

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Фронт-энд разработка

Отчет

Лабораторная работа #3

Выполнил:
Ребров Сергей

Группа:
К3439

Проверил:
Добряков Д. И.

Санкт-Петербург

2026 г.

Задача

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

Ход работы

В ходе выполнения работы были выделены компоненты Header, Footer и VacancyCard. В Header вынесена шапка сайта:

```
<template>
  <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
    <div class="container-fluid">
      <router-link class="navbar-brand d-flex align-items-center" :to="{ name: 'Search' }">
        <svg class="icon me-2"><use xlink:href="../../assets/imgs/sprite.svg#logo"></use></svg>
        <span>JobFinder</span>
      </router-link>
      <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#nav"
        aria-controls="nav" aria-expanded="false" aria-label="Toggle navigation">
        <span class="navbar-toggler-icon"></span>
      </button>

      <div class="collapse navbar-collapse" id="nav">
        <ul class="navbar-nav ms-auto">
          <li class="nav-item">
            <router-link class="nav-link" :to="{ name: 'Search' }">Поиск</router-link>
          </li>
          <li class="nav-item">
            <router-link class="nav-link" :to="{ name: 'Profile' }">Аккаунт</router-link>
          </li>
          <li class="nav-item">
            <a class="nav-link" href="#" @click.prevent="onLogout">Выйти</a>
          </li>
        </ul>
      </div>
    </div>
  </nav>
</template>
```

Далее были перенесены все страницы. Пример с использованием компонентов (Search):

```
<template>
  <div>
    <div class="filter-section mb-4">
      <div class="row g-2">
        <div class="col">
          <input v-model="q" @keyup.enter="search" type="text" class="form-control" placeholder="Введите текст" />
        </div>
        <div class="col-auto">
          <button class="btn btn-primary" @click="search">
            <svg class="icon"><use xlink:href="../assets/imgs/sprite.svg#search"></use></svg>
          </button>
        </div>
      </div>
    </div>
  </div>

  <div v-if="loading">Загрузка...</div>
  <div v-if="error">Ошибка: {{ error.message || error }}</div>

  <div id="vacancy-list">
    <VacancyCard v-for="v in vacancies" :key="v.id" :vacancy="v" @delete="onDelete" />
    <div v-if="vacancies.length === 0" class="text-muted">Вакансий не найдено</div>
  </div>
</div>
</template>
```

Также был подключен роутер:

```
import { createRouter, createWebHistory } from 'vue-router'
import Search from '@views/Search.vue'
import Login from '@views/Login.vue'
import Register from '@views/Register.vue'
import Profile from '@views/Profile.vue'
import Vacancy from '@views/Vacancy.vue'
import { getToken } from '@composables/useAuth'

const routes = [
  { path: '/', name: 'Home', component: Login },
  { path: '/search', name: 'Search', component: Search },
  { path: '/login', name: 'Login', component: Login },
  { path: '/register', name: 'Register', component: Register },
  { path: '/profile', name: 'Profile', component: Profile, meta: { requiresAuth: true } },
  { path: '/vacancy/:id', name: 'Vacancy', component: Vacancy, props: true },
]

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

router.beforeEach((to, from, next) => {
  if (to.meta.requiresAuth && !getToken()) {
    return next({ name: 'Login' })
  }
  next()
})

export default router
```

Пример запроса через axios для работы с API (регистрация пользователя):

```
async function onSubmit() {
  if (form.password !== form.confirmPassword) {
    alert('Пароли не совпадают')
    return
  }

  try {
    const response = await axios({
      method: 'post',
      url: `${import.meta.env.VITE_API_URL}/register`,
      headers: { 'Content-Type': 'application/json' },
      data: JSON.stringify({
        email: form.email,
        password: form.password,
        name: form.name,
        role: form.role
      })
    })

    localStorage.setItem('token', response.data.accessToken)
    localStorage.setItem('user', JSON.stringify(response.data.user))

    router.push({ name: 'Search' })
  } catch (e) {
    alert('Ошибка регистрации: ' + e.response.data)
  }
}
```

Вывод

В ходе работы приложение было полностью перенесено на Vue.js: реализована маршрутизация с помощью Vue Router, организован обмен данными с внешним API через axios, а интерфейс разделен на отдельные компоненты, демонстрируя принцип компонентного подхода и поддерживая чистую и удобную структуру кода.