

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ  
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

**Дисциплина:** Фронт-энд разработка

Отчет

Лабораторная работа №2

Выполнил:

Самаров Илья

Группа

K33402

Проверил:

Добряков Д. И.

Санкт-Петербург

2023 г.

## Задача

Привязать сверстаный сайт к внешнему API средствами fetch/axios/xhr.

Вариант: сайт-портфолио

## Ход работы

Было реализовано получение карточек услуг (их наполнения) с внешнего API. Выполнена реализация сортировки этих карточек по категориям и поиск по ним

Создан скрипт addCards.js, в котором написаны две функции:

- getCardHtml, которая возвращает html-код карточки услуги
- loadServices, которая делает запрос на сервер по параметрам поиска и добавляет карточку услуги по заданным параметрам на страницу

```
allCards = []
function getCardHtml({ Head, Text, imgSrc, }) {
  // console.log(typeof Head, Head)
  func = "order("+Head+")"
  // console.log(func)
  return `
  <div class="d-flex flex-row border service-item">
    <img class="service-img" src=${imgSrc}>
    <div class="d-flex flex-column service-text">
      <h3>${Head}</h3>
      <br>
      ${Text}

      <button class="orderbutton" data-bs-toggle="modal" data-bs-target="#modal-order">purchase</button>
    </div>
  </div>
  `
}
async function loadServices(searchString = "") {

  document.getElementById("ServiceContainer").innerHTML = ""

  let url = "http://localhost:3000/services"

  url = `${url}?${searchString}`

  const response = await fetch(url)

  const responseJson = await response.json()

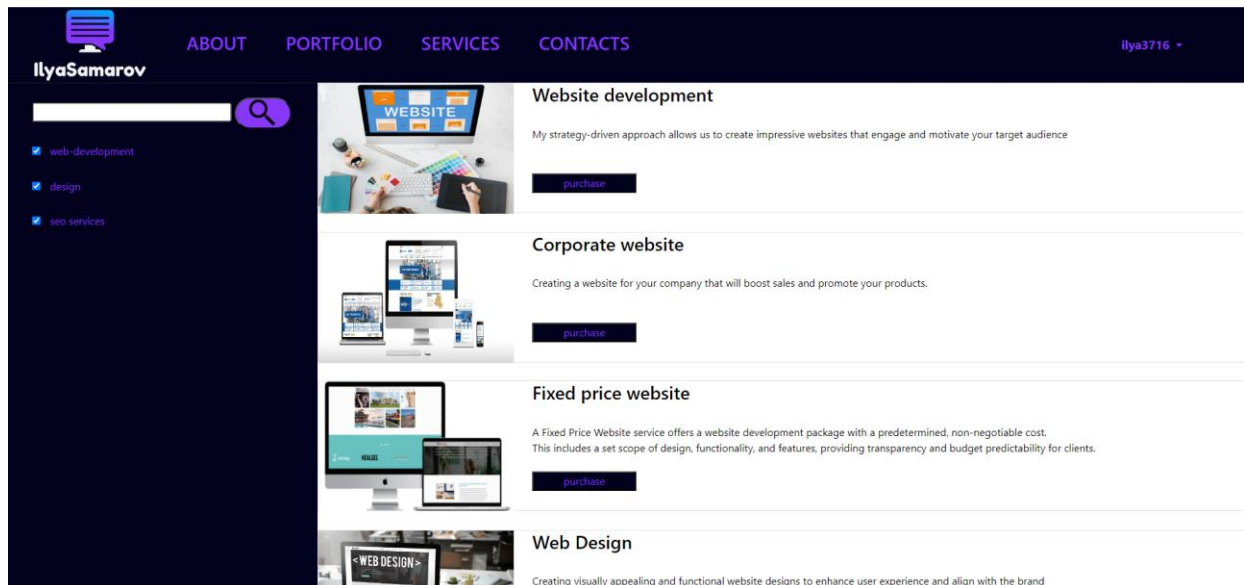
  // console.log('response', responseJson)
  for (const service of responseJson){
    allCards.push(service)
  }
  for (const service of responseJson) {

    document.getElementById("ServiceContainer").innerHTML += getCardHtml(service)
    document.getElementsByClassName
  }
}
```

Использование на странице услуг:

```
<script>
  document.addEventListener('DOMContentLoaded', () => {
    |   loadServices()
    | })
</script>
<script>
```

Результат:

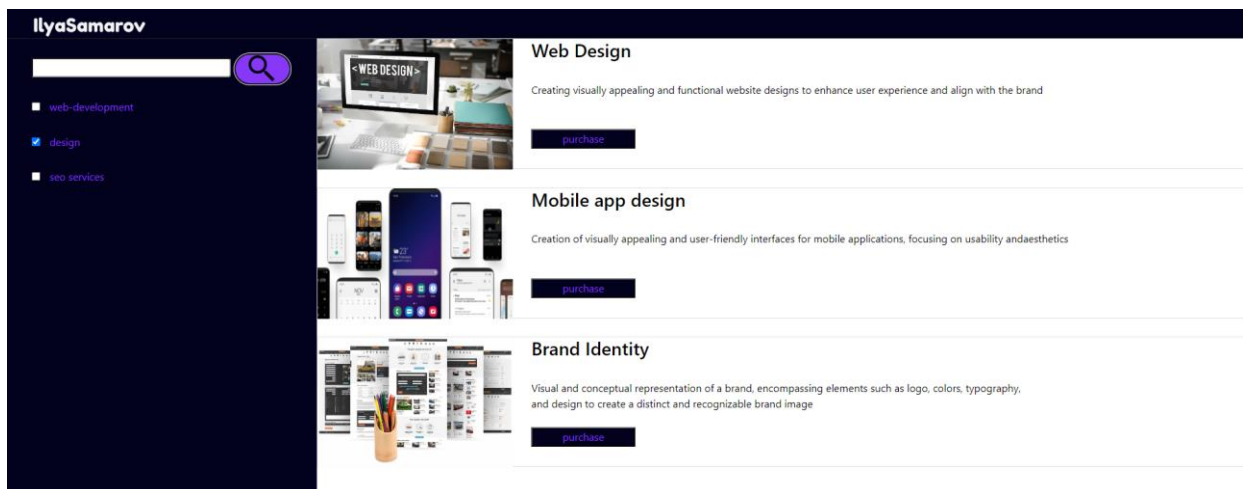


Сортировка карточек по категориям и поиск:

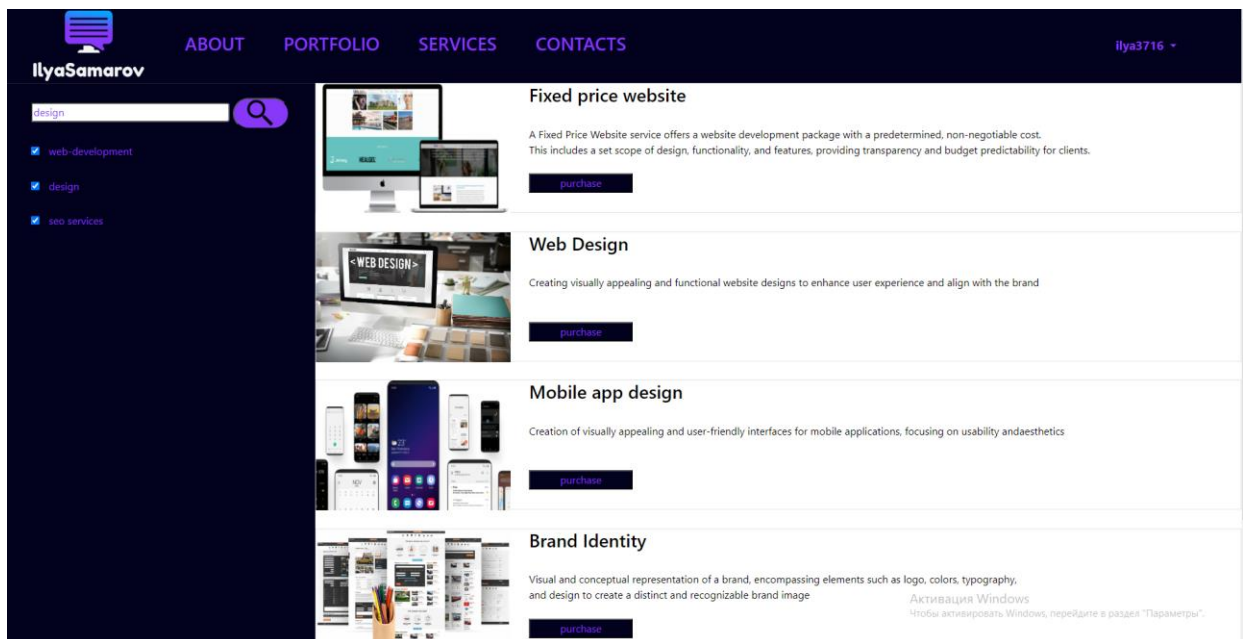
Создан скрипт search.js, который формирует параметры поиска и вызывает loadServices с этими параметрами

```
async function search(inputValue, checkbox) {  
  
    for (i = 0; i < checkbox.length; i++) {  
        if (checkbox[i]) {  
            console  
                loadServices(`Group=${i + 1}&q=${inputValue}`)  
        }  
    }  
}  
async function makeSearch() {  
  
    checks = []  
    for (i = 1; i < 4; i++)  
        checks.push(document.getElementById(`checkbox${i}`).checked)  
  
    search(document.getElementById('searchbox').value, checks)  
}
```

## Пример фильтрации по категориям:



## Пример поиска:



Через json-server реализованы логин и регистрация:

```
async function login(event) {
  event.preventDefault()
  let reload = true
  const inputs = Array.from(event.target.querySelectorAll('input'))
  const loginData = {}

  for (const input of inputs) {
    loginData[input.name] = input.value
  }

  console.log('login data', loginData)

  const response: Response
  const response = await fetch('http://localhost:3000/login', {
    method: "POST",
    body: JSON.stringify(loginData),
    headers: {
      'Content-Type': 'application/json'
    }
  })
  console.log(response)
  const responseJson = await response.json()

  const { accessToken, user } = responseJson

  console.log('response', responseJson)
  if (responseJson == "Cannot find user") {
    alert("User not found")
    reload = false
  }
  else {
    if (responseJson == "Incorrect password") {
      alert("password is incorrect")
      reload = false
    }
  }
  if (reload) {
    reload = true
    localStorage.accessToken = accessToken
    localStorage.user = JSON.stringify(user)
    console.log("reloading")
    location.reload()
  }
}
```

```

✓ async function register(event) {
  event.preventDefault()

  const inputs = Array.from(event.target.querySelectorAll('input'))
  const loginData = {}

✓   for (const input of inputs) {
    loginData[input.name] = input.value
  }

  console.log('login data', loginData)

✓   const response = await fetch('http://localhost:3000/users', {
    method: "POST",
    body: JSON.stringify(loginData),
    headers: {
      'Content-Type': 'application/json'
    }
  })

  window.location.href = "http://127.0.0.1:5500/services.html"
}

```

Также реализована логика создания и вывода заказов в профиле:

```

async function order(event) {
  event.preventDefault()
  // alert("ok")
  const Head = document.getElementById("inputGroupSelect01").value
  const date = document.getElementById("startDate").value
  const type = document.getElementById("comType").value
  var user = localStorage.getItem("user")
  var profileUsername = JSON.parse(user).id
  console.log(profileUsername)
  const response = await fetch('http://localhost:3000/orders', {
    method: "POST",
    body: JSON.stringify({
      "Head": Head,
      "date": date,
      "type": type,
      "username": profileUsername,
    }),
    headers: {
      'Content-Type': 'application/json'
    }
  })

  window.location.href = "http://127.0.0.1:5500/profile.html"
}

```

```

var user = localStorage.getItem("user")
console.log(user)
var profileUsername = JSON.parse(user).id
console.log(profileUsername)
const body = document.getElementById("Welcome")
body.innerHTML = "Welcome, " + profileUsername + "!<br>"
allCards = []
let imgSrc
function getOrderHtml({ Head, date, type },imgSrc) {

  return `
    <div class="d-flex flex-row border service-item">
      <img class="service-img" src=${imgSrc}>
      <div class="d-flex flex-column service-text">
        <h3>${Head}</h3>
        <br>
        I will contacy you ${date}
        <br>
        via
        ${type}
      </div>
    </div>
  `
}

async function loadOrders(username, orders) {
  console.log("updating", username)
  console.log(orders)
  orders.innerHTML = ""
  console.log("success")
  let url = "http://localhost:3000/orders"
  url = `${url}?username=${username}`

  const response = await fetch(url)

  const responseJson = await response.json()

  console.log('response', responseJson)
  for (const service of responseJson) {
    allCards.push(service)
  }
  for (const service of responseJson) {
    console.log()
  }
}

```



Пример оформления заказ услуги:

Order

Service Web Design

Choose preferred date and type of communication

Date 26.01.2024

Type Telegram

order

Corporate website

Creating a website for your company that will boost sales and promote your products.

purchase

Результат в профиле:

IlyaSamarov

ABOUT PORTFOLIO SERVICES CONTACTS

Welcome, ilya3716!

Here you can see your orders!

	<b>Web Design</b> I will contacy you 11.12.2023 via VK
	<b>Corporate website</b> I will contacy you 2023-12-27 via Telegram
	<b>Web Design</b> I will contacy you 2024-01-26 via Telegram

## Вывод

В данной лабораторной работе я познакомился json-server, на практике изучил работу с API (его подключение к сайту) и fetch. Реализовал поиск и фильтрацию, а также авторизацию на своем сайте