

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Фронт-энд разработка

Отчет

Лабораторная работа №3

Выполнила:

Скороходова Елена

Группа К33392

Проверил:

Добряков Д.И.

Санкт-Петербург

2023 г.

Задача

Мигрировать ранее написанный сайт на фреймворк Vue.JS.

Минимальные требования:

- Должен быть подключён роутер
- Должна быть реализована работа с внешним API
- Разумное деление на компоненты
- Использование composable

Ход работы

Создаем главный (root) компонент App.vue в папке views. С помощью него далее я отображаю страницу. У меня есть постоянные компоненты Header и Footer, и изменяющийся Main в зависимости от страницы.

```
dress_shop_vue > src > views > App.vue
1  <template>
2    <div>
3      <Header />
4      <router-view />
5      <Footer />
6    </div>
7  </template>
8
9  <script>
10   import Header from "@components/Header.vue";
11   import Footer from "@components/Footer.vue";
12
13   export default {
14     components: {
15       Header,
16       Footer,
17     },
18   };
19 </script>
20
21 <style>
22 html,
23 body {
24   height: 100%;
25 }
26
27 body {
28   background: var(--bg-color);
29   min-width: 320px;
30   font-family: "Montserrat", sans-serif;
31   color: #2d2d2d;
32 }
```

```

dress_shop_vue > src > components > Main.vue
1 <!-- eslint-disable vue/multi-word-component-names -->
2 <template>
3   <main>
4     <div class="container-fluid px-0">
5       <div
6         id="carouselExampleAutoplaying"
7         class="carousel slide"
8         data-bs-ride="carousel"
9         data-bs-interval="5000">
10        <div class="carousel-indicators d-md-none">
11          <!-- Скрыть на маленьких экранах -->
12          <button
13            type="button"
14            data-bs-target="#carouselExampleIndicators"
15            data-bs-slide-to="0"
16            class="active"
17            aria-current="true"
18            aria-label="Slide 1"></button>
19          <button
20            type="button"
21            data-bs-target="#carouselExampleIndicators"
22            data-bs-slide-to="1"
23            aria-label="Slide 2"></button>
24          <button
25            type="button"
26            data-bs-target="#carouselExampleIndicators"
27            data-bs-slide-to="2"
28            aria-label="Slide 3"></button>
29        </div>
30        <div class="carousel-inner">
31          <div class="carousel-item active">
32            
<template>
  <header id="header">
    <div class="header container-fluid">
      <div class="container">
        <div class="row align-items-center justify-content-between">
          <nav class="navbar col-lg-5 col-1 navbar-expand-lg">
            <button
              class="navbar-toggler"
              type="button"
              data-bs-toggle="collapse"
              data-bs-target="#navbarSupportedContent"
              aria-controls="navbarSupportedContent"
              aria-expanded="false"
              aria-label="Переключатель навигации"
            >
            <span class="navbar-toggler-icon">
            </span>
          </button>
          <div class="collapse navbar-collapse" id="navbarSupportedContent">
            <ul class="navbar-nav me-auto mb-2 mb-lg-0">
              <li class="nav-item">
                <a class="nav-link active" aria-current="page" href="#">
                  >Новинки</a>
                </li>
              <li class="nav-item">
                <router-link to="/catalog" class="nav-link">
                  Одежда
                </router-link>
              </li>
              <li class="nav-item">

```

С использованием Vue 3, Vue Router и Bootstrap, я создала main.js - основа для создания Vue.js приложения. Здесь я создаю экземпляр приложения с использованием App, настраиваю маршрутизацию: создаю экземпляр маршрутизатора Vue с использованием “createRouter” и определяю маршруты, каждый из которых связан с соответствующим компонентом Vue.

```

dress_shop_vue > src > JS main.js > ...
1  import { createApp } from 'vue';
2  import App from './views/App.vue';
3  import Main from './components/Main.vue';
4  import Enter from './components/Enter.vue';
5  import Registration from './components/Registration.vue';
6  import Order from './components/Order.vue';
7  import Cabinet from './components/Cabinet.vue';
8  import Catalog from './components/Catalog.vue';
9  import { createRouter, createWebHistory } from 'vue-router';
10 import './style.css';
11 import 'bootstrap/dist/css/bootstrap.min.css';
12 import 'bootstrap/dist/js/bootstrap.bundle.min.js';
13
14 //экземпляр приложения
15 const app = createApp(App);
16
17 // роутер
18 const router = createRouter({
19   history: createWebHistory(),
20   routes: [
21     {
22       path: '/main',
23       name: 'Main',
24       component: Main,
25     },
26     {
27       path: '/enter',
28       name: 'Enter',
29       component: Enter,
30     },
31     {
32       path: '/registration',
33       name: 'Registration',
34       component: Registration,
35     },
36     {
37       path: '/order',

```

Далее переходим к настройке регистрации и авторизации. Для этого я использую фреймворк Express(фреймворк для создания веб-приложений на Node.js) вместе с json-server. Здесь происходит настройка Express, статических файлов и парсинга JSON, настройка json-server, маршрутизация Express.

```
dress_shop_vue > server > JS server.js > ...
1  const express = require('express');
2  const path = require('path');
3  const bodyParser = require('body-parser');
4  const jsonServer = require('json-server');
5
6  const app = express();
7  const PORT = process.env.PORT || 3001;
8
9  app.use(express.static(path.join(__dirname, '../dist')));
10 app.use(bodyParser.json());
11
12
13 const jsonServerRouter = jsonServer.router(path.join(__dirname, '../db.json'));
14 app.use('/api', jsonServerRouter);
15
16
17 app.get('/main', (req, res) => {
18   res.sendFile(path.join(__dirname, '../dist/index.html'));
19 });
20
21
22 app.get('*', (req, res) => {
23   res.sendFile(path.join(__dirname, '../dist/index.html'));
24 });
25
26 app.listen(PORT, () => {
27   console.log(`Сервер запущен на порту ${PORT}`);
28 });
29 |
```

Для регистрации и авторизации я создаю composables, потом в соответствующие компоненты импортирую, чтобы использовать в них данные.

Во входе я проверяю на правильность пароля и почты, если в users их нет, то выдается сообщение “Неверные логин или пароль”. При успешном входе пользователя перенаправляет на личный кабинет.

```

dress_shop_vue > src > composables > JS useLogin.js > useLogin > loginUser
1  import { ref } from 'vue';
2  import { useRouter } from 'vue-router';
3  import axios from 'axios';
4
5  export default function useLogin() {
6    const email = ref('');
7    const password = ref('');
8    const loginError = ref(false);
9    const router = useRouter();
10
11    const loginUser = async () => {
12      try {
13
14        const response = await axios.get(`http://localhost:3001/users?email=${email.value}&password`
15
16
17        if (response.data.length > 0) {
18
19          loginError.value = false;
20          router.push('/cabinet');
21        } else {
22
23          loginError.value = true;
24        }
25      } catch (error) {
26        console.error('Ошибка при входе:', error);
27
28        loginError.value = true;
29      }
30    }
31  };
32
33  return {
34    email,
35    password,
36    loginError,
37    loginUser,

```

```

dress_shop_vue > src > components > Enter.vue
1  <!-- eslint-disable vue/multi-word-component-names -->
2  <template> ...
48 </template>
49
50 <script>
51 import useLogin from "@/composables/useLogin";
52
53 export default {
54   setup() {
55     const { email, password, loginError, loginUser } = useLogin();
56
57     return {
58       email,
59       password,
60       loginError,
61       loginUser,
62     };
63   },
64 };
65 </script>
66
67 <style scoped>
68 /* Форма входа */
69

```

При регистрации есть проверка на соответствие пароля, на вид почты и на галочку согласия с политикой конфиденциальности.

```
dress_shop_vue > src > composables > JS useRegistration.js > ...
1  import { ref } from 'vue';
2  import { useRouter } from 'vue-router';
3
4  export default function useRegistration() {
5    const email = ref('');
6    const password = ref('');
7    const confirmPassword = ref('');
8    const agreed = ref(false);
9    const router = useRouter();
10
11    const validatePasswords = () => {
12      if (password.value !== confirmPassword.value) {
13        console.error('Пароли не совпадают');
14        return false;
15      }
16      return true;
17    };
18
19    const validateAgreement = () => {
20      if (!agreed.value) {
21        console.error('Необходимо согласиться с политикой конфиденциальности');
22        return false;
23      }
24      return true;
25    };
26
27    const registerUser = async () => {
28      if (!validatePasswords() || !validateAgreement()) {
29        return;
30      }
31
32      try {
33        const response = await fetch('http://localhost:3001/users', {
34          method: 'POST',
35          headers: {
36            'Content-Type': 'application/json',
37          },
38          body: JSON.stringify({
39            email: email.value,
40            password: password.value,
41            confirmPassword: confirmPassword.value,
42            agreed: agreed.value,
43          })
44        });
45        if (!response.ok) {
46          throw new Error('Ошибка при регистрации');
47        }
48        router.push('/login');
49      } catch (error) {
50        console.error('Ошибка при регистрации:', error);
51      }
52    };
53  }
54
```

localhost:8080/enter

Новинки

Одежда

Аксессуары

Sale

SARAFAN
COLLECTION

Поиск

🛒

👤

Вход

Адрес электронной почты

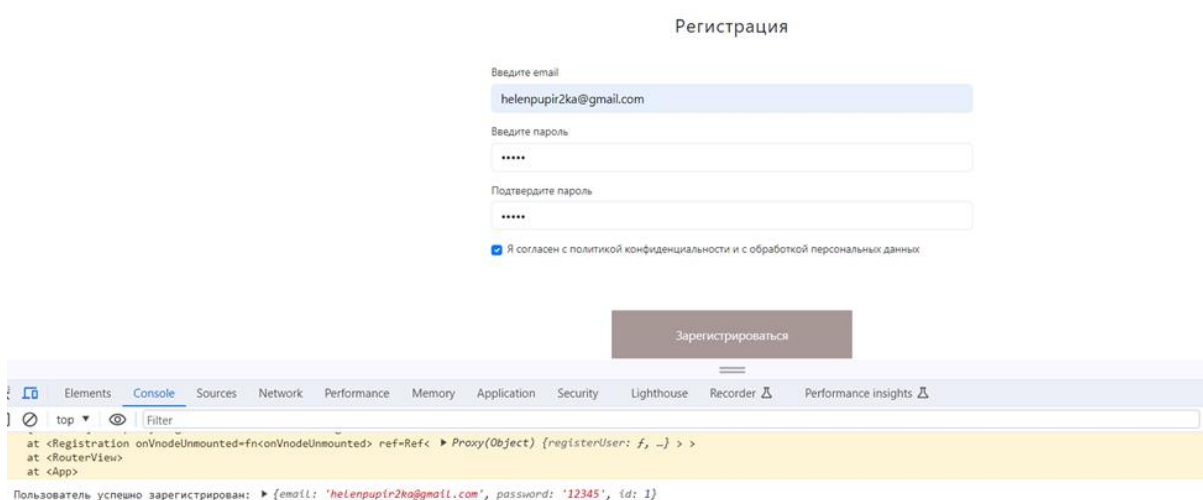
lala@gmail.com

Пароль

Неверный логин или пароль

Отправить

```
dress_shop_vue > src > components > Registration.vue
1  <!-- eslint-disable vue/multi-word-component-names -->
2  <template>...
63 </template>
64
65 <script>
66 import useRegistration from '../composables/useRegistration';
67
68 export default {
69   setup() {
70     const {
71       email,
72       password,
73       confirmPassword,
74       agreed,
75       registerUser,
76     } = useRegistration();
77
78     return {
79       email,
80       password,
81       confirmPassword,
82       agreed,
83       registerUser,
84     };
85   },
86 }
```



Данные о зарегистрированном пользователе я сохраняю в файл-бд db.json.


```
dress_shop_vue > server > {} db.json > ...
1 {
2   "users": [
3     {
4       "email": "helenpupir2ka@gmail.com",
5       "password": "12345",
6       "id": 1
7     },
8     {
9       "email": "vutia@gmail.com",
10      "password": "12345",
11      "id": 2
12    },
13    {
14      "email": "lala@gmail.com",
15      "password": "123",
16      "id": 3
17    },
18    {
19      "email": "vu6tia@gmail.com",
20      "password": "123",
21      "id": 4
22    },
23    {
24      "email": "v453utia@gmail.com",
25      "password": "123",
26      "id": 5
27    }
28  ]
29 }
```

Также реализовано взаимодействие с внешним API (получение карточек товаров). Для этого в composables создан useProductCard.json, который получает данные с api, и выводятся они в компоненте Catalog.vue.

```

dress_shop_vue > src > composables > JS useProductCards.js > useProductCards > fetchProductData
1  import { ref } from 'vue';
2
3  export function useProductCards() {
4    const productCards = ref([]);
5
6
7    async function fetchProductData(productId) {
8      try {
9        const response = await fetch(`https://fakestoreapi.com/products/${productId}`);
10       const data = await response.json();
11       const card = createProductCard(data);
12       productCards.value.push(card);
13
14
15       console.log('Я молодец!');
16       console.log('Полученная карточка продукта:', card);
17
18     } catch (error) {
19       console.error('Произошла ошибка при загрузке данных для продукта ${productId}:', error);
20     }
21   }
22
23   function createProductCard(data) {
24     const card = document.createElement("div");
25     card.className = "col";
26     card.classList.add("card", "h-100");
27
28     const img = document.createElement("img");
29     img.className = "card-img-top";
30     img.src = data.image;
31     img.alt = data.title;
32
33     const cardBody = document.createElement("div");
34     cardBody.className = "card-body";
35
36     const title = document.createElement("h5");
37     title.className = "card-title";

```

```

dress_shop_vue > src > components > Catalog.vue
1  <!-- eslint-disable vue/multi-word-component-names -->
2  <template> ...
123 </template>
124
125 <script>
126 import { useProductCards } from "@composables/useProductCards";
127
128 export default {
129   setup() {
130     const {
131       productCards,
132       fetchProductData,
133       sortByPriceDescending,
134       sortByPriceAscending,
135     } = useProductCards();
136     console.log("Я тут!");
137     console.log("Полученный массив:", productCards);
138     return {
139       productCards,
140       fetchProductData,
141       sortByPriceDescending,
142       sortByPriceAscending,
143     };
144   },
145 };
146 </script>
147
148 <style scoped> ...
271 </style>
272

```

Вывод

В данной лабораторной работе я мигрировала свой сайт на Vue, добавила регистрацию и авторизацию при помощи JSON сервера.