САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

Дисциплина: Фронд-энд разработка

Отчет

Лабораторная работа 3 на тему: "Разработка одностраничного веб-приложения (SPA) с использованием фреймворка Vue.JS "

Выполнил: Безруков А.И. (К33402)

Проверил: Добряков Д.И.

Санкт-Петербург, 2022 г.

Цель работы: Перенести написанный ранее сайт на фреймворк Vue.JS

Ход работы:

Роутер:

```
import Vue from 'vue'
import VueRouter from 'vue-router'
import Main from '@/views/Main.vue'
import Signup from '@/views/Signup.vue'
import Account from '@/views/Account.vue'
import Search from '@/views/Search.vue'

Vue.use(VueRouter)

const routes = [
  {
    path: '/',
    name: 'Main',
    component: Main
  },
  {
    path: '/signup',
    name: 'Signup',
    component: Signup
  },
  {
    path: '/account',
    name: 'Account',
    component: Account
  },
  {
    path: '/search',
    name: 'Search',
    component: Search
  }
]

const router = new VueRouter({
  mode: 'history',
  base: process.env.BASE_URL,
  routes
})

export default router
```

Работа с внешнем API: (компонент Weather.vue)

```
<script>
import axios from 'axios'
export default {
  name: 'Weather',
  data: () => ({
    city: '',
    savedCity: '',
    info: {
```

```
      main: {
        temp: '',
        feels_like: ''
      },
      wind: {
        speed: ''
      }
    },
    location: {
      coords: {
        latitude: '',
        longitude: ''
      }
    }
  }),
  methods: {
    submitCity () {
      localStorage.setItem('city', this.city)
      this.savedCity = this.city
      this.$refs.form.reset()
    },
    getWeather () {
      axios
        .get('https://api.openweathermap.org/data/2.5/weather?q=' + this.city +
'&appid=314864fa7ae959cda85ac9d534443060&lang=en&units=metric')
        .then(response => (this.info = response.data))
    },
    async getLocation () {
      navigator.geolocation.getCurrentPosition(pos => {
        this.location = pos
      })
      axios
        .get('https://api.openweathermap.org/data/2.5/weather?lat=' +
this.location.coords.latitude + '&lon=' + this.location.coords.longitude +
'&appid=62f76307202d2bbb00f83a4de8ac7393&lang=en&units=metric')
        .then(response => (this.info = response.data))
    }
  }
}
</script>
```

Компонента (на примере навигации – Navigation.vue):

```
<template>
<nav class="navbar navbar-expand-lg navbar-dark">
    <!--SVG sprite-->
    <svg display="none">
        <symbol id="search" viewBox="0 0 612.01 612.01">
            <g>
                <path
d="M606.209,578.714L448.198,423.228C489.576,378.272,515,318.817,515,253.393C514.9
8,113.439,399.704,0,257.493,0
```

```html
                    C115.282,0,0.006,113.439,0.006,253.393s115.276,253.393,257.48
7,253.393c61.445,0,117.801-21.253,162.068-56.586
                    l158.624,156.099c7.729,7.614,20.277,7.614,28.006,0C613.938,59
8.686,613.938,586.328,606.209,578.714z M257.493,467.8
                    c-120.326,0-217.869-95.993-217.869-
214.407S137.167,38.986,257.493,38.986c120.327,0,217.869,95.993,217.869,214.407
                    S377.82,467.8,257.493,467.8z"/>
            </g>
        </symbol>
        <symbol id="account" viewBox="0 0 792 792">
            <g>
                <path d="M568.037,404.712c31.68-42.743,50.713-97.515,50.713-
157.212C618.75,110.806,519.032,0,396,0
                    S173.25,110.806,173.25,247.5c0,59.697,19.033,114.469,50.713,1
57.212C165.8,424.834,123.75,479.481,123.75,544.5v99
                    c0,82.021,66.479,148.5,148.5h247.5c82.021,0,148.5-
66.479,148.5-148.5v-99C668.25,479.481,626.2,424.834,568.037,404.712z
                    M222.75,247.5c0-109.346,77.566-198,173.25-
198s173.25,88.654,173.25,198s-77.566,198-173.25,198S222.75,356.846,222.75,247.5z
                    M618.75,631.125c0,61.504-54.4,111.375-
121.498,111.375H294.748c-67.097,0-121.498-49.871-121.498-111.375v-74.25
                    c0-52.618,39.922-96.451,93.456-
108.108C303.188,477.774,347.737,495,396,495s92.812-17.226,129.294-46.233
                    c53.534,11.657,93.456,55.49,93.456,108.108V631.125z"/>
            </g>
        </symbol>
        <symbol id="signup" viewBox="0 0 612 612">
            <g>
                <path d="M331.685,425.378c-7.478,7.479-
7.478,19.584,0,27.043c7.479,7.478,19.584,7.478,27.043,0l131.943-131.962
                    c3.979-3.979,5.681-9.276,5.412-14.479c0.269-5.221-1.434-
10.499-5.412-14.477L358.728,159.56
                    c-7.459-7.478-19.584-7.478-27.043,0c-7.478,7.478-
7.478,19.584,0,27.042l100.272,100.272H19.125C8.568,286.875,0,295.443,0,306
                    c0,10.557,8.568,19.125,19.125,19.125h412.832L331.685,425.378z
M535.5,38.25H153c-42.247,0-76.5,34.253-76.5,76.5v76.5h38.25
                    v-76.5c0-21.114,17.117-38.25,38.25-
38.25h382.5c21.133,0,38.25,17.136,38.25,38.25v382.5c0,21.114-17.117,38.25-
38.25,38.25H153
                    c-21.133,0-38.25-17.117-38.25-38.25v-
76.5H76.5v76.5c0,42.247,34.253,76.5,76.5,76.5h382.5c42.247,0,76.5-34.253,76.5-
76.5
                    v-382.5C612,72.503,577.747,38.25,535.5,38.25z"/>
            </g>
        </symbol>
    </svg>
<div class="container" id="nav">
        <a href = "/" class="navbar-brand">Pogodka</a>
        <button
            class="navbar-toggler"
            type="button"
```

```html
                aria-label="Navigation bar toggler"
                data-bs-toggle="collapse"
                data-bs-target="#navmenu">
                <span class="navbar-toggler-icon"></span>
            </button>
            <div class="collapse navbar-collapse" id="navmenu">
                <ul class="navbar-nav ms-auto">
                    <li class="nav-item">
                        <a href="/search" class="nav-link">
                            <svg class="fig">
                                <use xlink:href="#search"></use>
                            </svg>
                        </a>
                    </li>
                    <li class="nav-item">
                        <a href="/account" class="nav-link">
                            <svg class="fig">
                                <use xlink:href="#account"></use>
                            </svg>
                        </a>
                    </li>
                    <li class="nav-item">
                        <a href="/signup" class="nav-link">
                            <svg class="fig">
                                <use xlink:href="#signup"></use>
                            </svg>
                        </a>
                    </li>
                </ul>
            </div>
            </div>
        </nav>
</template>

<script>
export default {
  name: 'Navigation'
}
</script>

<style>
html, body {
    height: 100%;
    }

body {
    display: flex;
    flex-direction: column;
    background-color: #00203FFF;
    }
nav {
```

```
    background-color: #00203FFF;
    }

.fig{
    display: block;
    fill: white;
    width:1.5em;
    height: 1.5em;
}
</style>
```

Компонента (на примере страницы регистрации – Register.vue):

```
<template>
  <section class="p-5">
        <div class="container">
            <div class = "d-md-flex align-items-center justify-content-around">
                <h3 class="mb-3">First time here? Sign up!</h3>
                    <form>
                        <div class="mb-3">
                            <label for="InputName" class="form-
label">Name</label>
                            <input type="name" class="form-control"
id="InputName">
                        </div>
                        <div class="mb-3">
                            <label for="InputEmail" class="form-label">Email
address</label>
                            <input type="email" class="form-control"
id="InputEmail" aria-describedby="emailHelp">
                        </div>
                        <div class="mb-3">
                            <label for="InputPassword" class="form-
label">Password</label>
                            <input type="password" class="form-control"
id="InputPassword">
                        </div>
                        <div class="mb-3">
                            <label for="InputPassword" class="form-label">Repeat
the password</label>
                            <input type="password" class="form-control"
id="InputPassword">
                        </div>
                        <div class="mb-3 form-check">
                            <input type="checkbox" class="form-check-input"
id="exampleCheck1">
                            <label class="form-check-label" for="exampleCheck1"
required>I agree to the <a href='#'>terms and conditions</a></label>
                          </div>
                        <button type="submit" class="btn btn-light">Sign
up</button>
```
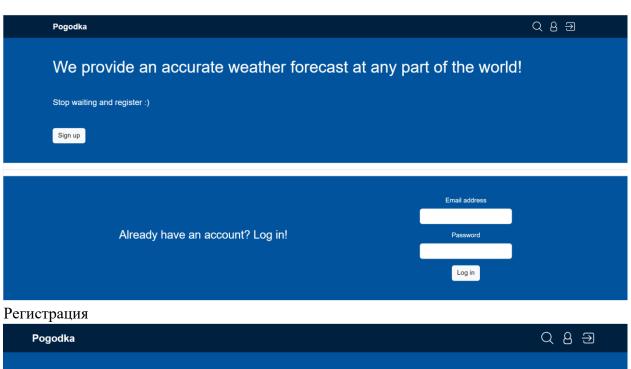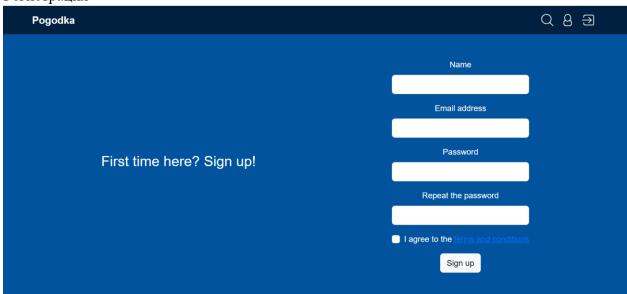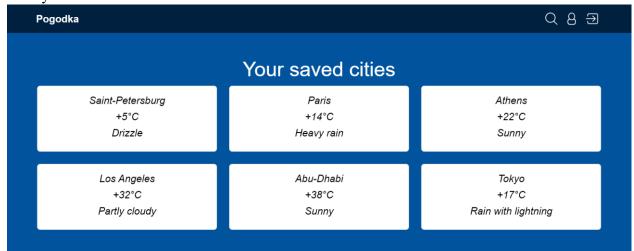
```
                </form>
            </div>
        </div>
    </section>
</template>

<script>
export default {
  name: 'Register'
}
</script>

<style>
  section{
    background-color: #00539CFF;
}
  h3, label {
    color: white;
  }
</style>
```

Представление: (Signup.vue):

```
<template>
    <div>
        <Navigation/>
        <Register/>
    </div>
</template>

<script>
import Navigation from '@/components/Navigation'
import Register from '@/components/Register'
export default {
  name: 'Signup',
  components: {
    Register,
    Navigation
  }
}
</script>
```

Страницы приложения:

- Главная

- Регистрация

- Аккаунт

- Поиск погоды

Вывод: В ходе данной работы, ранее написанный сайт с помощью HTML, CSS и Javascript был перенесен на фреймворк Vue.JS. Библиотека axios дала возможность поработать с внешним API. Поскольку фреймворк Vue.JS использует компонентный подход - страницы сайта были разделены на более мелкие компоненты, используемые при создании новых страниц.