

Министерство образования и науки Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО
ОБРАЗОВАНИЯ

УНИВЕРСИТЕТ ИТМО

Факультет инфокоммуникационных технологий

Образовательная программа 09.03.03

Направление подготовки (специальность) Мобильные сетевые технологии

О Т Ч Е Т

о курсовой работе

Тема задания: разработка одностраничного веб-приложения (SPA) с использованием
фреймворка Vue.js

Обучающийся Каратецкая Мария Юрьевна, К33402

Руководитель: Добряков Давид Ильич, преподаватель

Оценка за курсовую работу ____

Подписи членов комиссии:

_____ (Добряков Д.И.)
(подпись)

Дата ____

Санкт-Петербург
2020

ВВЕДЕНИЕ	4
Актуальность	4
Цели и задачи	4
ГЛАВА 1. СРЕДСТВА РАЗРАБОТКИ И ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ	5
1.1 Средства разработки	5
1.2 Функциональные требования	5
ГЛАВА 2. ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ	6
2.1. Проектирование и реализация моделей	6
2.2 Реализация фронтенда	6
2.3 Реализация вывода билетов пользователя	10
ЗАКЛЮЧЕНИЕ	12
СПИСОК ЛИТЕРАТУРЫ	13

ВВЕДЕНИЕ

Актуальность

Сервисы для поиска билетов позволяют людям быстро найти и определить для себя маршрут следования, наличие личного кабинета упрощает процесс поиска данных по уже купленным билетам и распределения своего времени для деловых поездок и путешествий. Использование Vue.js для разработки одностраничного приложения оправдано в первую очередь тем, что одностраничные приложения, как правило, быстрее загружают страницы сайта, что ведёт к лучшему восприятию пользователем процесса взаимодействия с ним.

Цели и задачи

1. Определение средств разработки
2. Определение функциональных требований
3. Проектирование и реализация фронтенда

ГЛАВА 1. СРЕДСТВА РАЗРАБОТКИ И ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ

1.1 Средства разработки

Для разработки фронтенда был использован фреймворк Vue.js, так как его изучению была посвящена часть курса “Фронт-энд разработка”. Также для специализированного функционала использовались библиотеки Axios, BootstrapVue, Vue router.

1.2 Функциональные требования

1. Набрать минимально осмысленное число функциональных страниц, покрывающее следующий функционал: логин, регистрация, сброс пароля, поиск билетов, вывод результата, вход пользователя и вывод данных по билетам пользователя,

ГЛАВА 2. ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ

2.1. Проектирование и реализация моделей

Основной сущностью веб-приложения по выбранной мной тематике является билеты. Действия в приложении могут совершать все пользователи, но вывод данных о купленных билетах, могут только зарегистрированные пользователи. Пользователь может сменить данные входа. Также для всех билетов своя картинка, которая отображает город прилета.

Исходя из этого, был определён следующий список страниц, необходимый к реализации:

1. Регистрация
2. Вход
3. Смена пароля
4. Главная страница/страница поиска по билетам
5. Страница с результатами поиска
6. Мои билеты
7. Контакты

2.2 Реализация фронтенда

На каждой странице приложения должна присутствовать шапка проекта с навигацией. Навигацию между страницами удобно осуществлять с помощью библиотеки Vue Router. Содержание файла router/index.js, который определяет доступные в приложении маршруты.

```
1  import Vue from 'vue'
2  import VueRouter from 'vue-router'
3  import MyTicketPage from '../views/MyTicketPage'
4  import AnswerPage from '../views/AnswerPage'
5  import SignupPage from '../views/SignupPage'
6  import SignInPage from '../views/SignInPage'
7  import MainPage from '../views/MainPage'
8  import ContactPage from '../views/ContactPage'
9  import ChangePage from '../views/ChangePage'
10
11  Vue.use(VueRouter)
12
13  const routes = [
14    {
15      path: '/Signup',
16      name: 'Signup',
17      component: SignupPage
18    },
19  ],
```

Для простого и быстрого создания адаптивной верстки страницы можно применить библиотеку Bootstrap Vue. Рассмотрим это на примере главной страницы

```

<div class="container">
  <div class="row">
    <div class="col-12 col-lg-5">
      <form class="form-fly" @submit.prevent="warn()" method="get">
        <div class="mb-3">
          <label class="label-text" form="form">From: </label>

```

Для отправки формы на бэкэнд, используем библиотеку axios. Для быстрого обмена данными, мы сохраняем результаты в локальное хранилище, которые потом используем на других страницах.

```

methods: {
  warn: function () {
    axios.get('http://localhost:3000/tickets?from=${this.from}&to=${this.to}&date=${this.date}').then(response => {
      console.log('123')
      if (response.data.length !== 0) {
        this.form = response.data
        console.log(response.data[0])
        localStorage.setItem('profil', JSON.stringify(this.form))
        this.Class = ''
      } else {
        this.ans = '1'
        console.log(this.ans)
        localStorage.setItem('pro', JSON.stringify(this.ans))
        this.answer = JSON.parse(localStorage.getItem('pro'))
      }
    })
  }
}

```

На примере одной страницы(Главная страница) покажем, как собираются несколько компонентов

```

<template>
  <div class="about">
    <Header/>
    <Main/>
    <Footer/>
  </div>
</template>

<script>
import Header from '../components/Header'
import Footer from '../components/Footer'
import Main from '../components/Main'
export default {
  name: 'MainPage',
  components: {
    Main,
    Footer,
    Header
  }
}
</script>

```

Представление главной страницы

```

<template>
  <footer>
    <div class="container mt-3">
      <div class="row">
        <div class="col-4 foot">...</div>
        <div class="col-4 foot">...</div>
        <div class="col-4 foot">...</div>
      </div>
    </div>
    <div class="container">
      <p class="around">&copy; AroundFly 2017-2021</p>
      <div class="social-media center">...</div>
    </div>
  </footer>
</template>

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

<style scoped>
</style>

```

Содержимое компонента футера

```

<template>
  <header class="container container-fluid mb-5">
    <nav class="navbar navbar-expand-lg navbar-light mt-1">
      <a class="navbar-brand" href="/Main">
        
        <button class="navbar-toggler">...</button>
      </nav>
      <div class="collapse navbar-collapse" id="navbarSupportedContent">
        <ul class="navbar-nav me-auto menu_ul mb-2 mb-lg-0 mr-0">
          <li class="nav-item menu_nav">
            <a class="nav-link" href="/Main">Главная</a>
          </li>
          <li class="nav-item menu_nav">
            <a class="nav-link" href="/MyTicket">Мои билеты</a>
          </li>
          <li class="nav-item menu_nav">
            <a class="nav-link" href="/Contact">Контакты</a>
          </li>
          <li class="nav-item menu_nav log">
            <a class="nav-link" href="/SignIn">Login</a>
          </li>
          <li class="nav-item menu_nav">
            <a class="nav-link" href="/Signup">Sign up</a>
          </li>
        </ul>
      </div>
    </header>
  </template>

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

```

Содержимое компонента хедера


```

template>
<main>
  <div class="container">
    <div class="row">
      <div class="col-12 col-lg-5">
        <form class="form-fly" @submit.prevent="warn()" method="get">
          <div class="mb-3">
            <label class="label_text" for="from">Пункт отправления:</label>
            <input type="text" class="form-control" id="from" name="from" placeholder="Откуда" v-model="from">
          </div>
          <div class="mb-3">
            <div class="mb-3"></div>
            <div class="mb-3"></div>
          </div>
          <div class="mb-3">
            <select id="pass" class="form-select form-select-lg form-control"></select>
          </div>
          <div class="mb-3">
            <select id="class_pas" class="form-select form-select-lg form-control"></select>
          </div>
          <button v-if="Class" type="submit" class="btn btn-primary show">Смотреть билеты</button>
          <button v-else class="btn btn-primary show"><a href="/Answer">Смотреть билеты</a></button>
        </form>
      </div>
      <div class="col-12 col-lg-7">
        <h5 v-if="answer" mt-5>По вашему запросу ничего не найдено</h5>
      </div>
    </div>
  </main>
</template>

```

Содержимое компонента контента

Страны	Авиакомпании	Аэропорты
Россия	Аэрофлот	Шереметьево
Великобритания	Air France	Домодедово

Вид главной страницы

Аналогично были реализованы остальные страницы.

2.3 Реализация вывода билетов пользователя

Для начала нужно, чтобы пользователь вошел в систему. Для этого он вводит свои данные на страницу /SignIn, данные сверяются с бекендом- если по адресу <http://localhost:3000/person> есть такой пользователь, то происходит вход, в локальное хранилище сохраняется id пользователя и происходит переход на страницу /MyTicket. Если данные неверные, то выводится сообщение- данные для входа неверные, и предлагаются 2 ссылки – регистрация и восстановить пароль.

При переходе на страницу с моими билетами делается поиск по <http://localhost:3000/mytickets>, и если находятся билеты для пользователя с указанным id, то выводятся все его билеты, иначе пишется, что билетов у пользователя нет.

```
<script>
import axios from 'axios'
export default {
  name: 'SignIn',
  data: () => ({
    login: null,
    sign: '1',
    passw: null,
    form: null,
    true_form: null,
    ans2: null,
    wrong: null,
    new_url: ''
  }),
  methods: {
    enter: function () {
      axios.get('http://localhost:3000/person?login=${this.login}&passw=${this.passw}').then(response => {
        if (response.data.length !== 0) {
          this.form = response.data
          console.log(this.form[0].id)
          console.log('12345')
          localStorage.setItem('maha', JSON.stringify(this.form[0].id))
          this.sign = ''
          this.wrong = ''
          this.true_form = '1'
        } else {
          this.true = ''
          this.sign = ''
          this.wrong = '1'
        }
      })
    }
  }
}
```

Реализация SignIn скрипт

```
<template>
<main>
<div class="container">
<div class="col-lg-3"></div>
<div class="col-lg-6 col-12 login">
<div class="title"><a>Вход</a></div>
<form class="form-fly" @submit.prevent="enter()" method="get">
<div class="mb-3">
<input type="text" name="login" id="login" class="form-control" placeholder="Логин" v-model="login">
</div>
<div class="mb-3">
<input type="password" class="form-control" name="passw" id="passw" placeholder="Пароль" v-model="passw">
</div>
<div v-for="(f, index) in wrong" :key="index" class="answer">Неверный логин или пароль</div>
<button v-if="sign" type="submit" class="btn btn-primary show_btn mb-4" id="clear"><a>Войти</a></button>
<button v-if="true_form" :key="index" type="submit" class="btn btn-primary show_btn mb-4"><a href="/MyTicket">Войти</a></button>
<button v-if="wrong" :key="index" class="btn btn-primary show_btn mb-4"><a href="/SignIn">Войти</a></button>
<div v-for="(f, index) in wrong" :key="index" class="left"><a href="/SignUp">Регистрация</a></div>
<div v-for="(f, index) in wrong" :key="index" class="right"><a href="/Change">Забыли пароль?</a></div>
</form>
</div>
<div class="col-lg-3"></div>
</div>
</main>
</template>
```

Реализация формы SignIn

```

<template>
  <main>
    <div class="container">
      <div class="title"><a>Мои билеты</a></div>
      <h5 v-if="ans2" class="answer">У вас нет билетов</h5>
      <div class="row card-group form-fly">
        <div class="col-lg-4 col-md-6 col-sm-12 col-card" v-for="(f, index) in json2" :key="index">
          <div class="card">
            <img :src=getImgUrl(json2[index]) class="card-img-top" alt="">
            <div class="card-body">
              <h5 class="card-title">{{f.from}} - {{f.to}}</h5>
              <p class="card-text">Количество пассажиров - {{f.pass}}</p>
              <p class="card-text">Цена - {{f.price}} руб</p>
              <p class="card-text">Дата вылета - {{f.date}}</p>
              <p class="card-text">Время вылета - {{f.time}}</p>
              <button class="btn btn-primary little"><a href="#">Подробнее</a></button>
            </div>
          </div>
        </div>
      </div>
    </div>
  </main>
</template>

```

Вывод билетов MyTicket

```

<script>
  import axios from 'axios'
  export default {
    name: 'MyTicket',
    props: {
      json1: null,
      json2: null,
      ans2: null,
      to: null,
      from: null,
      date: null,
      time: null,
      pass: null,
      price: null
    },
    mounted () {
      this.json1 = JSON.parse(localStorage.getItem('maha'))
      console.log('qwe')
      axios.get('http://localhost:3000/mytickets?id_person={{this.json1}}').then(response => {
        if (response.data.length !== 0) {
          this.json2 = response.data
          console.log('12345')
          console.log(this.json2)
        } else {
          this.ans2 = '1'
          console.log(this.ans2)
        }
      })
    },
    methods: {
      getImgUrl (array) {
        console.log('...../public/image/' + array.to + '.jpg')
        return require('...../public/image/' + array.to + '.jpg')
      }
    }
  }
</script>

```

Реализация MyTicket скрипт

ЗАКЛЮЧЕНИЕ

В ходе выполнения этой работы я получила практический опыт использования Vue.js для создания страниц сайта с билетами. Также я разобралась, как для реализации своего приложения применить популярные библиотеки: Vue Router для навигации, Axios для отправки запросов на бэкенд, Bootstrap Vue для создания адаптивной верстки. Кроме того, я стала использовать больше инструментов Vue.js: директивы v-if/v-else.

СПИСОК ЛИТЕРАТУРЫ

1. Документация Vue.js [Электронный ресурс] <https://vuejs.org/v2/guide/>
2. Документация Vue Router [Электронный ресурс] <https://router.vuejs.org/guide/>
3. Документация BootstrapVue [Электронный ресурс] <https://bootstrap-vue.org/docs>
4. Документация Axios [Электронный ресурс] <https://axios-http.com/docs/intr>
5. Видео-урок на платформе YouTube
<https://www.youtube.com/watch?v=lwec8maPrrI&list=PLvTBThJr861wYlwBaaMy3tZUWpUvtJ9xE>