

Федеральное государственное автономное образовательное учреждение
высшего образования «Национальный исследовательский университет
ИТМО»

Лабораторная работа 3
«Разработка одностраничного веб-приложения (SPA) с использованием
фреймворка Vue.JS»

Выполнил:
Студент гр. К33402
Тарабанов Алексей Вячеславович
Проверил:
Добряков Д. И.

Санкт-Петербург

2024 год

Задача лабораторной работы 3:

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:

Исходя из задачи, все страницы были переписаны под новый стек.

Далее представлены основные из них:

- Файл Header.vue:

```
<template>
  <header :class="{ 'light-theme': !isDarkTheme, 'dark-theme': isDarkTheme }">
    <h1>Музыкальный плеер</h1>
    <nav>
      <router-link to="/">Главная</router-link>
      <router-link v-if="!isAuthenticated" to="/profile">Профиль</router-link>
      <button v-if="isAuthenticated" @click="logout"
class="button">Выход</button>
      <router-link v-if="!isAuthenticated" to="/login">Вход</router-link>
      <router-link v-if="!isAuthenticated"
to="/register">Регистрация</router-link>
      <button @click="toggleTheme" class="button">Тема</button>
    </nav>
  </header>
</template>

<script>
import { mapGetters } from 'vuex';
import { useAuth } from '../composables/useAuth';
import { useTheme } from '../composables/useTheme';
import { store } from '../store';
export default {
  props: {
    isDarkTheme: Boolean
  },
```

```

name: 'Header',
computed: {
  ...mapGetters(['isDarkTheme'])
},
components: {},
setup() {
  const { isAuthenticated, logout } = useAuth();
  const { isDarkTheme } = useTheme();
  const toggleTheme = () => {
    store.commit('toggleTheme');
  };
  return { isAuthenticated, logout, toggleTheme, isDarkTheme };
}
}
</script>

<style scoped>
header {
  background-color: #ff0000;
  padding: 10px 0;
  color: #949393;
  display: flex;
  justify-content: space-between;
  align-items: center;
}

h1 {
  margin-left: 20px;
}

nav a, button {
  margin-left: 10px;
  color: #fff;
  text-decoration: none;
  border: none;
  background: none;
  cursor: pointer;
}

nav a:hover, button:hover {
  text-decoration: underline;
}

button.button {
  margin-left: 10px;
  color: #fff;
  text-decoration: none;
  border: none;
  background: none;
  cursor: pointer;
}

```

```

.light-theme {
  --header-bg-color: #007bff;
  --header-text-color: #fff;
  --link-text-color: #fff;
  --button-text-color: #fff;
  --button-bg-color: #007bff;
}

.dark-theme {
  --header-bg-color: #343a40;
  --header-text-color: #fff;
  --link-text-color: #fff;
  --button-text-color: #fff;
  --button-bg-color: #343a40;
}
</style>

```

- Файл Home.vue:

```

<template>
  <transition name="fade">
    <div class="wrapper" :style="{ backgroundColor: bgColor }">
      <div :class="{ 'home': true, 'light-theme': !isDarkTheme, 'dark-theme':
isDarkTheme }">

        <div class="content-container">

          <h1 class="title-center">Топ 3 песни</h1>

          <div class="top-songs">
            <router-link v-if="isAuthenticated" to="/profile"
class="profile-link">Profile</router-link>
            <transition-group name="fade" tag="div">
              <div v-for="(song, index) in topSongs" :key="index"
class="song-card">
                
                <h2 class="song-title">{{ song.replace("AC-DC - ", "") }}</h2>
                <audio :src="'/music/${encodeURIComponent(song)}.mp3'"
controls></audio>
              </div>
            </transition-group>
          </div>

          <h2 class="title-center">Все песни</h2>
          <div class="all-songs">
            <div v-for="(song, index) in songs" :key="index"
class="song-card">
              
              <h2 class="song-title">{{ song.title }}</h2>
              <audio :src="song.audio" controls></audio>
            </div>
          </div>
        </div>
      </div>
    </transition>
  </template>

```

```

        </div>
      </div>
    </div>
  </div>
</transition>
</template>

<script>
import { mapGetters } from "vuex";

export default {
  name: 'Home',
  data() {
    return {
      topSongs: [],
      songs: [
        { id: 1, title: 'Moneytalks ', artist: 'AC-DC', audio: '/music/AC-DC - Moneytalks.mp3' },
        { id: 2, title: 'Back In Black ', artist: 'AC-DC', audio: '/music/AC-DC - Back In Black.mp3' },
        { id: 3, title: 'House Of Jazz', artist: 'AC-DC', audio: '/music/AC-DC - House Of Jazz.mp3' },
        { id: 4, title: 'Mistress For Christmas', artist: 'AC-DC', audio: '/music/AC-DC - Mistress For Christmas.mp3' }
      ]
    };
  },
  methods: {
    async fetchTopSongs() {
      try {
        const response = await fetch('http://localhost:3000/api/topSongs');
        if (!response.ok) {
          throw new Error('Error fetching top songs');
        }
        this.topSongs = (await response.text()).split('#').map(song => song.trim()).filter(Boolean);
      } catch (error) {
        console.error(error);
      }
    },
    startMarquee(event) {
      const element = event.target.querySelector('.song-title');
      element.classList.add('marquee');
    },
    stopMarquee(event) {
      const element = event.target.querySelector('.song-title');
      element.classList.remove('marquee');
    }
  },
  computed: {
    ...mapGetters(['isDarkTheme', 'isAuthenticated']),
    bgColor() {

```

```

        return this.isDarkTheme ? "var(--dark-bg-color)" :
"var(--light-bg-color)";
    }
},
    async mounted() {
        await this.fetchTopSongs();
    }
};
</script>

<style scoped>

.light-theme {
    --light-bg-color: #f2f2f2;
    --bg-color: #f2f2f2;
    --content-bg-color: #fff;
    --header-bg-color: #007bff;
    --header-text-color: #fff;
    --link-text-color: #fff;
    --button-text-color: #fff;
    --button-bg-color: #007bff;
    --song-title-color: #333;
}

.dark-theme {
    --dark-bg-color: #343a40;
    --bg-color: #343a40;
    --content-bg-color: #333;
    --header-bg-color: #343a40;
    --header-text-color: #fff;
    --link-text-color: #fff;
    --button-text-color: #fff;
    --button-bg-color: #333;
    --song-title-color: #fff;
}

.wrapper {
    display: flex;
    justify-content: center;
    background-image: url('Fon.jpg');
    background-size: cover;
}

.dark-theme .wrapper {
    background-image: url('Fon.jpg');
}

.content-container {
    width: 87.7%;
    background-color: var(--content-bg-color);
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
    padding: 20px;
    min-height: 100vh;
}

```

```
}

.home {
  display: flex;
  flex-direction: column;
  align-items: center;
}

.title-center {
  text-align: center;
}

.profile-link {
  margin-top: 20px;
  padding: 10px 20px;
  border: none;
  border-radius: 5px;
  background-color: #007bff;
  color: #fff;
  font-weight: bold;
  text-decoration: none;
  cursor: pointer;
}

.profile-link:hover {
  background-color: #0056b3;
}

.song-info h2 {
  margin: 0;
  font-size: 18px;
  font-weight: bold;
  color: #333;
}

audio {
  margin-top: 10px;
  width: 550px;
}

audio {
  margin-top: 10px;
  width: 850px;
}

.audio-player-timeline input[type="range"] {
  width: 150%;
  height: 5px;
  background-color: #ccc;
  border: none;
  border-radius: 5px;
  appearance: none;
  outline: none;
}
```

```
}

.audio-player-volume-slider input[type="range"] {
  appearance: none;
  background-color: #ccc;
  width: 150%;
  height: 5px;
  border-radius: 5px;
}

.audio-player-volume-slider input[type="range"]::-webkit-slider-thumb {
  appearance: none;
  width: 12px;
  height: 12px;
  border-radius: 50%;
  background-color: #007bff;
}

.audio-player-mute-button {
  background-color: transparent;
  border: none;
  cursor: pointer;
}

.profile-link {
  margin-top: 20px;
  padding: 10px 20px;
  border: none;
  border-radius: 5px;
  background-color: #007bff;
  color: #fff;
  font-weight: bold;
  text-decoration: none;
  cursor: pointer;
}

.profile-link:hover {
  background-color: #0056b3;
}

.song-card {
  width: 70%;
  margin: auto;
  display: flex;
  align-items: center;
  padding: 10px;
  border: 1px solid #ccc;
  border-radius: 10px;
}

.song-thumbnail {
  width: 50px;
  height: 50px;
```



```

margin-right: 10px;
border-radius: 5px;
}

.song-title {
display: inline-block;
width: 100%;
white-space: nowrap;
text-overflow: ellipsis;
transition: transform 1s ease-in-out;
}

.song-title:hover {
overflow: auto;
animation-play-state: paused;
}

@keyframes marquee {
from { transform: translateX(100%); }
to { transform: translateX(-100%); }
}
</style>

```

Файл Login.vue:

```

<template>
  <transition name="fade">
    <div class="wrapper">
      <div :class="{ 'login': true, 'light-theme': !isDarkTheme, 'dark-theme':
isDarkTheme }">
        <div class="content-container">
          <h1 :class="{ 'text-light': isDarkTheme, 'text-dark': !isDarkTheme
}">Вход</h1>
          <form @submit.prevent="login" class="form">
            <input type="email" v-model="email" placeholder="Почта" required
:class="{ 'input-light': isDarkTheme, 'input-dark': !isDarkTheme }">
            <input type="password" v-model="password" placeholder="Пароль"
required :class="{ 'input-light': isDarkTheme, 'input-dark': !isDarkTheme }">
            <button type="submit" class="button" :class="{ 'button-light':
isDarkTheme, 'button-dark': !isDarkTheme }">Вход</button>
          </form>
          <router-link to="/register" class="link" :class="{ 'link-light':
isDarkTheme, 'link-dark': !isDarkTheme }">Регистрация</router-link>
        </div>
      </div>
    </div>
  </transition>
</template>

<script>
import { ref } from 'vue';
import { useAuth } from '../composables/useAuth';
import { useTheme } from '@composables/useTheme';

```

```

import {mapGetters} from "vuex";

export default {
  props: {
    isDarkTheme: Boolean // объявляем пропс isDarkTheme
  },
  name: 'Login',
  computed: {
    ...mapGetters(['isDarkTheme'])
  },
  components: {},
  setup() {
    const { login } = useAuth();
    const email = ref('');
    const password = ref('');
    const { isDarkTheme } = useTheme();

    return { email, password, login, isDarkTheme };
  }
}
</script>

<style scoped>
/* Styles for Login.vue */
.wrapper {
  display: flex;
  justify-content: center;
  background-color: #f2f2f2;
}

.content-container {
  width: 100%;
  background-color: #fff;
  box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
  padding: 10px;
  min-height: 100vh;
}

.login {
  display: flex;
  flex-direction: column;
  align-items: center;
  margin-top: 50px;
}

.form {
  width: 300px;
  margin-top: 20px;
}

.input-light,
.input-dark {
  width: 100%;

```

```
padding: 10px;
margin-bottom: 10px;
border: none;
border-radius: 5px;
box-shadow: 0 0 5px rgba(0, 0, 0, 0.1);
}

.button-light,
.button-dark {
width: 100%;
padding: 10px;
border: none;
border-radius: 5px;
cursor: pointer;
transition: background-color 0.3s;
}

.button-light {
background-color: #1DB954;
color: white;
}

.button-dark {
background-color: #333;
color: white;
}

.button-light:hover {
background-color: #168f3f;
}

.button-dark:hover {
background-color: #555;
}

.link-light,
.link-dark {
margin-top: 10px;
color: #1DB954;
text-decoration: none;
transition: color 0.3s;
}

.link-light:hover,
.link-dark:hover {
color: #168f3f;
}

</style>
```

- Файл Profile.vue:

- <template>

```

<div class="wrapper">
  <div :class="{ 'profile': true, 'light-theme': !isDarkTheme,
'dark-theme': isDarkTheme }">
    <div class="content-container" style="width: 150%;">
      <h1 :class="{ 'text-light': isDarkTheme, 'text-dark':
!isDarkTheme }">Профиль</h1>
      <div :class="{ 'text-light': isDarkTheme, 'text-dark':
!isDarkTheme }">
        <p>Имя: {{ user.name }}</p>
        <p>Почта: {{ user.email }}</p>
      </div>
      <h2 :class="{ 'text-light': isDarkTheme, 'text-dark':
!isDarkTheme }">Плейлисты</h2>
      <div class="playlists">
        <div v-for="(playlist, index) in playlists" :key="index"
class="playlist-item" :class="{ 'playlist-item-light': !isDarkTheme,
'playlist-item-dark': isDarkTheme }">
          <p :class="{ 'text-light': isDarkTheme, 'text-dark':
!isDarkTheme }">{{ playlist.name }}</p>
          <button @click="deletePlaylist(index)" class="button"
:class="{ 'button-light': isDarkTheme, 'button-dark': !isDarkTheme
}">Удалить</button>
        </div>
        <button @click="addPlaylist" class="add-button" :class="{
'button-light': isDarkTheme, 'button-dark': !isDarkTheme }">Добавить
плейлист</button>
      </div>
    </div>
  </div>
</template>

<script>
import { useTheme } from "@composables/useTheme.js";

export default {
  name: 'Профиль',
  setup() {
    const { isDarkTheme } = useTheme();
    return { isDarkTheme };
  },
  data() {
    return {
      user: { name: 'Алексей;', email: 'tarabanov@mail.ru' },
      playlists: [
        { id: 1, name: 'Избранное' },
      ]
    };
  },
  methods: {

```

```

    addPlaylist() {
      const newPlaylist = { id: this.playlists.length + 1, name:
`Плейлист ${this.playlists.length + 1}` };
      this.playlists.push(newPlaylist);
    },
    deletePlaylist(index) {
      this.playlists.splice(index, 1);
    }
  }
};
</script>

```

```

<style scoped>
/* Styles for Profile.vue */

.wrapper {
  display: flex;
  justify-content: center;
}

.content-container {
  width: 87.7%;
  background-color: var(--content-bg-color);
  box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
  padding: 20px;
  min-height: 100vh;
}

.profile {
  display: flex;
  width: 60%;
  flex-direction: revert;
  align-items: center;
  margin-top: 50px;
}

.profile h1 {
  font-size: 24px;
  margin-bottom: 20px;
}

.profile p {
  margin-bottom: 10px;
}

.playlists {
  margin-top: 20px;
}

```

```
.playlist-item {
  display: grid;
  align-items: flex-start;
  justify-content: flex-start;
  margin-bottom: 10px;
  background-color: rgba(255, 255, 255, 0); /* Прозрачный цвет фона */
  border-radius: 5px;
  padding: 10px;
}

.playlist-item-light {
  background-color: #f9f9f9;
}

.playlist-item-dark {
  background-color: #333;
}

.add-button, .button {
  padding: 8px 16px;
  border: none;
  border-radius: 4px;
  cursor: pointer;
  transition: background-color 0.3s ease;
}

.button-light {
  background-color: #1DB954;
  color: white;
}

.button-dark {
  background-color: #333;
  color: white;
}

.button-light:hover {
  background-color: #168f3f;
}

.button-dark:hover {
  background-color: #555;
}

.text-light {
  color: white;
}

.text-dark {
```

```

    color: black;
  }

  .light-theme {
    background-color: #f2f2f2;
  }

  .dark-theme {
    background-color: #343a40;
  }
</style>

```

- Файл Register.vue:

```

<template>
<div class="wrapper">
  <div :class="{ 'register': true, 'light-theme': !isDarkTheme,
'dark-theme': isDarkTheme }">
    <div class="content-container">
      <h1 :class="{ 'text-light': isDarkTheme, 'text-dark':
!isDarkTheme }">Регистрация</h1>
      <form @submit.prevent="register" class="form">
        <input type="text" v-model="name" placeholder="Имя" required
:class="{ 'input-light': isDarkTheme, 'input-dark': !isDarkTheme }">
        <input type="email" v-model="email" placeholder="Почта"
required :class="{ 'input-light': isDarkTheme, 'input-dark':
!isDarkTheme }">
        <input type="password" v-model="password"
placeholder="Пароль" required :class="{ 'input-light': isDarkTheme,
'input-dark': !isDarkTheme }">
        <button type="submit" class="button" :class="{
'button-light': isDarkTheme, 'button-dark': !isDarkTheme
}">Регистрация</button>
      </form>
      <router-link to="/login" class="link" :class="{ 'link-light':
isDarkTheme, 'link-dark': !isDarkTheme }">Вход</router-link>
    </div>
  </div>
</div>
</template>

<script>
import { ref } from 'vue';
import { useAuth } from '../composables/useAuth';
import { useTheme } from '@composables/useTheme';
import { mapGetters } from "vuex";

export default {
  props: {
    isDarkTheme: Boolean
  }
}

```

```

    },
    name: 'Register',
    computed: {
      ...mapGetters(['isDarkTheme'])
    },
    setup() {
      const { register } = useAuth();
      const name = ref('');
      const email = ref('');
      const password = ref('');
      const { isDarkTheme } = useTheme();

      return { name, email, password, register, isDarkTheme };
    }
  }
</script>

<style scoped>
.wrapper {
  display: flex;
  justify-content: center;
  background-color: #f2f2f2;
}

.content-container {
  width: 100%;
  background-color: #fff;
  box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
  padding: 10px;
  min-height: 100vh;
}

.register {
  display: flex;
  flex-direction: column;
  align-items: center;
  justify-content: center;
  height: 100vh;
}

.form {
  width: 300px;
  margin-top: 20px;
}

.input-light, .input-dark {
  width: 100%;
  padding: 10px;
  margin-bottom: 10px;

```



```

• border: none;
• border-radius: 5px;
• box-shadow: 0 0 5px rgba(0, 0, 0, 0.1);
• }
•
• .button-light, .button-dark {
•   width: 100%;
•   padding: 10px;
•   border: none;
•   border-radius: 5px;
•   cursor: pointer;
•   transition: background-color 0.3s;
• }
•
• .button-light {
•   background-color: #1DB954;
•   color: white;
• }
•
• .button-dark {
•   background-color: #333;
•   color: white;
• }
•
• .button-light:hover {
•   background-color: #168f3f;
• }
•
• .button-dark:hover {
•   background-color: #555;
• }
•
• .link-light, .link-dark {
•   margin-top: 10px;
•   color: #1DB954;
•   text-decoration: none;
•   transition: color 0.3s;
• }
•
• .link-light:hover, .link-dark:hover {
•   color: #168f3f;
• }
• </style>

```

Итог выполненной работы:

Скриншоты страниц в браузере:

- Страница “Регистрация”:

← → ↻ 🛡 📄 localhost:5173/register ☆ 📧 N 📄

Музыкальный плеер Главная Профиль Вход Регистрация Тема

Регистрация

Имя

Aleksey

••••••••••

Регистрация

[Вход](#)

- Страница “Вход в профиль”:

The screenshot shows a web browser window with the address bar displaying 'localhost:5173/login'. The page has a red header with the title 'Музыкальный плеер' and navigation links: 'Главная', 'Профиль', 'Вход', 'Регистрация', and 'Тема'. The main content area is light gray and contains a white login form titled 'Вход'. The form has two yellow input fields: the first contains the text 'Aleksey' and the second contains ten black dots representing a password. Below the password field is a dark blue button labeled 'Вход'. Underneath the button is a green link labeled 'Регистрация'.

Музыкальный плеер

Главная Профиль Вход Регистрация Тема

Вход

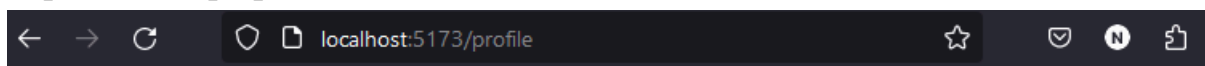
Aleksey

••••••••••

Вход

[Регистрация](#)

- Страница “Профиль”:



Музыкальный плеер

[Главная](#) [Профиль](#) [Вход](#) [Регистрация](#)

[Тема](#)

Профиль

Имя: Алексей;

Почта: tarabanov@mail.ru

Плейлисты

Избранное

Удалить

Плейлист 2

Удалить

Добавить плейлист

- Главная страница плеера:

Музыкальный плеер [Главная](#) [Профиль](#) [Вход](#) [Регистрация](#) [Тема](#)

Топ 3 песни

Back In Black 0:00 / 4:08

Moneytalks 0:00 / 4:19

Rock or Bust 0:00 / 3:04

Все песни

Moneytalks 0:00 / 4:19

Back In Black 0:00 / 4:08

House Of Jazz 0:00 / 3:54

Mistress For Christmas 0:00 / 4:00

Вывод:

Третья лабораторная работа была направлена на миграцию ранее написанного сайта на фреймворк Vue.js с соблюдением минимальных требований, включая подключение роутера, работу с внешним API, разумное деление на компоненты и использование композиционных функций, а именно - composables.

В ходе выполнения работы был использован Vue Router для управления маршрутизацией в приложении, что в свою очередь позволило организовать навигацию между различными страницами. Для работы с внешним API был реализован метод fetchTopSongs, который получает топовые песни с сервера - обеспечение динамической загрузки данных с удаленного источника.

Также, весь проект был разделен на компоненты с целью повышения его масштабируемости и обеспечения лучшей организации кода; каждый компонент отвечает за конкретный функциональный блок и может быть переиспользован.

Был использован composable useTheme, который позволяет управлять темой и предоставляет доступ к текущей теме через компоненты, для управления темой приложения.

Подводя общий вывод по проделанной работе, можно отметить, что проект соответствует требованиям и обеспечивает хорошую основу для дальнейшего развития, а использование Vue.js значительно улучшает организацию кода, делает приложение более масштабируемым и поддерживаемым.