

**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

```

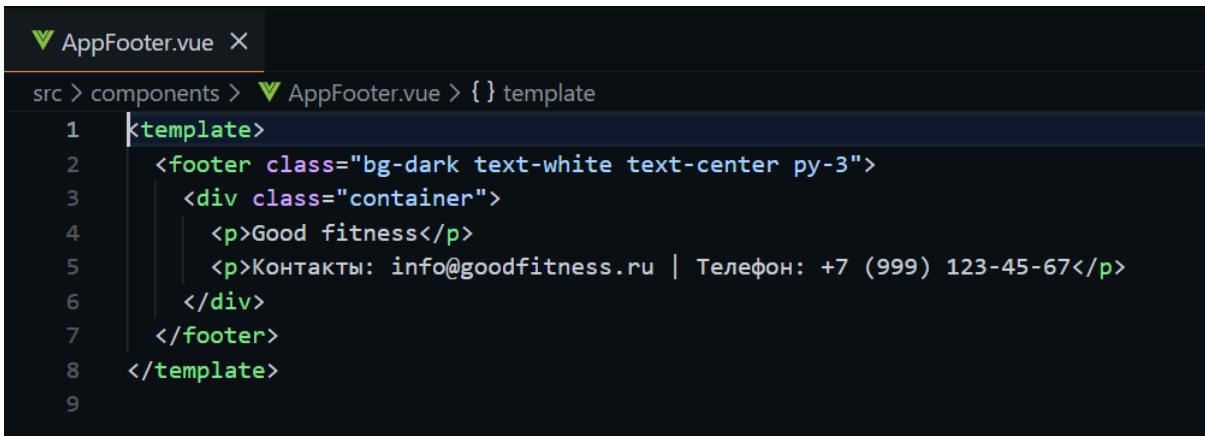
export function useApi() {
  const { currentUser } = useAuth();

  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



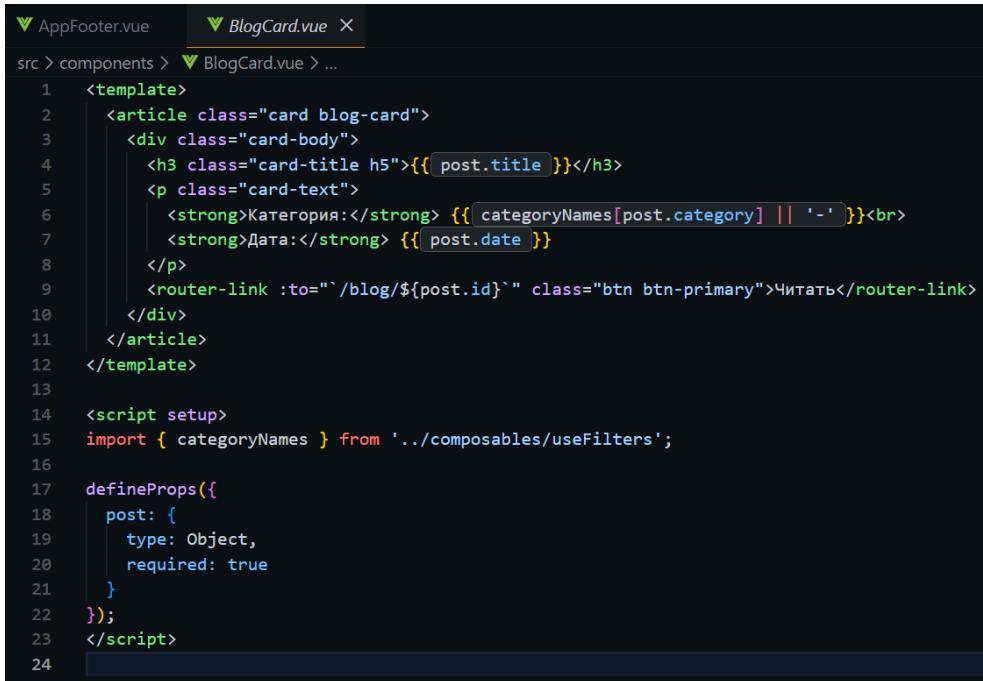
The screenshot shows the code for the `AppFooter.vue` component. It contains a single template block with the following content:

```

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

```

Рисунок 4 – Компонент футер



The screenshot shows the code for the `BlogCard.vue` component. It includes both a template block and a `<script setup>` block. The template block defines an article card with various sections like title, text, category, date, and a read more link. The `<script setup>` block imports a variable and defines props.

```

<template>
  <article class="card blog-card">
    <div class="card-body">
      <h3 class="card-title h5">{{ post.title }}</h3>
      <p class="card-text">
        <strong>Категория:</strong> {{ categoryNames[post.category] || '-' }}<br>
        <strong>Дата:</strong> {{ post.date }}
      </p>
      <a href="/blog/${post.id}" class="btn btn-primary">Читать</a>
    </div>
  </article>
</template>

<script setup>
import { categoryNames } from '../composables/useFilters';

```

Рисунок 5 – Компонент BlogCard

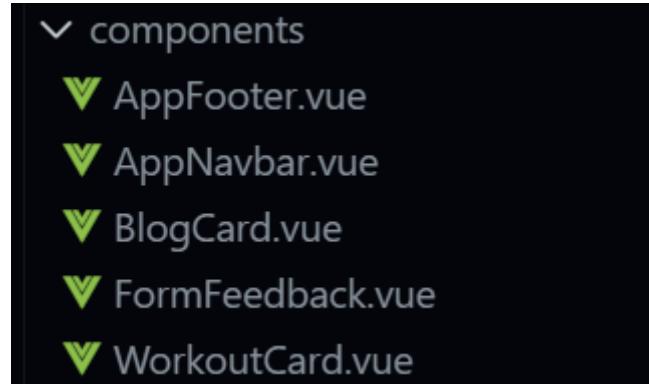


Рисунок 6 – Все компоненты

#### 4. Разделение на composable

Общая логика и функции вынесены в composable.

```
AppFooter.vue      JS useTheme.js X
src > composable > JS useTheme.js > ...
1  import { ref, onMounted, onUnmounted } from 'vue';
2
3  const THEME_STORAGE_KEY = 'preferred-theme';
4
5  const theme = ref('light');
6
7  function getStoredTheme() {
8    try {
9      return localStorage.getItem(THEME_STORAGE_KEY);
10   } catch (error) {
11     console.warn('Не удалось прочитать тему из localStorage', error);
12     return null;
13   }
14 }
15
16 function storeTheme(value) {
17   try {
18     localStorage.setItem(THEME_STORAGE_KEY, value);
19   } catch (error) {
20     console.warn('Не удалось сохранить тему в localStorage', error);
21   }
22 }
23
24 function getSystemTheme() {
25   return window.matchMedia && window.matchMedia('(prefers-color-scheme: dark)').matches
26     ? 'dark'
27     : 'light';
28 }
```

The code editor shows a file named 'useTheme.js' with the following content:

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.