

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Фронтэнд-разработка

Отчет

Лабораторная работа №3

Выполнил:

Гнеушев Владислав

К3439

Проверил:

Добряков Д. И.

Санкт-Петербург

2025 г.

Задача

Мигрировать ранее разработанное приложение (в рамках ЛР1 и ЛР2) на фреймворк Vue.JS.

Каким требованиям ваше приложение должно соответствовать:

- 1) Должен быть подключён роутер
- 2) Должна быть реализована работа с внешним API (желательно посредством axios)
- 3) Разумное деление на компоненты (продемонстрируйте понимание компонентного подхода)
- 4) Использование composable для выделения повторяющегося функционала в отдельные файлы

Выбранный вариант - Платформа для фитнес-тренировок и здоровья

Ход работы

1. Подключение роутера

Добавлен роутер с путями, прикрепленными к разным View.

```

    {
      path: '/login',
      name: 'login',
      component: LoginView
    },
    {
      path: '/register',
      name: 'register',
      component: RegisterView
    },
    {
      path: '/profile',
      name: 'profile',
      component: ProfileView
    }
  ];

  const router = createRouter({
    history: createWebHistory(),
    routes
  });

  export default router;

```

Рисунок 1 – Роутер

2. Добавлено использование axios вместо fetch

```

const apiClient = axios.create({
  baseURL: API_BASE_URL,
  headers: {
    'Content-Type': 'application/json'
  }
});

```

Рисунок 2 – API клиент axios

```

export function useApi() {
  const { currentUser } = useAuth();

  async function fetchWorkouts(filters = {}) {
    try {
      const { data } = await apiClient.get('workouts', { params: filters });
      return data;
    } catch (error) {
      throw new Error('Не удалось получить список тренировок');
    }
  }
}

```

Рисунок 3 – Использование API клиента

3. Добавлено разделение на компоненты

▼ AppFooter.vue X

src > components > ▼ AppFooter.vue > {} template

```

1 <template>
2   <footer class="bg-dark text-white text-center py-3">
3     <div class="container">
4       <p>Good fitness</p>
5       <p>Контакты: info@goodfitness.ru | Телефон: +7 (999) 123-45-67</p>
6     </div>
7   </footer>
8 </template>
9

```

Рисунок 4 – Компонент футер

▼ AppFooter.vue ▼ BlogCard.vue X

src > components > ▼ BlogCard.vue > ...

```

1 <template>
2   <article class="card blog-card">
3     <div class="card-body">
4       <h3 class="card-title h5">{{ post.title }}</h3>
5       <p class="card-text">
6         <strong>Категория:</strong> {{ categoryNames[post.category] || '-' }}<br>
7         <strong>Дата:</strong> {{ post.date }}
8       </p>
9       <router-link :to="/blog/${post.id}" class="btn btn-primary">Читать</router-link>
10    </div>
11  </article>
12 </template>
13
14 <script setup>
15 import { categoryNames } from '../composables/useFilters';
16
17 defineProps({
18   post: {
19     type: Object,
20     required: true
21   }
22 });
23 </script>
24

```

Рисунок 5 – Компонент BlogCard

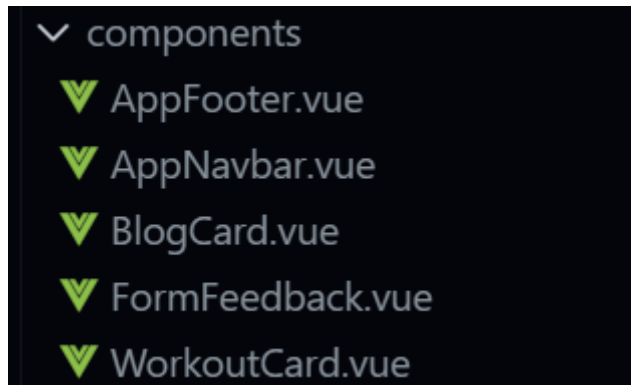


Рисунок 6 – Все компоненты

4. Разделение на composables

Общая логика и функции вынесены в composables.

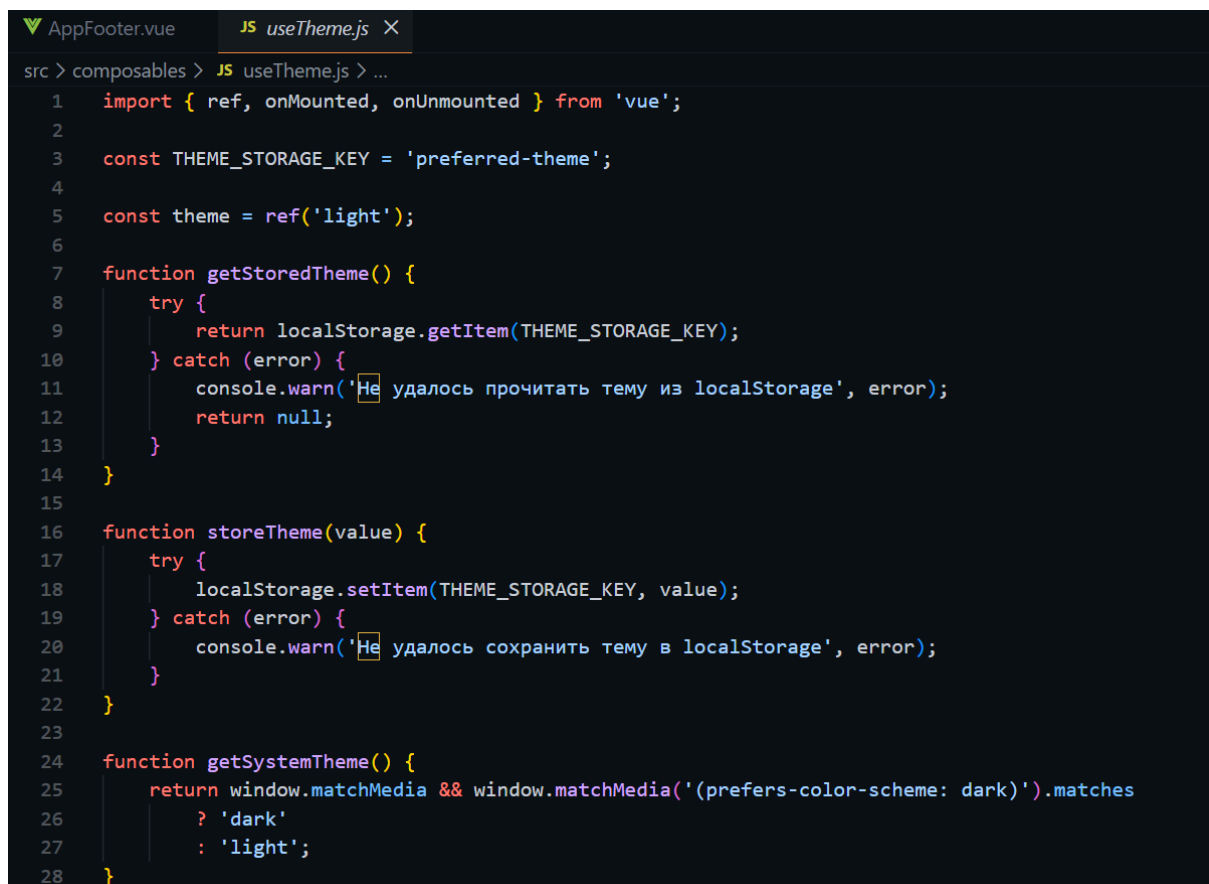


Рисунок 7 – Composable для управления темой

Вывод

В результате выполнения лабораторной работы проект был мигрирован на фреймворк Vue. В ходе работы был добавлен роутер с путями, прикрепленными к View, добавлено использование axios, выделены компоненты.