САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

Дисциплина: Фронд-энд разработка

Отчет
Лабораторная работа №3
“Разработка одностраничного веб-приложения (SPA) с использованием фреймворка Vue.JS”

Выполнил:

Кондрашов Е. Ю.

Группа K33402

Проверил:
Добряков Д. И.

Санкт-Петербург

2021 г.

Задача

Мигрировать ранее написанный сайт на фреймворк Vue.JS.

Ход работы

Исходный код:

https://github.com/e-kondr01/ITMO-ICT-Frontend-2021/tree/master/labs/K33402/Kondrashov_Egor/lw3

Подключённый роутер:

```
import Index from "@/views/Index.vue"
import Login from "@/views/Login.vue"
import MyBookings from "@/views/MyBookings.vue"
import Registration from "@/views/Registration.vue"
import SearchResults from "@/views/SearchResults.vue"
import Vue from "vue"
import VueRouter from "vue-router"

Vue.use(VueRouter)

const routes = [
 {
   path: "/",
   name: "Index",
   component: Index
 },
 {
   path: "/login",
   name: "Login",
   component: Login
 },
 {
   path: "/register",
   name: "Registration",
   component: Registration
 },
 {
   path: "/my-bookings",
   name: "MyBookings",
   component: MyBookings
 },
 {
   path: "/search-results",
   name: "SearchResults",
   component: SearchResults
 }
]

const router = new VueRouter({
 mode: "history",
 base: process.env.BASE_URL,
 routes
```

```
})

export default router
```

Работа с внешним API:

```
<script>
import AppHeader from "../components/AppHeader"
import SearchResultsMain from "../components/SearchResultsMain.vue"

export default {
 name: "SearchResults",

 components: {
   AppHeader,
   SearchResultsMain
 },

 methods: {
   async getHotelItems() {
     try {
       let city = this.$route.query.city
       if (!city) {
         city = "-553173"
       } else {
         city = await this.getCityIdByName(city)
       }
       let numberOfGuests = this.$route.query.numberOfGuests
       if (!numberOfGuests) {
         numberOfGuests = 1
       }
       let arrivalDatetime = this.$route.query.arrivalDatetime
       if (!arrivalDatetime) {
         arrivalDatetime = "2022-07-24"
       }
       let departureDatetime = this.$route.query.departureDatetime
       if (!departureDatetime) {
         departureDatetime = "2022-07-25"
       }
       const response = await this.axios.get(

`https://booking-com.p.rapidapi.com/v1/hotels/search?dest_type=city&che
ckin_date=${arrivalDatetime}&room_number=1&checkout_date=${departureDat
etime}&order_by=popularity&dest_id=${city}&adults_number=${numberOfGues
ts}&units=metric&filter_by_currency=AED&locale=en-gb`,
         {
           headers: {
             "x-rapidapi-host": "booking-com.p.rapidapi.com",
             "x-rapidapi-key":
               "7364d7fb71msh2d462a6fff9d1b1p102b53jsnff2949c01c27"
           }
         }
       )

       if (response.status !== 200) {
         throw new Error(response.error)
       }
       this.hotelItems = response.data.result
       console.log(this.hotelItems)
```

```
    } catch (e) {
      console.error("Error from Booking API: ", e)
    }
  },
  async getCityIdByName(city) {
    const res = await this.axios.get(

`https://booking-com.p.rapidapi.com/v1/hotels/locations?locale=en-gb&na
me=${city}`,
      {
        headers: {
          "x-rapidapi-host": "booking-com.p.rapidapi.com",
          "x-rapidapi-key":
            "7364d7fb71msh2d462a6fff9d1b1p102b53jsnff2949c01c27"
        }
      }
    )
    if (res.status === 200) {
      return res.data[0].dest_id
    } else {
      return null
    }
  }
 },
 beforeMount() {
   this.getHotelItems()
 },
 data: () => ({
   hotelItems: []
 })
}
</script>
```

Деление на компоненты:

Пример со страницей Index.vue.

Представление:

```
<template>
 <section class="page-content">
   <app-header />
   <index-main />
 </section>
</template>
```

Компонента app-header:

```
<template>
 <header>
   <b-navbar>
     <div class="container">
       <header-nav-brand />
       <b-navbar-nav class="justify-content-end">
         <header-nav-item navLink="login" navText="Вход" />
         <header-nav-item navLink="register" navText="Регистрация" />
                  <header-nav-item  navLink="my-bookings"  navText="Мои
бронирования" />
```

```
      </b-navbar-nav>
    </div>
  </b-navbar>
 </header>
</template>
```

Компонента header-nav-brand:

```
<template>
 <a class="navbar-brand text-white" href="/">Бронирование жилья</a>
</template>
```

Компонента header-nav-item:

```
<template>
 <b-nav-item class="px-3" :href="navLink">
   {{ navText }}
 </b-nav-item>
</template>

<script>
export default {
 name: "HeaderNavItem",
 props: {
   navLink: String,
   navText: String
 }
}
</script>
```

Вывод

В ходе работы сайт, реализованный с помощью HTML, CSS и JS был перенесён на Vue.js.

Был подключен роутинг с помощью vue-router, благодаря чему можно переходить к страницам, указывая их имя, параметры и query параметры.

Для работы с внешним API используется библиотека axios. Запросы можно вызывать, определяя их в методах, расположенных в секции <script> компонент.

В Vue.js используется компонентный подход: сайт можно поделить на смысловые части, каждую из которых в свою очередь можно разделить на более мелкие. Разделение на компоненты помогает переиспользовать часто встречающиеся элементы интерфейса и упрощает процесс разработки больших сайтов.