**Зміст:**

1. **Опис завдання, його мета і тема – ст.**
2. **План та шаблони для проекту – ст.**
3. **Технічне забезпечення – ст.**
4. **Використані технології – ст.**
5. **Опис принципу роботи і приклади із наведеними скріншотами – ст.**
6. **Висновок – ст.**
7. **Використана література – ст.**
8. **Архів з додатком проекту та посилання на проект на GitHub – ст.**
9. **Опис завдання, його мета і тема:**
   1. **Тема:**

Створення веб-додатку для підприємства, що займається виробництвом ювелірних виробів. Сайт повинен надавати клієнтам необхідну інформацію про товар, його якість і спосіб його виробництва, надаючи можливість надсилати замовлення продукції підприємства, одержувати інформацію про вакансії та контактні дані керівництва підприємства.

* 1. **Мета:**

У результаті виконання даної роботи необхідно отримати працюючий веб-сайт, який дозволить користувачам знаходити інформацію про підприємство, залишати контактні дані для зв’язку та співробітництва, розглядати вакансії та дивитися продукцію підприємства.

* 1. **Опис Завдання:**

Структура сайту повинна містити наступні змістовні часини:

–можливість введення пошукового запиту та забезпечення його виконання;

–можливість входу в систему різним користувачам: вхід для зареєстрованих користувачів, реєстрація для нового користувача, можливість отримання нового паролю на e-mail адресу на випадок забутого;

–наявність меню навігації по стандартним пунктам системи. Якщо користувач має права адміністратора, йому доступні лінки на адміністративні сторінки управління сайтом та налаштування профілю користувача. Якщо користувач не має особливих привілеїв в системі, він отримає лише лінки на налаштування профілю та вихід із системи;

–наявність меню, яке буде наповнюватися лінками та кнопками переходу до основних сторінок сайту;

–відображення кількості користувачів, які зараз знаходяться на сайті.

1. **План та шаблони для проекту:**
   1. **План:**

Створити працюючий додаток на vue3 на декілька сторінок, кожна з яких буде наповнюватись окремим контентом. Тема – магазин ювелірних виробів, тому структура сайту має такий вигляд:

***Головна сторінка*** – вона завантажується, коли користувач заходить на сайт. Наповнення контентом – мінімальне, тільки привітальне сповіщення.

***Каталог*** – завантажується і відображає всі товари, які є у базі даних.

***Кошик*** – завантажується і відображає всі товари, які користувач додав у кошик.

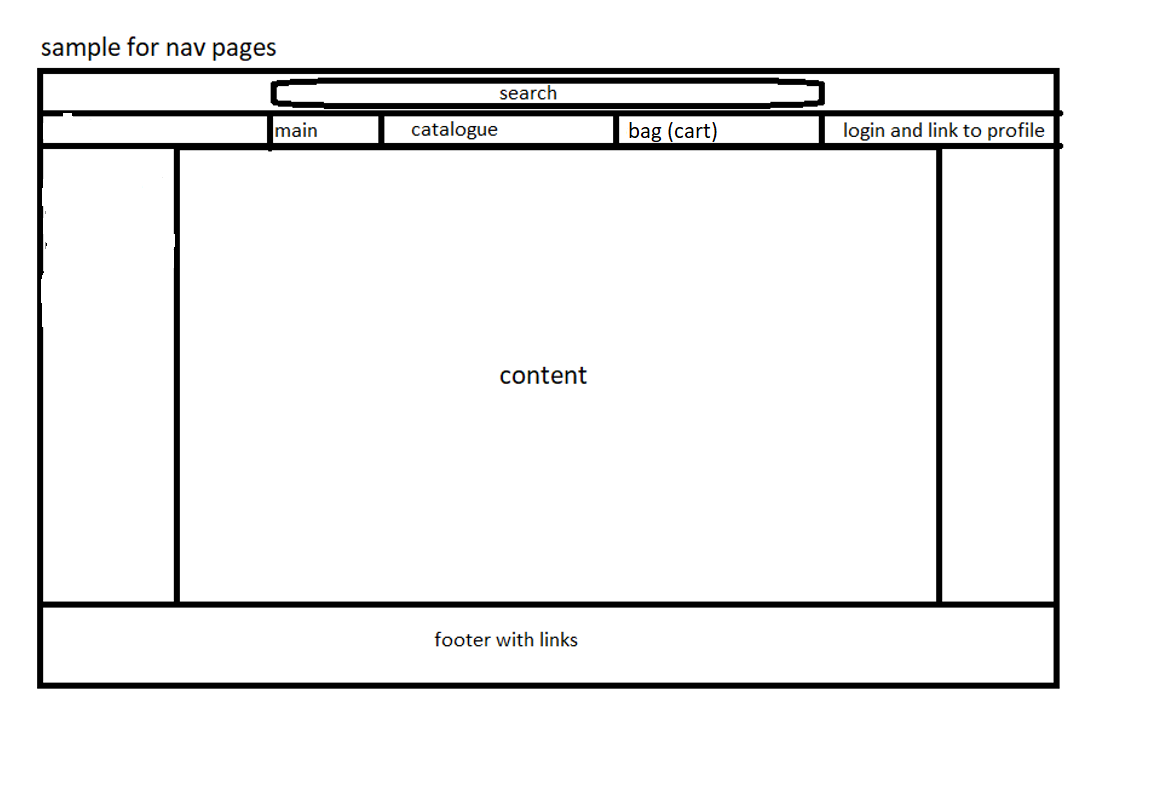
***Профіль/Регістрація/Логін*** – дає користувачу можливість перейти до свого профілю, якщо він авторизований, або зареєструватись та авторизуватись.

***Панель адміна*** – відображує кількість користувачів на сайті.

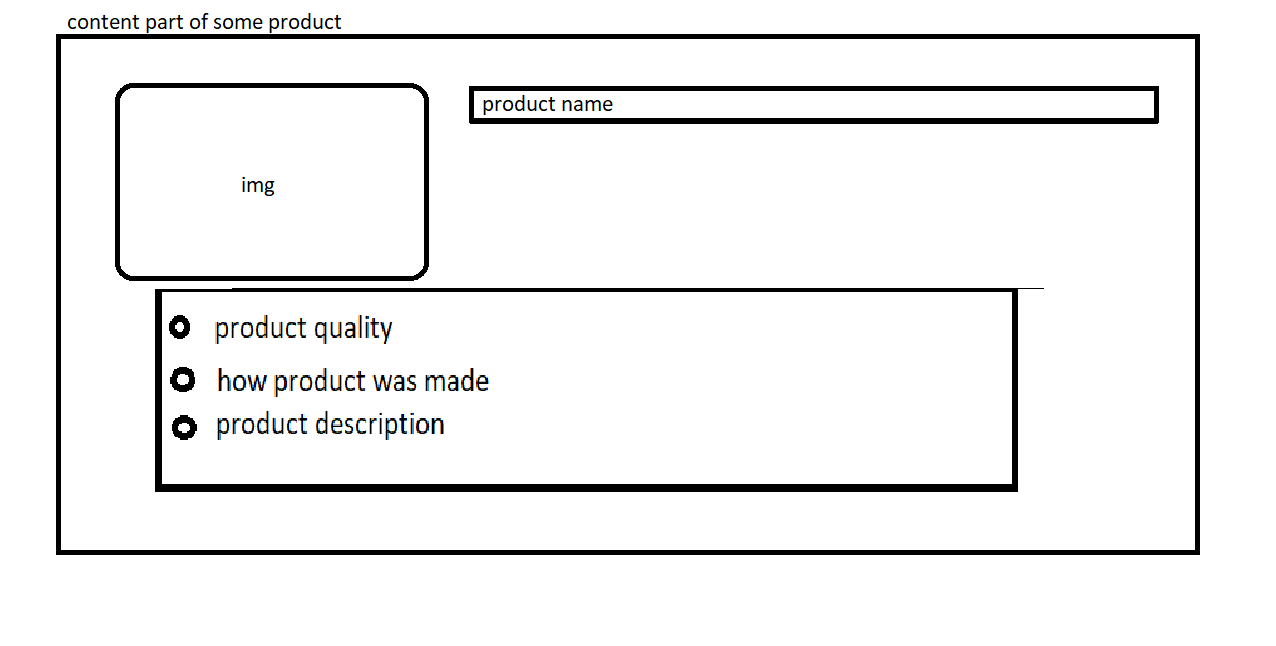
***Про нас*** – лінка на пошту команді розробки сайту (моя пошта).

* 1. **Шаблони:**

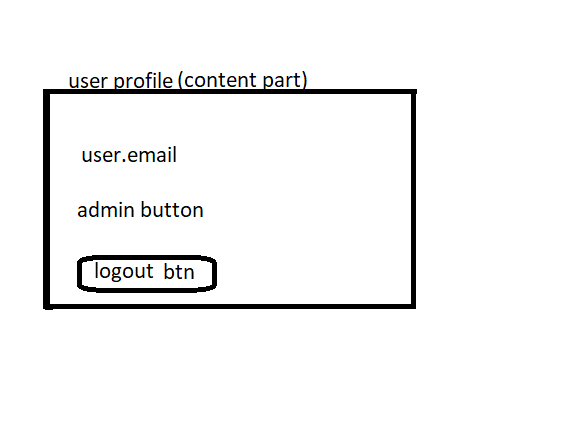
Ці шаблони – ранні заготовки, які змінювались по ходу реалізації їх у проекті. (Додалися і нові сторінки та компоненті, але шаблони були виконані не для всіх сторінок)



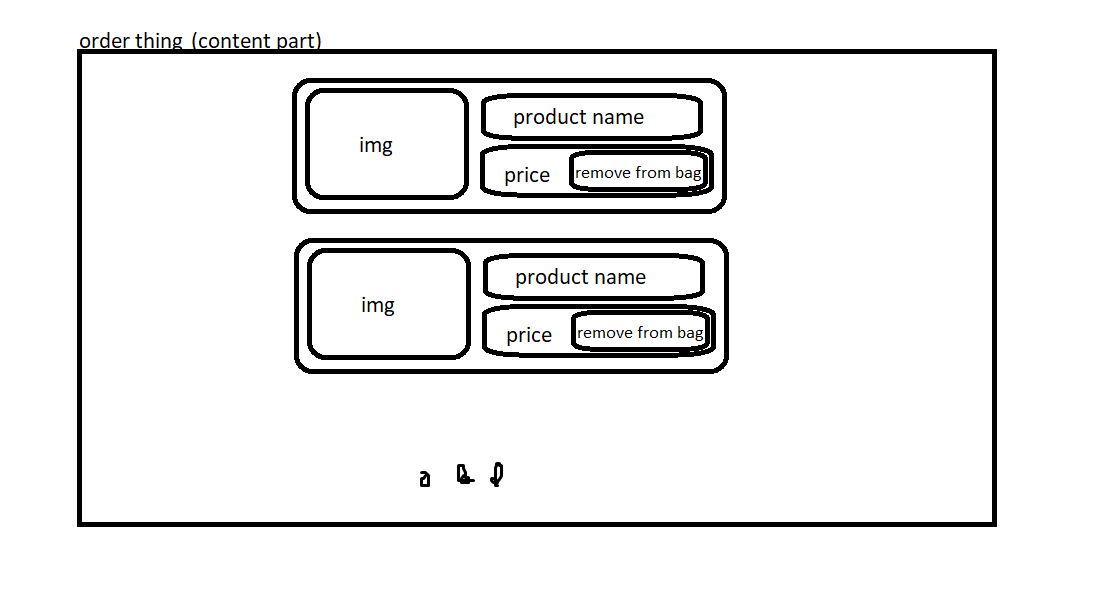
(рис.1 – шаблон для меню навігації)



(рис.2 – шаблон для сторінки продукта).



(рис.3 – шаблон для сторінки користувача)



(рис.4 – шаблон сторінки кошика користувача)

1. **Технічне забезпечення:**

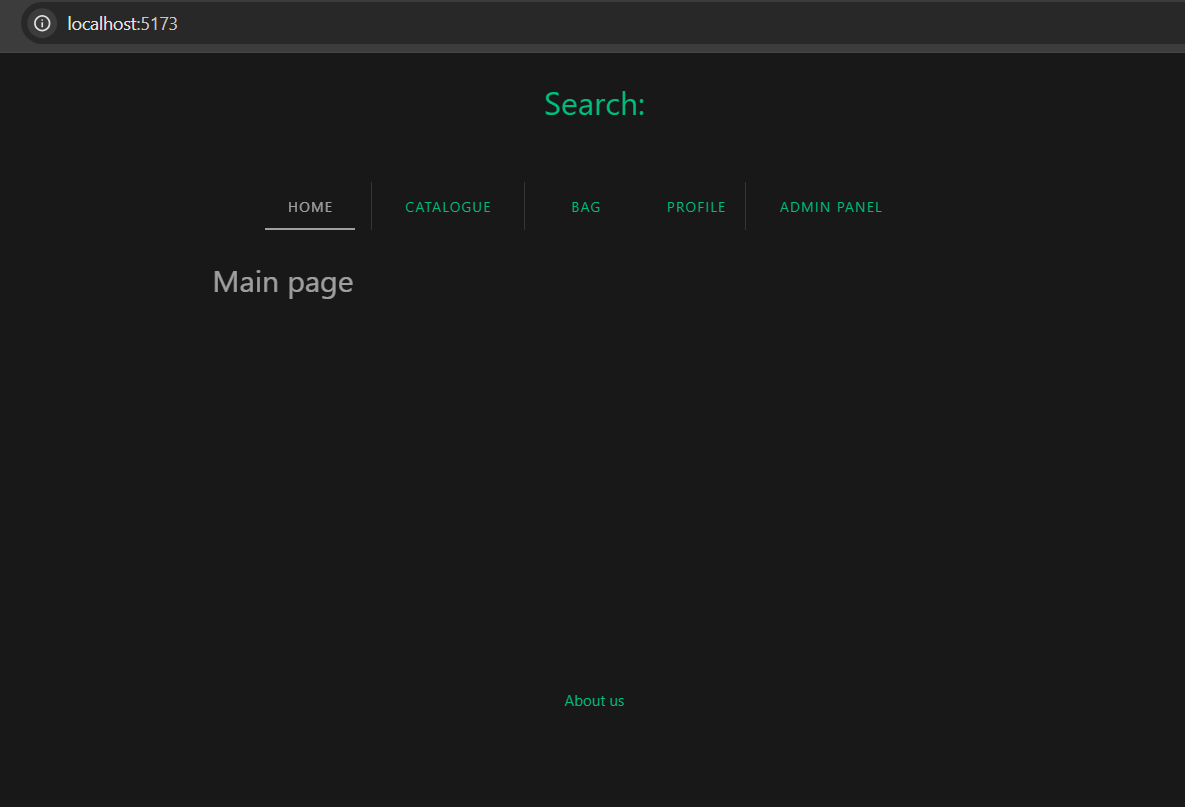
Рекомендовані вимоги до характеристик компонентів технічного забезпечення, при яких значення тимчасових параметрів Системи повинні відповідати наступним вимогам:

Для персонального комп’ютера користувача:

* Процесор IntelPentium 1.5GGh;
* Обсяг оперативної пам’яті 256Mb;
* Дискова підсистема 40Gb;
* Мережевий адаптер 100Мб/с.

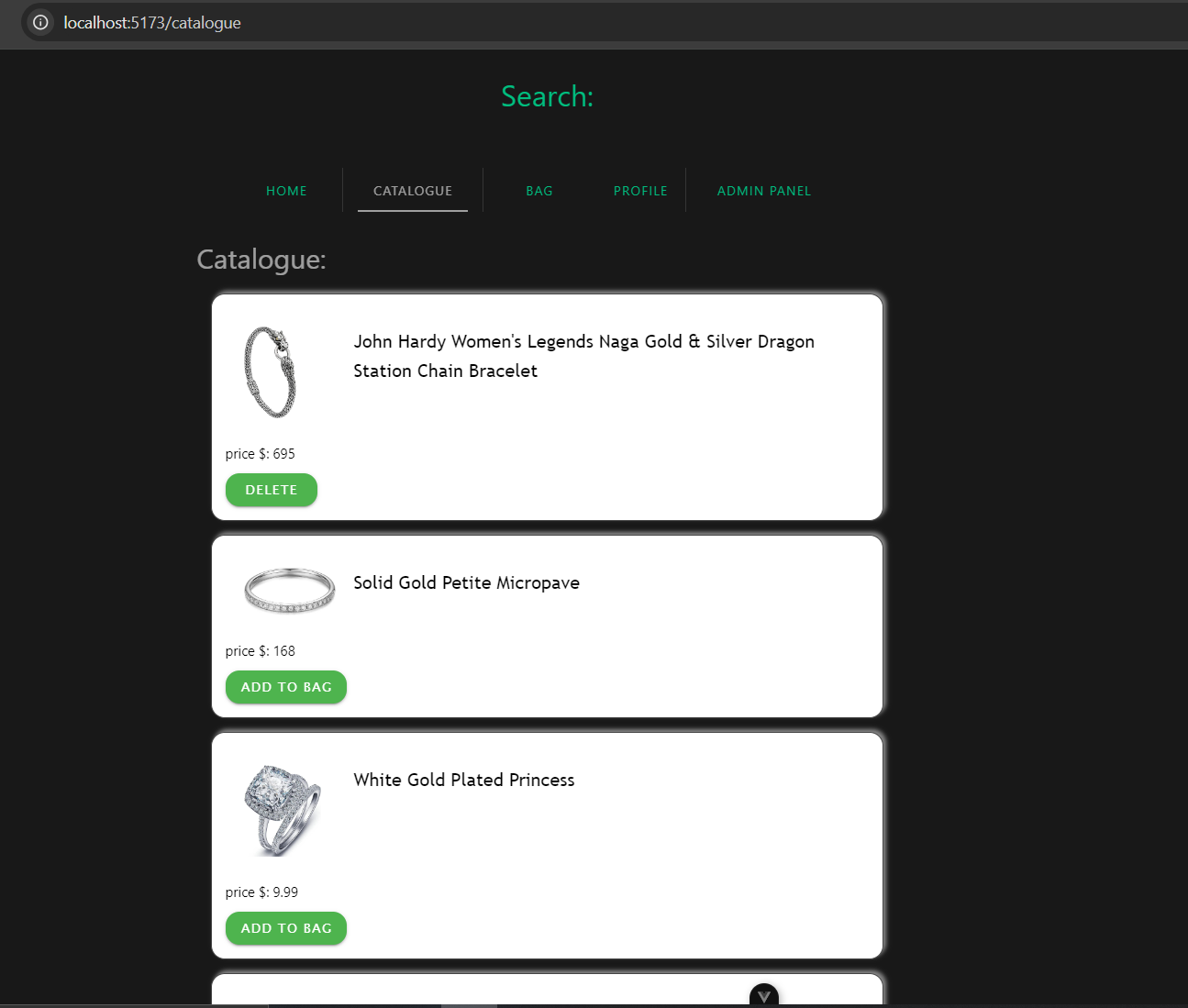
1. **Використані технології:**
   1. **Back-end:**
      1. Серверна частина працює на базі google firebase. Це безкоштовна (у деякій мірі) база, яка надає доступ до можливості зареєструвати користувача, зберегти його дані та авторизувати його. Набір можливостей у безкоштовній версії обмежено, тому функції адміна довелось реалізувати у тестовому режимі.
      2. Список товарів відображається за допомогою fakestoreapi. Це безкоштовний у деякій мірі сервіс, який дозволяє знайти товар з картинкою, описом товару, його ціною та id.
   2. **Front-end:**
2. Vue3js (html, css, js (composition API))
3. Vuetify (UI-framework для vue3)
4. Axios – додаток для vue3 для більш зручної обробки даних з сервера.
5. Vue-router – додаток для vue3 для зручної розробки сайту з кількома сторінками.
6. **Опис принципу роботи і приклади зі скрінами:**

При запуску сайту користувач попадає на головну сторінку:



(рис.5 – головна сторінка)

При переході у каталог, сторінка завантажує список продуктів із сервера і при готовності виводить цей список користувачу:

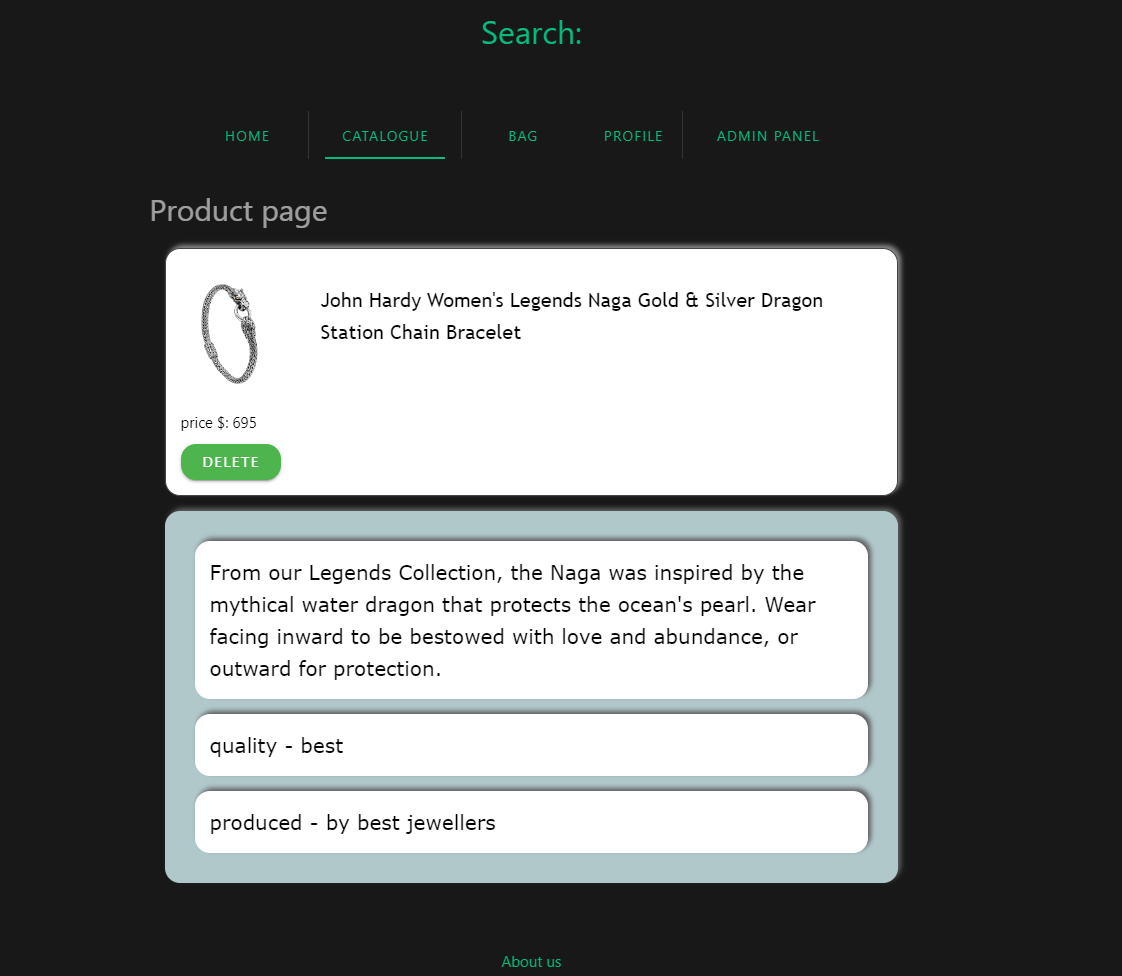


(рис.6 – каталог)

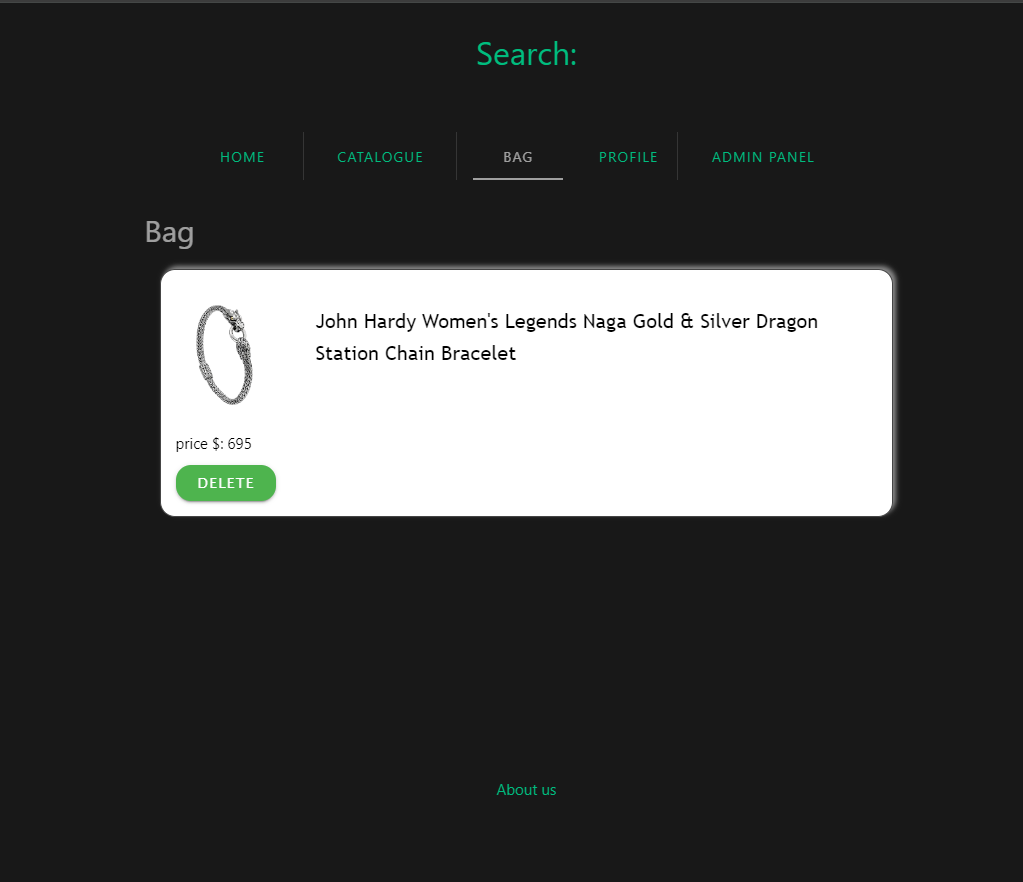
У користувача в свою чергу є можливості:

-Додати товар у кошик, або видалити з кошика (якщо це товар знаходиться у кошику)

-Клікнути на товар щоб перейти на сторінку цього товару:

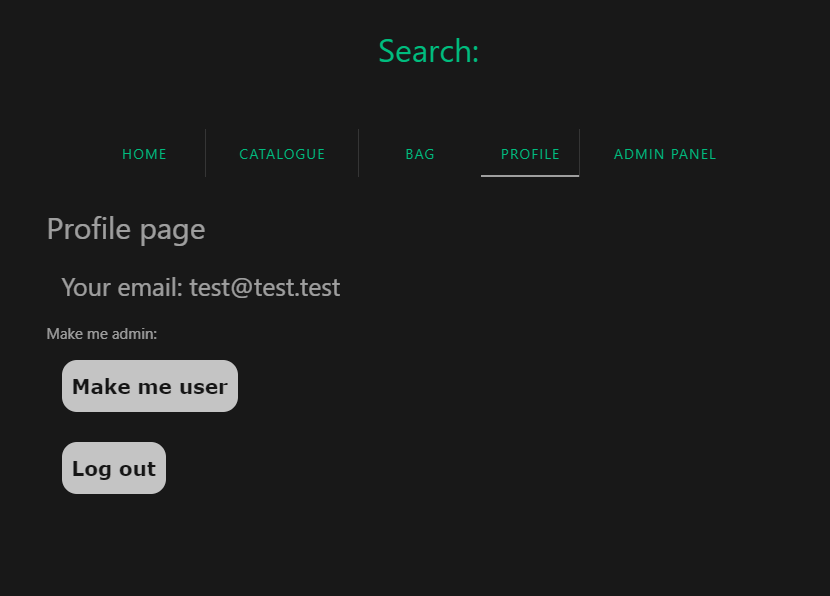


(рис.7 – сторінка товару)



(рис.8 – сторінка кошику).

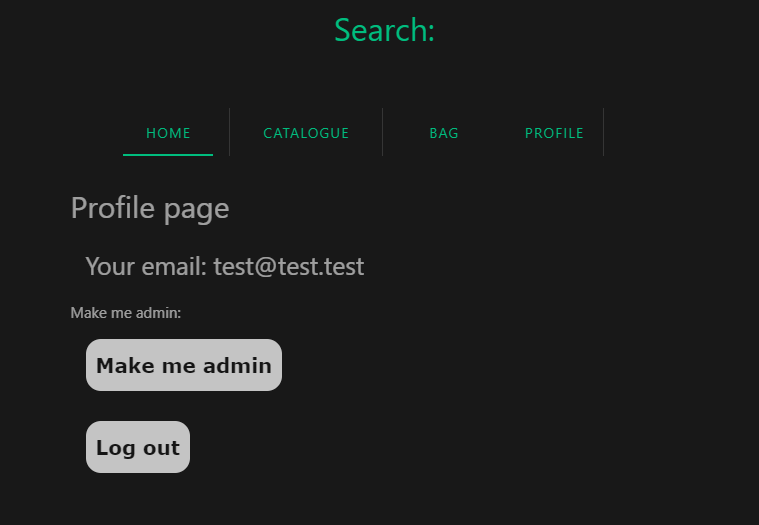
Кошик зберігається у local storage браузеру користувача і виводить товар без запиту на сервер.



(рис.9 – сторінка користувача з правами адміністратора)

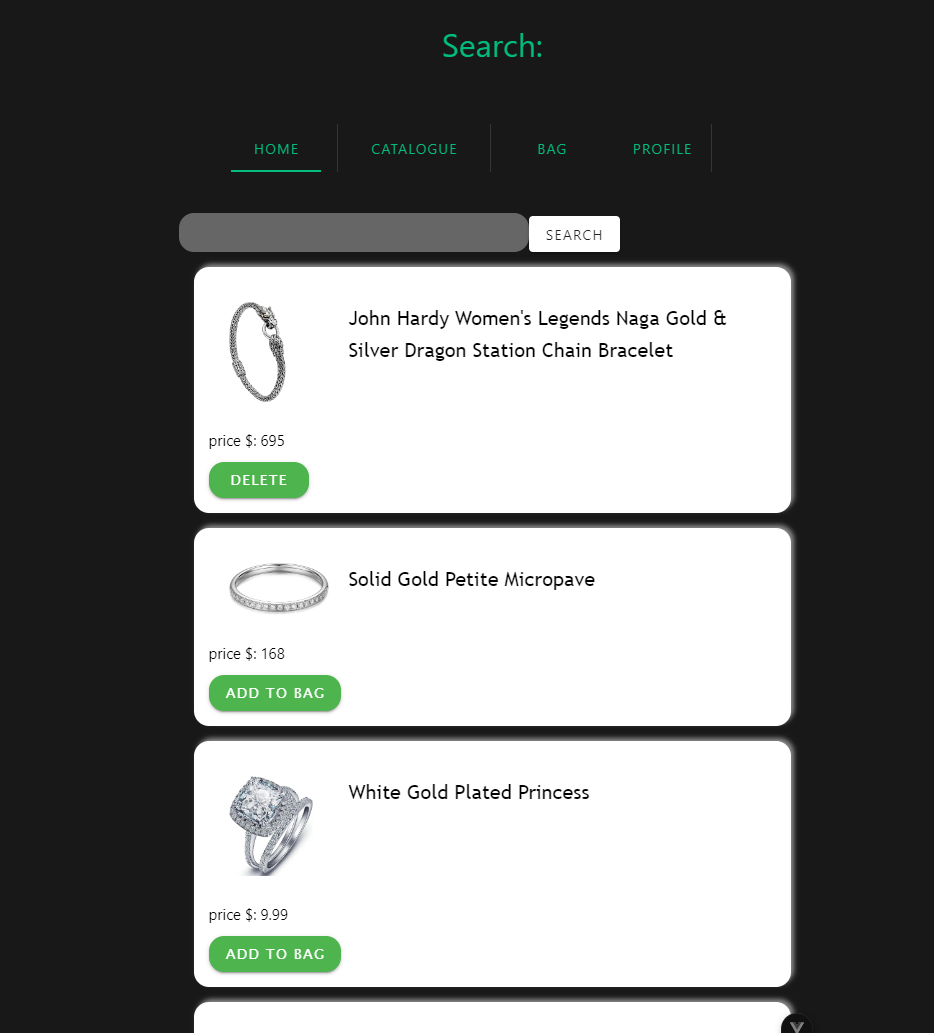
У користувача є можливість вийти з облікового запису, та зробити його адміном/звичайним користувачем. З початку так заплановано не було, але відсутність можливості безкоштовно надавати ролі користувачам firebase примусила прийти до такого рішення з правами адміна, адже навіть така тестова реалізація відповідає умові завдання:

–наявність меню навігації по стандартним пунктам системи. Якщо користувач має права адміністратора, йому доступні лінки на адміністративні сторінки управління сайтом та налаштування профілю користувача.



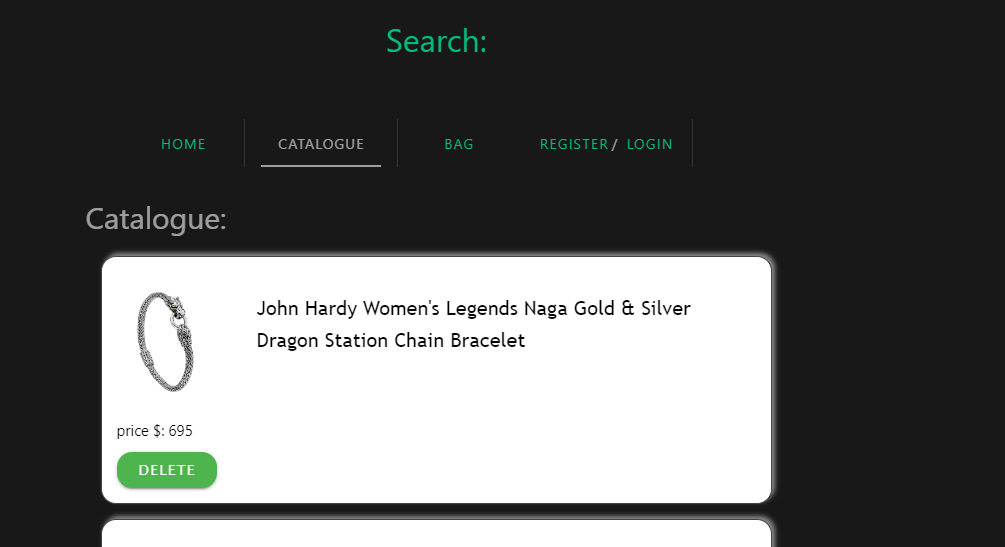
(рис.10 – сторінка користувача без прав адміністратора)

Без прав адміністратора у користувача немає доступу до адмін панелі.

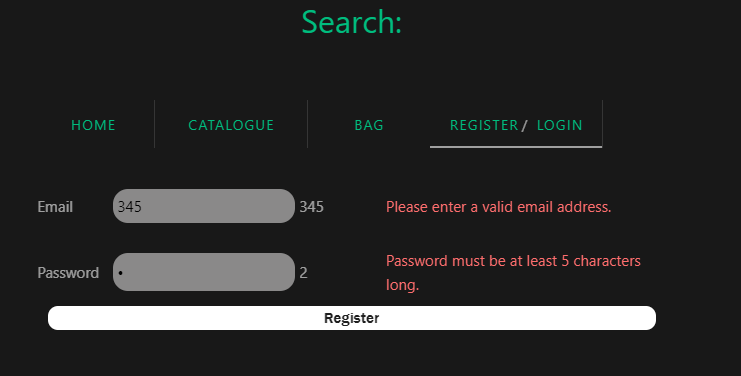


(рис.11 – панель пошуку)

При натисканні Search над панеллю навігації користувача перекидає на сторінку пошуку із вбудованим каталогом. При введені пошукового запросу та натисканні кнопки пошуку, користувачеві відображаються всі товари в назві яких є запрос користувача. Враховано використання символів обох регістрів (великі/маленькі) та введення зайвих пробілів на кінці та початку запросу.

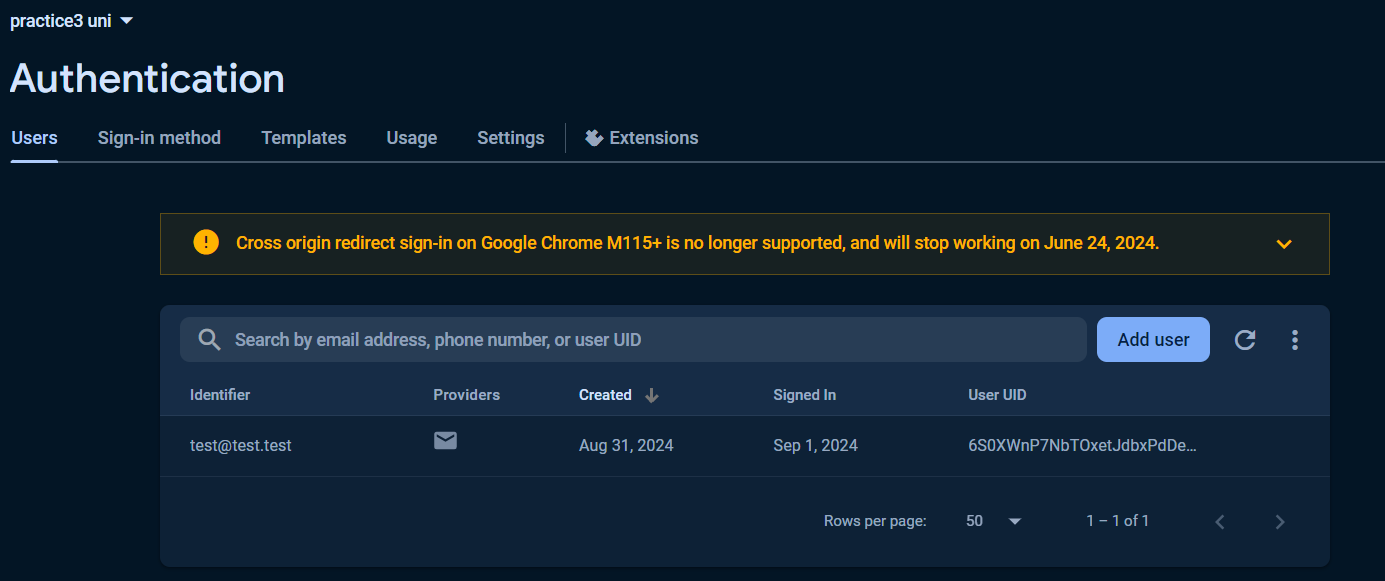


(рис.12 – сторінка, яку бачить неавторизований користувач)

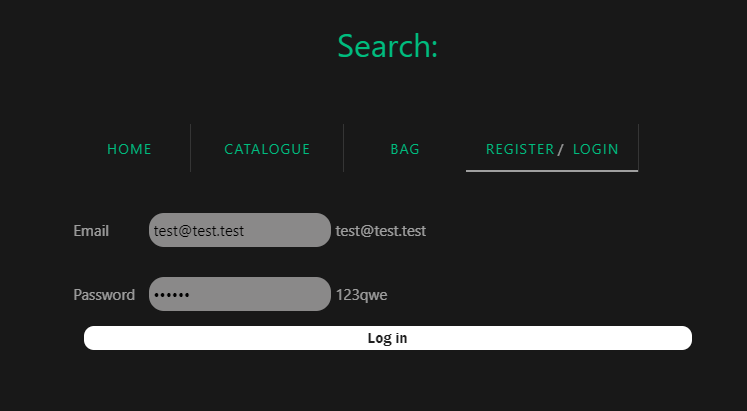


(рис.13 – сторінка реєстрації)

При реєстрації є валідація форми, відображення введених даних сбоку від форми. При помилці у введені відобразиться помилка, яка надасть користувачу про формат вводу пошти та паролю. При відсутності помилок користувача зареєструє у firebase:

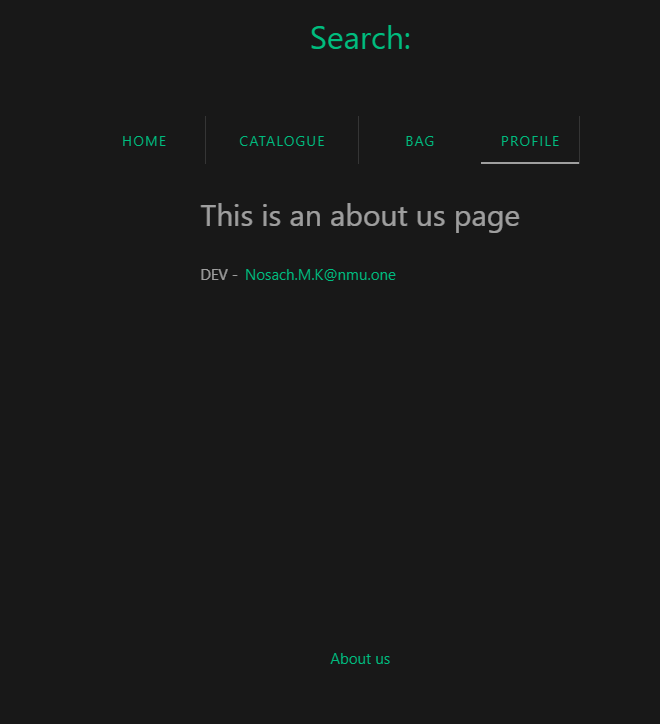


(рис.14 – зареєстрований користувач)



(рис.15 – сторінка логіну)

Така сама форма як при реєстрації, але вона на реєструє користувача, а відправляє запит на сервер підтверджуючи дані користувача.



(рис.16 – сторінка «про нас»)

Сторінка «про нас» відображає пошту і вакансію розробника додатку.

1. **Висновок:**

У результаті виконання даної роботи отримано працюючий веб-сайт, який дозволяє користувачам знаходити інформацію про підприємство, залишати контактні дані для зв’язку та співробітництва, розглядати вакансії та дивитися продукцію підприємства.

1. **Використана література:**
2. **Архів з додатком проекту та посилання на проект на GitHub:**