

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Фронт-энд разработка

**ОТЧЕТ
ПО ДОМАШНЕЙ РАБОТЕ №5**

Выполнила: Лапшина Екатерина

Группа: К3440

Проверил: Добряков Д. И.

Санкт-Петербург

2025 г.

Задание

В рамках данной работы необходимо изучить основные команды пакетного менеджера NPM и научиться стартовать проект на Vue.js.

Проект: Сервис "АрендаНедвижимости" – приложение для поиска и аренды жилой недвижимости.

Ход работы

1. Установка и инициализация

Для создания проекта использовался инструмент Vite как более современная и быстрая альтернатива Vue CLI. Были установлены следующие зависимости: vue (JavaScript-фреймворк для создания пользовательских интерфейсов) и bootstrap (CSS-фреймворк для стилизации).

Команды для создания и инициализации проекта представлены в листинге 1.

Листинг 1 – Команды для создания проекта

```
npm create vite@latest hw5 -- --template vue
cd hw5
npm install
npm install bootstrap
```

1. Структура проекта

Проект организован с использованием компонентного подхода. Основные файлы включают точку входа приложения, корневой компонент и набор переиспользуемых компонентов.

Точка входа приложения (src/main.js)

Файл main.js инициализирует приложение Vue и подключает необходимые стили Bootstrap. Код точки входа представлен в листинге 2.

Листинг 2 – Инициализация приложения Vue

```
import { createApp } from 'vue'
import 'bootstrap/dist/css/bootstrap.min.css' import
'bootstrap'
import './style.css'
import App from './App.vue'

createApp(App).mount('#app')
```

Корневой компонент (src/App.vue)

Компонент App.vue собирает страницу из трех основных компонентов: Header (шапка сайта), SearchPage (основной контент с поиском) и Footer (подвал сайта). Структура корневого компонента показана в листинге 3.

Листинг 3 – Корневой компонент приложения

```
<script setup>
import Header from './components/Header.vue'
import Footer from './components/Footer.vue'
import SearchPage from './components/SearchPage.vue'
</script>

<template>
  <Header />
  <SearchPage />
  <Footer />
</template>
```

2. Компоненты

Компонент Header (src/components/Header.vue)

Компонент Header реализует шапку сайта с навигацией. Включает логотип, навигационное меню с пунктами "Поиск", "Мой профиль" и "Сообщения", а также кнопки входа и регистрации. Используются классы Bootstrap для стилизации. Структура компонента представлена в листинге 4.

Листинг 4 – Компонент шапки сайта

```
<template>
  <header class="p-3 text-bg-dark mb-3">
    <div class="container">
      <div class="d-flex flex-wrap align-items-center justify-content-center
        justify-content-lg-start">
        <a href="#" class="d-flex align-items-center mb-2 b-lg-0 text-white
          text-decoration-none">
          <!-- SVG Icon -->
          <span class="fs-4">Аренда</span>
```

```

</a>

<ul class="nav col-12 col-lg-auto me-lg-auto mb-2
      justify-content-center mb-md-0">
  <li><a href="#" class="nav-link px-2 text-secondary">
    Поиск</a></li>
  <li><a href="#" class="nav-link px-2 text-white">
    Мой профиль</a></li>
  <li><a href="#" class="nav-link px-2 text-white">
    Сообщения</a></li>
</ul>

<div class="text-end">
  <a href="#" class="btn btn-outline-light me-2">Вход</a>
  <a href="#" class="btn btn-warning">Регистрация</a>
</div>
</div>
</div>
</header>
</template>

```

Компонент PropertyCard (src/components/PropertyCard.vue)

Компонент PropertyCard предназначен для отображения карточки отдельного объекта недвижимости. Принимает данные через props и отображает изображение, название, тип недвижимости, местоположение и цену аренды. Также включает вспомогательную функцию для перевода типа недвижимости на русский язык. Реализация компонента показана в листинге 5.

Листинг 5 – Компонент карточки недвижимости

```

<template>
  <div class="col">
    <div class="card shadow-sm h-100">
      
      <div class="card-body">
        <h5 class="card-title">{{ property.title }}</h5>
        <p class="card-text text-muted">
          {{ typeToRussian(property.type) }}</p>
        <p class="card-text text-muted">{{ property.location }}</p>
        <p class="card-text fw-bold">
          {{ property.price.toLocaleString() }} ₱ / мес</p>
        <div class="d-flex justify-content-between align-items-center mt-3">
          <div class="btn-group">
            <button class="btn btn-sm btn-outline-secondary">
              Подробнее</button>
          </div>
        </div>
      </div>
    </div>
  </div>

```

```

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

<script setup>
defineProps({
  property: {
    type: Object,
    required: true
  }
})

function typeToRussian(type) {
  const types = {
    'apartment': 'Квартира',
    'house': 'Дом',
    'studio': 'Студия'
  }
  return types[type] || type
}
</script>

```

Компонент SearchPage (src/components/SearchPage.vue)

Компонент SearchPage является основным компонентом логики поиска. Содержит блок фильтров (выбор типа недвижимости, минимальной и максимальной цены) и список карточек объектов недвижимости. Использует ref для реактивности данных. При загрузке компонента отображается индикатор загрузки. Реализация компонента представлена в листинге 6.

Листинг 6 – Компонент страницы поиска

```

<template>
  <main class="container">
    <div class="row mb-3">
      <div class="col-md-12">
        <div class="card p-4">
          <h3>Фильтры поиска</h3>
          <form @submit.prevent="applyFilters" class="row g-3">
            <div class="col-md-3">
              <label for="inputType" class="form-label"> Тип
                недвижимости</label>
              <select id="inputType"
                class="form-select"
                v-model="filters.type">
                <option value="">Все типы</option>
                <option value="apartment">Квартира</option>
                <option value="studio">Студия</option>
                <option value="house">Дом</option>

```

```

        </select>
      </div>
      <div class="col-md-3">
        <label for="inputMinPrice" class="form-label">
          Мин. Цена (₽)</label>
        <input type="number" class="form-control"
          id="inputMinPrice"
          v-model.number="filters.minPrice">
      </div>
      <div class="col-md-3">
        <label for="inputMaxPrice" class="form-label">
          Макс. Цена (₽)</label>
        <input type="number" class="form-control"
          id="inputMaxPrice"
          v-model.number="filters.maxPrice">
      </div>
      <div class="col-md-3 d-flex align-items-end">
        <button type="submit" class="btn btn-primary w-100">
          Найти</button>
      </div>
    </form>
  </div>
</div>

<div v-if="loading" class="text-center w-100">
  <div class="spinner-border" role="status">
    <span class="visually-hidden">Loading...</span>
  </div>
</div>

<div v-else class="row row-cols-1 row-cols-sm-2
  row-cols-md-3 g-3">
  <PropertyCard v-for="property in properties"
    :key="property.id"
    :property="property" />
</div>
</main>
</template>

<script setup>
import { ref, onMounted } from 'vue'
import PropertyCard from './PropertyCard.vue'

const properties = ref([])

const filters = ref({
  type: '', minPrice:
  null, maxPrice:
  null
})
const loading = ref(true)

```

```
// Функции загрузки и фильтрации данных
</script>
```

Результаты работы

В ходе выполнения работы был успешно развернут Vue-проект с помощью Vite, подключен Bootstrap для стилизации. Функционал поиска недвижимости, ранее реализованный на нативном JavaScript, был перенесен на компоненты Vue.

Реализованы следующие функциональные возможности:

- Компонентная структура приложения (шапка, подвал, карточка объекта, страница поиска);
- Реактивная фильтрация данных по типу недвижимости и диапазону цен;
- Динамическая отрисовка списка объектов с использованием директивы v-for;
- Передача данных между компонентами через props;
- Обработка событий формы с использованием директивы @submit.prevent.

Вывод

В ходе выполнения домашнего задания были изучены основные команды npm и инструменты инициализации проекта (Vite). Освоены базовые принципы работы с Vue 3: Composition API, реактивность (ref), создание и композиция компонентов, передача данных через props и обработка событий. Получен практический опыт организации проектной структуры и применения компонентного подхода для создания интерактивного веб-приложения.