

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Фронт-энд разработка

Отчет

Лабораторная работа №2

Выполнил:
Тишалович Л.М.

Группа
K33392

Проверил:
Добряков Д. И.

Санкт-Петербург

2023 г.

Задача

Магазин одежды: Разработка интернет-магазина одежды, включающего функции добавления товаров в корзину, оформления заказа и фильтрации по категориям.

Ход работы

В ходе данной лабораторной работы был создан JSON-сервер.

```
{
  "users": [],
  "cart": [],
  "clothes": [
    {
      "id": 0,
      "title": "Black SweatShort",
      "img": "/src/assets/itemsCovers/blackSweatshort.svg",
      "price": 123,
      "brand": "Jhanvi's Brand",
      "gender": "female"
    },
    {
      "id": 1,
      "title": "Black Hoodie",
      "img": "/src/assets/itemsCovers/blackHoodie.svg",
      "price": 37,
      "brand": "As Brand",
      "gender": "female"
    },
    {
      "id": 2,
      "title": "Black Shorts",
      "img": "/src/assets/itemsCovers/blackShorts.svg",
      "price": 37,
      "brand": "MM Brand",
      "gender": "female"
    },
    {
      "id": 3,
      "title": "White T-Shirt",
      "img": "/src/assets/itemsCovers/whiteTshirt.svg",
      "price": 30,
      "brand": "As Brand",
      "gender": "male"
    }
  ],
}
```

Было реализовано получение карточек товаров при обращении к серверу.

```
useEffect(() => {
  axios
    .get("http://localhost:3000/clothes")
    .then((response) => {
      console.log(response.data);
      setData(response.data);
    })
    .catch((error) => {
      console.error("There was an error fetching data", error);
    });
}, []);
```

Была реализована фильтрация пользователей по гендеру.

```
useEffect(() => {
  let url = "http://localhost:3000/clothes";

  const params = new URLSearchParams();

  if (selectedGender !== "all") {
    params.append("gender", selectedGender);
  }

  if (searchQuery.trim() !== "") {
    params.append("q", searchQuery.trim());
  }

  if (params.toString()) {
    url += `?${params.toString()}`;
  }

  axios
    .get(url)
    .then((response) => {
      setData(response.data);
    })
    .catch((error) => {
      console.error("There was an error fetching data", error);
    });
}, [selectedGender, searchQuery]);
```

```
<Header
  setSelectedGender={setSelectedGender}
  cartItems={cartItems}
  onSearch={setSearchQuery}
/>
```

```
const handleGenderClick = (gender: string) => {
  setSelectedGender(gender);
};
```

Были реализованы регистрация и вход уже зарегистрированных пользователей.

```
const handleSignIn = () => {
  const isEmail = /\S+@\S+\.\S+/.test(emailOrUsername);
  if (!isEmail) {
    alert("Please enter a valid email address for registration.");
    return;
  }

  axios
    .get("http://localhost:3000/users", {
      params: { email: emailOrUsername },
    })
    .then((response) => {
      if (response.data.length > 0) {
        alert("This email is already registered.");
        return;
      }
      axios
        .post("http://localhost:3000/users", {
          email: emailOrUsername,
          password,
        })
        .then(() => {
          alert("Registration successful!");
          navigate("/main");
        })
        .catch((error) => {
          console.error("Registration error:", error);
        });
    })
    .catch((error) => {
      console.error("Error checking registration:", error);
    });
};
```

```
const handleLogIn = () => {
  axios
    .get("http://localhost:3000/users", {
      params: { email: emailOrUsername, password },
    })
    .then((response) => {
      if (response.data.length === 0) {
        alert("Incorrect email or password.");
        return;
      }
      alert("Login successful!");
      navigate("/main");
    })
    .catch((error) => {
      console.error("Login error:", error);
    });
};
```

Была реализована функция добавления товара в корзину

```

useEffect(() => {
  axios
    .get("http://localhost:3000/cart")
    .then((response) => {
      setCartItems(response.data);
    })
    .catch((error) => {
      console.error("There was an error fetching cart items", error);
    });
}, [cartItems]);

const addToCart = (item: ItemData) => {
  axios
    .post("http://localhost:3000/cart", item)
    .then(() => {
      alert("Item is added to cart");
    })
    .catch((error) => {
      console.error("There was an error adding the item to the cart!", error);
    });
};

```

```

import style from "./CartPopup.module.sass";

type CartItem = {
  title: string;
  price: number;
  img: string;
};

type CartPopupProps = {
  items: CartItem[];
};

export const CartPopup = ({ items }: CartPopupProps) => {
  return (
    <div className={style.wrapper}>
      {items.length > 0 ? (
        items.map((item, index) => (
          <div key={index} className={style.item}>
            <p>{item.title}</p>
            <p>${item.price}</p>
            <img src={item.img} />
          </div>
        ))
      ) : (
        <p>Your cart is empty.</p>
      )}
    </div>
  );
};

```

Вывод: были получены навыки создания своего собственного JSON-сервера и работы с API