

**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);
    });
};

```

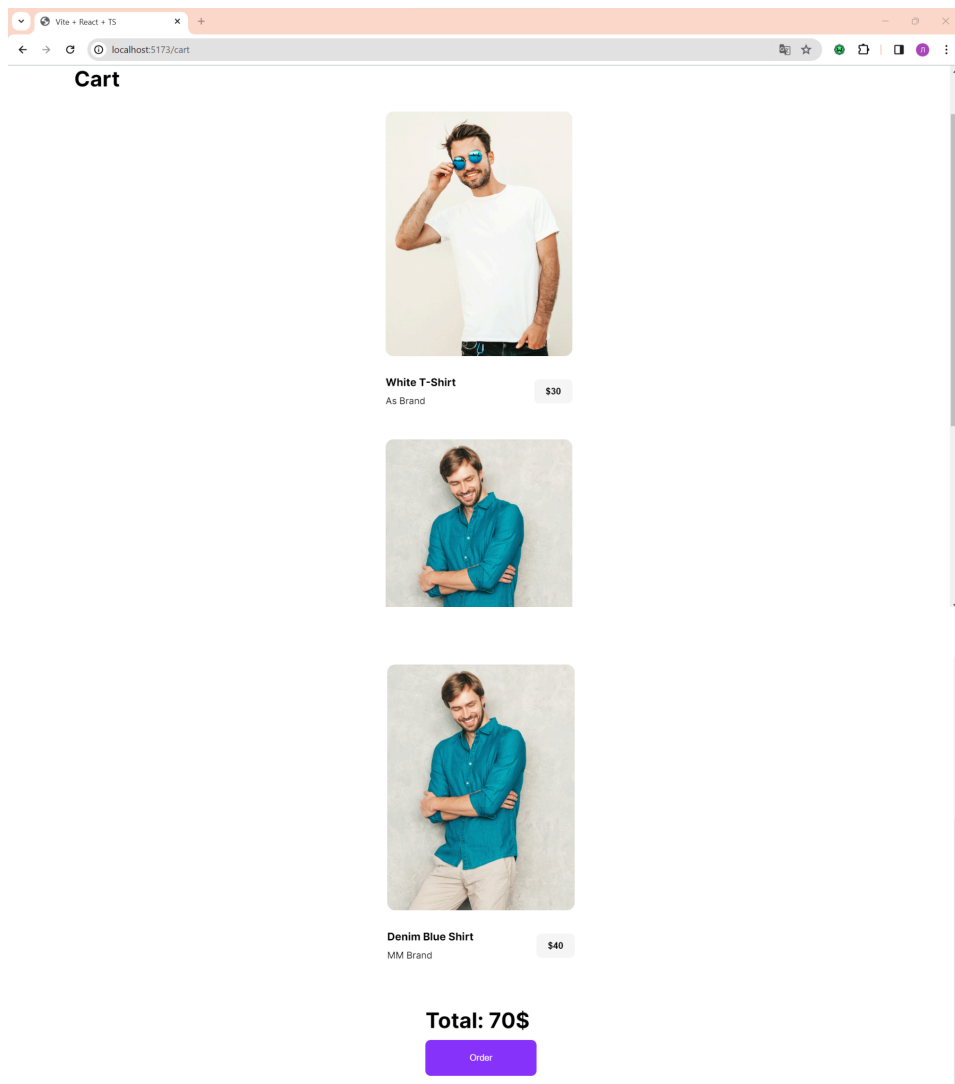
Была реализована функция добавления товара в корзину и страница корзины с последующим оформлением заказа

```
const addToCartDispatch = (item: ItemData) => {
  dispatch(
    addToCart({
      title: item.title,
      price: item.price,
      img: item.img,
      brand: item.brand,
      gender: item.gender,
      id: item.id,
    })
  );
};
```

```
return (
  <div className={style.cartWrapper}>
    <h1 className={style.cartTitle}>Cart</h1>
    <div className={style.mainContent}>
      <div className={style.itemSelection}>
        {cartItems.map((item: ItemData) => {
          const props = {
            id: item.id,
            title: item.title,
            img: item.img,
            price: item.price,
            brand: item.brand,
          };
          return <Item {...props} />;
        })}
      </div>
      <div className={style.total}>
        <h1>Total: {totalAmount}$</h1>
        <button onClick={handleOrder}>Order</button>
      </div>
    </div>
  </div>
);
```

```
const totalAmount = cartItems.reduce((total, item) => total + item.price, 0);
```

```
const handleOrder = async () => {  
  const orderData = {  
    items: cartItems.map((item) => item.title),  
    totalAmount,  
  };  
  try {  
    await axios.post("http://localhost:3000/orders", orderData);  
    alert("Order placed successfully:");  
  } catch (error) {  
    console.error("Error placing order:", error);  
  }  
};
```



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