головній сторінці я розмістив основні дані про кожен продукт, а саме:

- Назву продукту
- Вартість
- Зображення
- Кнопку перегляду товару на сторінці продукту
- Кнопку «add to cart»

Всі дані з продукту я розмістив на сторінці продукту, щоб головна сторінка інтернет магазину виглядала краще, та зайва інформація не заважала користувачу.

На рисунку 2.8 можна побачити який вигляд має сторінка продукту, на якій було розміщено всі дані про продукт, а також кнопка «add to cart»

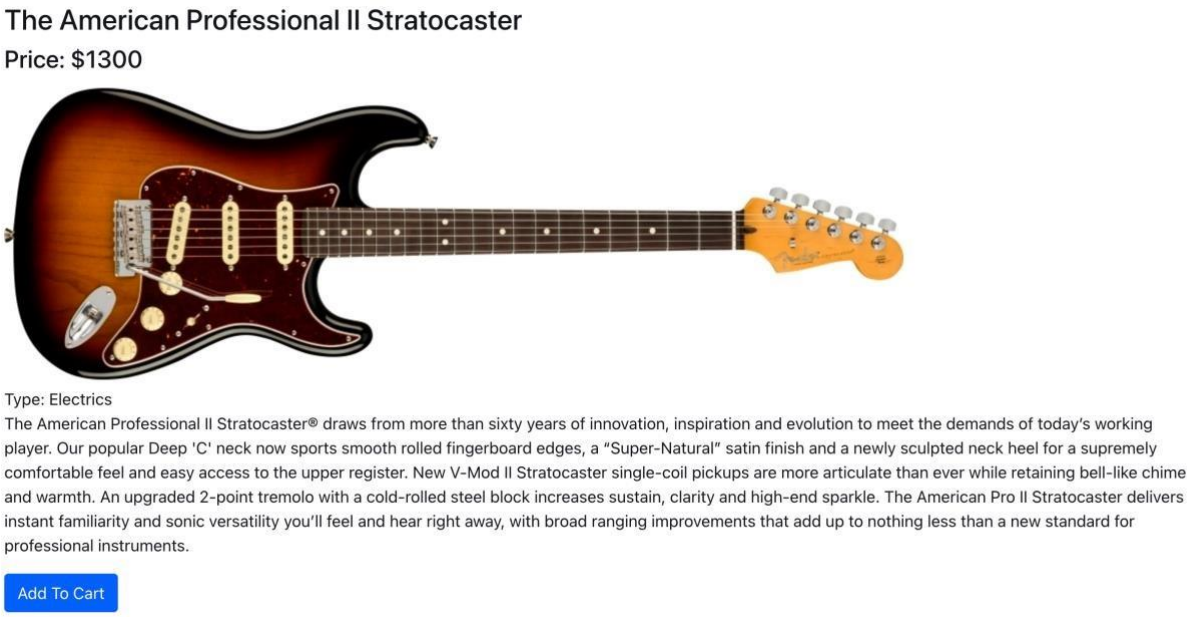

Рисунок 2.8

Використовуючи бібліотеку Redux було впроваджено функціонал корзини. В сховищі Local Storage зберігаються дані про користувача, що пройшов автентифікацію, а також вміст корзини. При натисканні “Add to cart” вміст корзини поповнюється вибраним продуктом, про що також свідчить лічильник продуктів в корзині, розміщений в хедері сторінки.

Рисунок 2.9

Після натисканні ‘Add to cart’ в масив продуктів cartItems, який знаходиться в Local Storage поповнюється обраним продуктом. Відповідно, ми можемо відобразити продукт, або перелік цих продуктів на сторінці корзини. На рисунку 2.10 зображено перелік продуктів на сторінці корзини.

Рисунок 2.10

Також на сторінці корзини ми можемо видалити обраний товар з корзини.

Оформлення замовлення відбувається наступним чином. Поряд з колекцією продуктів в Firestore database знаходиться колекція замовлень (рисунок 2.11).

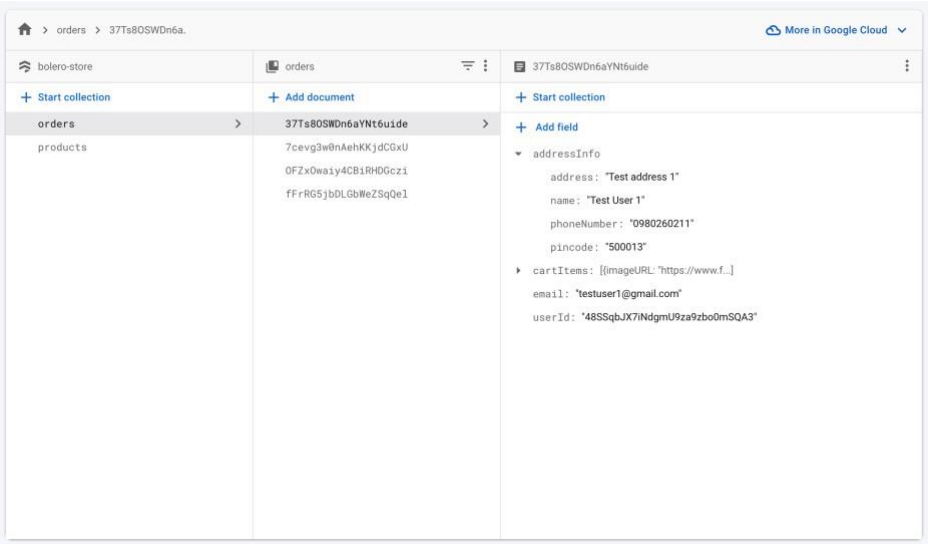

Рисунок 2.11

При натисканні “Place Order” з’являється модальне вікно, в якому необхідно ввести наступні дані:

- Ім’я
- Адресу
- Поштовий індекс
- Номер телефону

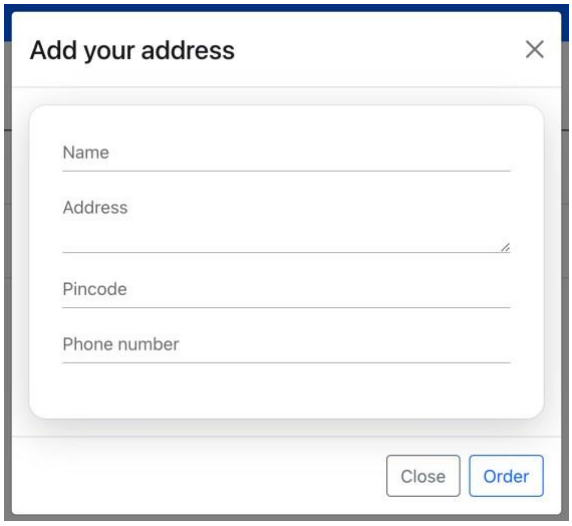

Рисунок 2.12

Після введення даних та натискання кнопки ‘Order’ створюєтья об’єкт orderInfo, який відправляється в базу даних за допомогою імпортованої функції addDoc()

```
const placeOrder = async () => {
    const addressInfo = {
        name,
        address,
        pincode,
        phoneNumber
    }

    const orderInfo = {
        cartItems,
        addressInfo,
        email: JSON.parse(localStorage.getItem('currentUser')).user.email,
        userId: JSON.parse(localStorage.getItem('currentUser')).user.uid
    }

    try {
        setLoading(true);
        const result = await addDoc(collection(fireDB, "orders"), orderInfo);
        setLoading(false);
        toast.success('Order placed successfully');
        handleClose();
    } catch (error) {
        setLoading(false);
        toast.error('Order failed');
    }

}
```

Рисунок 2.13

Далі я приступив до роботи над автентифікацією, а саме реєестрація та вхід користувача. Для цього було використано Firebase Authentication. Було створено сторінку логіну та реєстрації.

Для реєстрації користувача було використано функцію createUserWithEmailAndPassword(), для входу – signInWithEmailAndPassword().

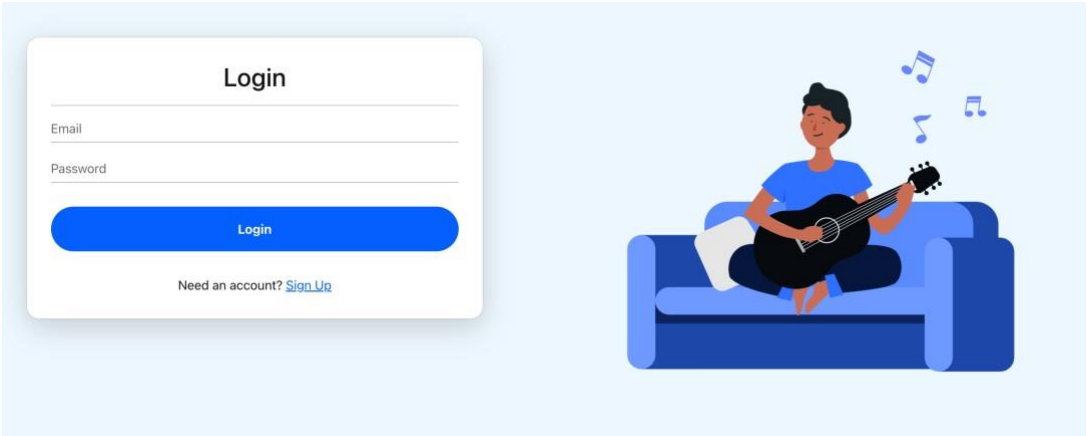

Рисунок 2.14

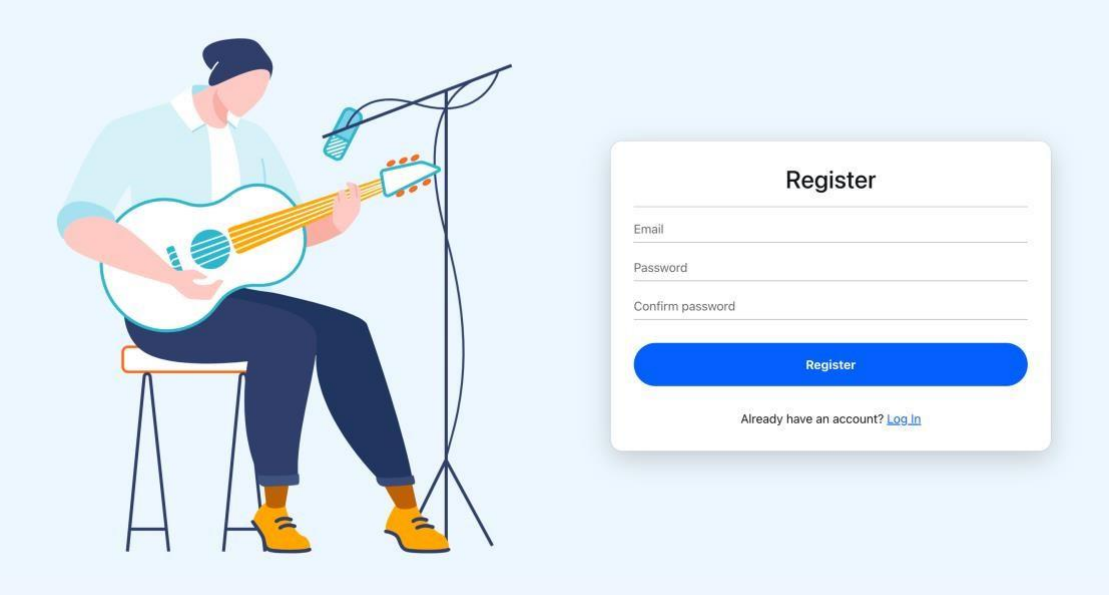

Рисунок 2.15

На головній сторінці я також зробив фільтр по назві та по категорії, використовуючи функцію filter().

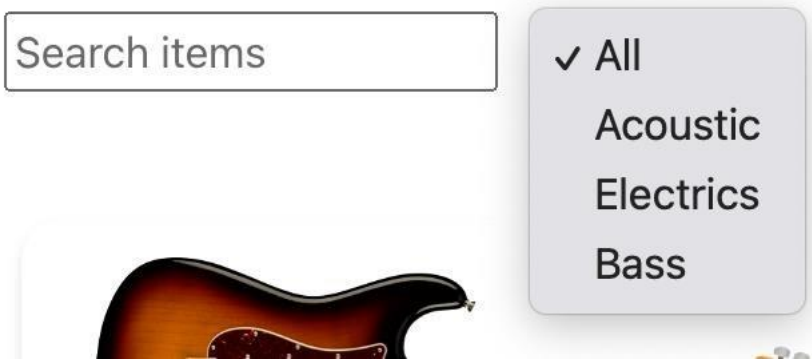

Рисунок 2.16

Заключим етапом роботи над інтернет магазином стала сторінка адміна. Сторінка адміна містить дві вкладки:

- Перелік всіх продуктів
- Перелік всіх замовлень

На рисунку 2.17 зображена перша вкладка, з переліком всіх продуктів та їх основними даними. Тут були реалізовані наступні функції:

- Видалення продукту з бази даних магазину
- Редагування даних про продукт
- Можливість додати новий продукт

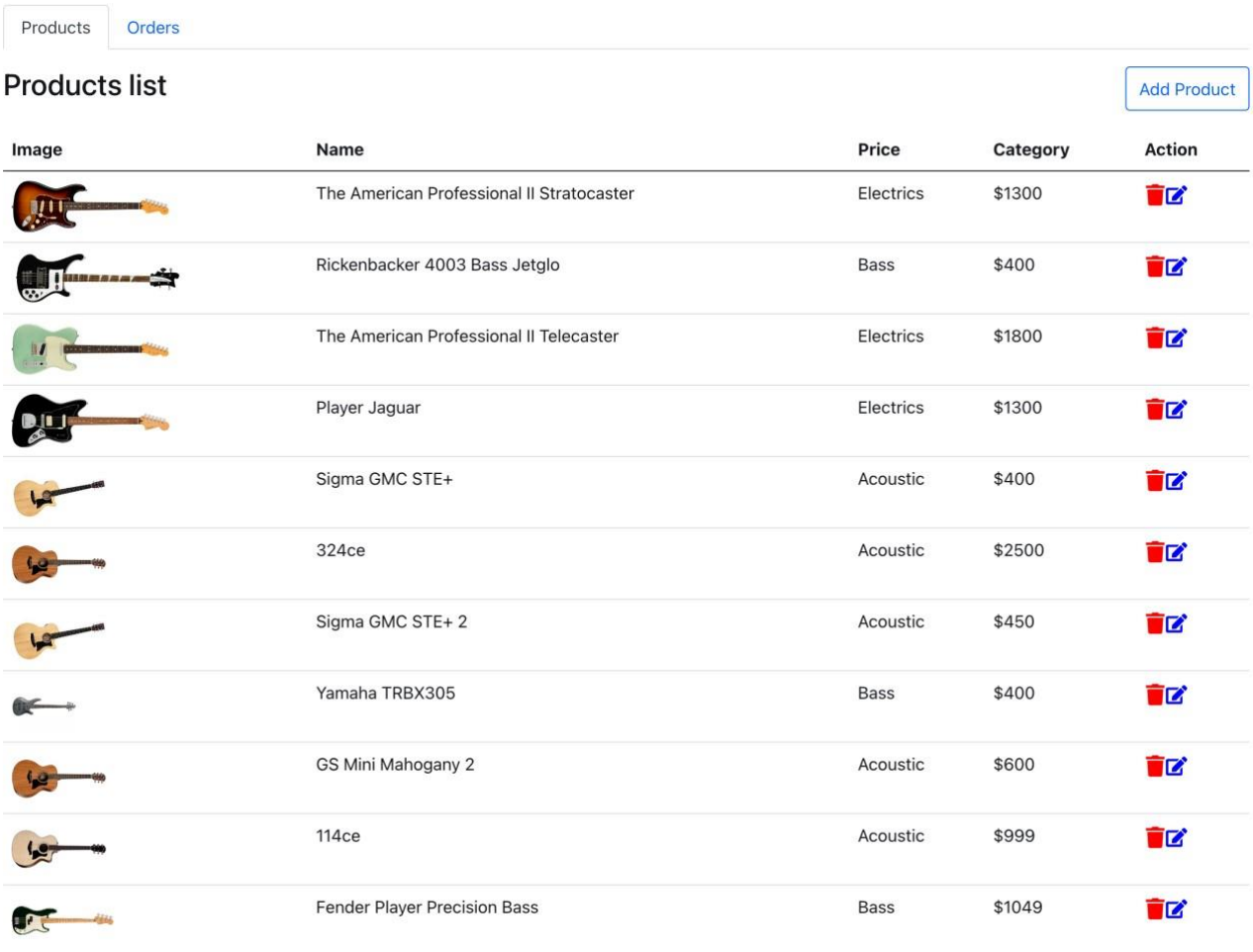

Рисунок 2.17

На рисунку 2.18 зображено перелік всіх замовлень у вигляді двох таблиць. В першій таблиці зображені дані користувача, який зробив замовлення. В другій таблиці перелік продуктів, які замовив користувач.

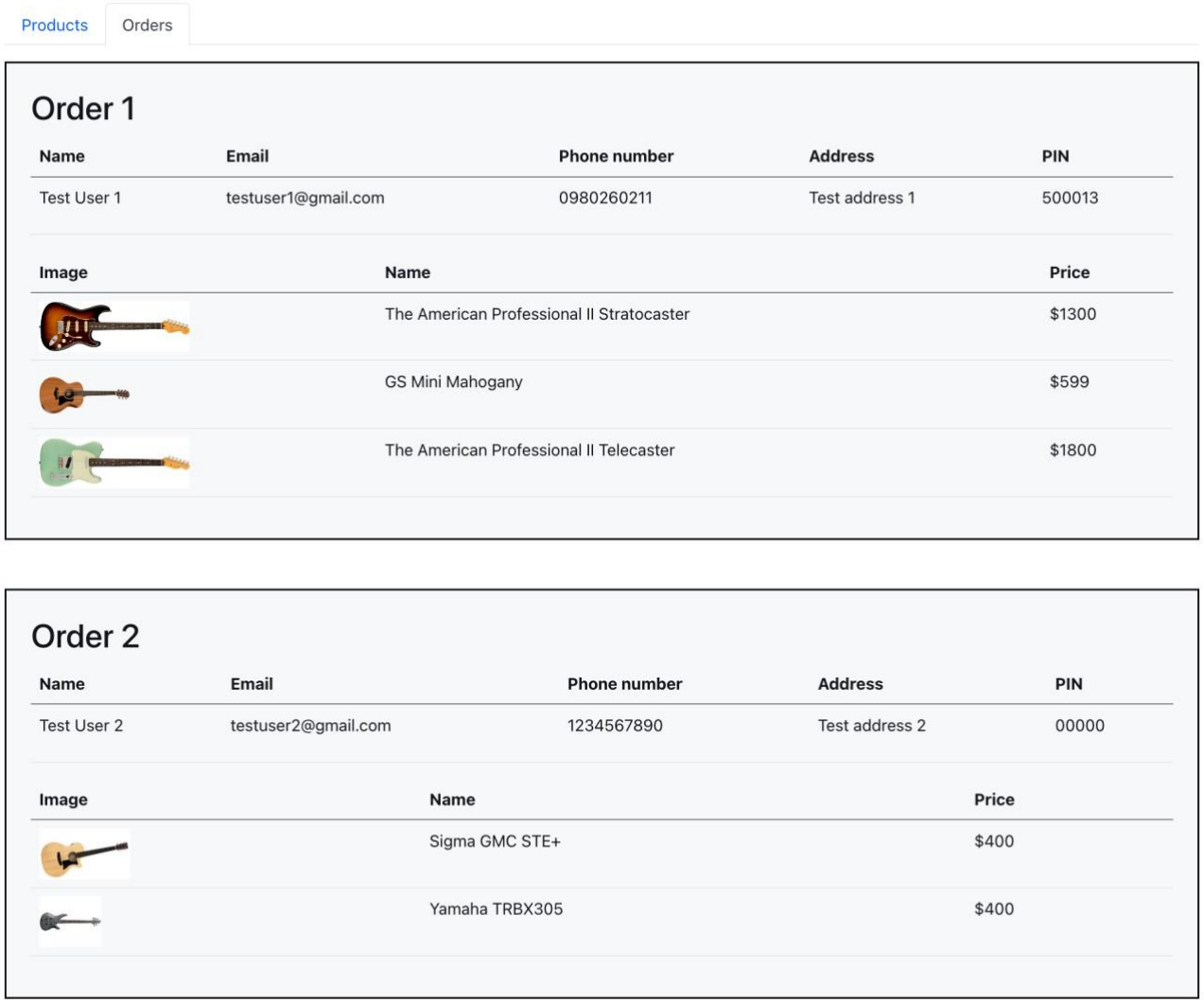

Рисунок 2.18

ВИСНОВКИ

У даній дипломній роботі було розглянуто процес розробки інтернетмагазину музичних інструментів з використанням бібліотеки React, бібліотеки Redux та сервісу Firebase. Під час розробки були вирішені такі завдання, як планування структури проекту, налаштування середовища розробки, проектування та розробка інтерфейсу користувача, розробка системи управління станом додатку, налаштування та інтеграція з Firebase для зберігання даних та аутентифікації користувачів.

За допомогою React було створено ефективний та зручний для розробки інтерфейс користувача. Redux був використаний для керування станом додатку та управління його поведінкою. Firebase дозволив зберігати дані та робити аутентифікацію користувачів, що значно спростило процес розробки та роботи з базою даних.

Отже, можна стверджувати, що використання React, Redux та Firebase дало можливість створити ефективний та зручний для користування інтернет-магазин музичних інструментів зі зручною системою управління станом та безпечним зберіганням даних. Ці технології можуть бути використані в подальших проектах з розробки веб-додатків та інтернет-магазинів, що потребують високої продуктивності та зручності користування.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Wojciech Baranowski. React JS: Advantages and Dosadvantages. Режим доступу: https://massivepixel.io/blog/react-advantages-disadvantages/
2. Advantages and Disadvantages of ReactJS. Режим доступу: https://www.w3schools.blog/advantages-disadvantages-reactjs
3. What is Angular? Режим доступу: https://angular.io/guide/what-is-angular
4. Advantages and Disadvantages of Angular. Режим доступу: https://www.knowledgehut.com/blog/web-development/advantages-anddisadvantages-of-angular
5. Angular vs React: Difference Between Angular and React. Режим доступу: https://www.interviewbit.com/blog/angular-vs-react/
6. Edpresso Team. What is Firebase? Режим доступу: https://www.educative.io/edpresso/what-is-firebase
7. Bogdan Guriev. Firebase Pros and Cons: When You Should and Shouldn’t Use Firebase | OSDB. Режим доступу: https://osdb.io/firebase-pros-and-cons-whenyou-should-and-shouldnt-use-firebase-osdb/
8. Simplilearn. What is AWS(Amazon Web Services): Services, Applications, Advantages and More. Режим доступу: https://www.simplilearn.com/tutorials/aws-tutorial/what-is-aws
9. Saurabh Barot. Firebase vs AWS: Which One to Choose in 2022. Режим доступу: https://aglowiditsolutions.com/blog/firebase-vs-aws/
10. Cloud Firestore. Режим доступу: https://firebase.google.com/docs/firestore

11. Pros and Cons of JavaScript – Weigh them and Choose wisely! Режим доступу: https://data-flair.training/blogs/advantages-disadvantages-javascript/
12. Использование с React. Режим доступу: https://rajdee.gitbooks.io/redux-inrussian/content/docs/basics/UsageWithReact.html