

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования

Нижегородский государственный технический
университет им. Р.Е.Алексеева

Институт радиоэлектроники и информационных технологий
Кафедра прикладной математики и информатики

Отчёт № 9

«Разработка ИС по принципу SPA»

по дисциплине

web-разработка

Выполнил:

студент группы 23-ПМ-1

Блинов А. С.

Проверил:

Курушин Е. А.

Нижний Новгород

2025 год

Оглавление

Цели и задачи.....	3
Теоретическая справка	4
Реализация	4
Вывод	8
Приложение	11

Цели и задачи

В ходе данной работы были поставлена следующая цель:

Развернуть локальную среду для разработки веб-приложения на Vue.js, организовать структуру проекта и реализовать главную страницу в виде переиспользуемого компонента.

Для из реализации нужно выполнить следующие задачи:

1. Установите node.js
2. Установите фреймворк разработки web-приложения и создайте проект (vue приложение).
3. Оформите структуру ИС согласно фреймворку.
4. Оформите главную страницу ИС в форме компонента.

Теоретическая справка

Современная веб-разработка стремится к повышению эффективности за счёт модульности и переиспользуемости кода. Node.js и Vue.js представляют собой мощный дуэт технологий, позволяющий создавать производительные и масштабируемые приложения. В данном материале рассматриваются ключевые аспекты работы с этими инструментами: от основ Node.js до принципов компонентного подхода во Vue, что особенно важно для разработчиков, переходящих от традиционных методов к современным практикам.

1. Что такое Node.js? Для чего используется?

Node.js — это кросс-платформенная среда выполнения JavaScript, основанная на движке V8. Она выводит JavaScript за пределы браузера, позволяя разрабатывать:

- Серверные приложения (REST API, WebSocket-серверы)
- Утилиты для сборки проектов (сборщики, транспайлеры)
- Инструменты для автоматизации (скрипты развёртывания, тестирования)
- CLI-приложения

Пример создания простого сервера:

```
const http = require('http');

http.createServer((req, res) => {

  res.end('Hello Node.js!');

}).listen(3000);
```

2. Как происходит установка пакетов в Node.js?

Управление зависимостями осуществляется через npm (или альтернативы вроде yarn/pnpm). Особенности:

- Локальная установка (в папку node_modules проекта):

```
npm install axios
```

- Глобальная установка (для CLI-инструментов):

```
npm install -g @vue/cli
```

- Запись зависимостей в package.json:

```
npm install --save-dev eslint # devDependencies
```

```
npm install --save lodash # dependencies
```

3. В чём отличие старой структуры ИС от новой?

Отличия представлены в таблице 1.

Таблица 1 – отличие старой структуры ИС и новой

Критерий	Традиционный подход	Современный подход (Vue)
Организация	Файлы по типам (js/, css/)	Компоненты по функционалу
Модульность	Слабая	Высокая (изолированные .vue-файлы)
Сборка	Ручное подключение скриптов	Автоматическая (Vite/Webpack)
Пример	project/ ├── index.html ├── script.js └── style.css	src/ ├── components/ ├── composables/ └── views/

4. Что такое компонент?

Компонент во Vue — это автономный блок интерфейса с:

- 1) Шаблоном (HTML-разметка с директивами Vue)
- 2) Логикой (JavaScript в Composition/Options API)
- 3) Стилями (scoped CSS или CSS-модули)

Пример расширенного компонента:

```
<template>

  <button

    @click="increment"

    :class="{ 'active': isActive }"

  >

    Счёт: {{ count }}

  </button>

</template>

<script setup>

import { ref } from 'vue';

const count = ref(0);

const isActive = ref(false);

const increment = () => {

  count.value++;

  isActive.value = !isActive.value;
```

```
};
```

```
</script>
```

```
<style scoped>
```

```
button {
```

```
    padding: 8px 16px;
```

```
}
```

```
.active {
```

```
    background: teal;
```

```
}
```

```
</style>
```

Ключевые преимущества компонентов:

- Переиспользуемость (можно вставлять в разные части приложения)
- Изолированность (scoped-стили не влияют на другие компоненты)
- Тестируемость (логику можно проверять отдельно от верстки)

Реализация

1. Установка Node.js

С официального сайта <https://nodejs.org> была загружена и установлена актуальная LTS-версия.

Проверка установки производилась через командную строку.

`node -v` – отображает версию Node.js

`npm -v` – отображает версию менеджера пакетов npm.

2. Установка проекта

`npm create vue@latest`

`npm install`

`npm run dev`

3. Структура проекта

Автоматически генерирует структуру проекта, адаптированную под компонентный подход. Стандартная структура выглядит следующим образом:

JAVISTA/

|— public/

| └─ index.html ← базовая HTML-обёртка

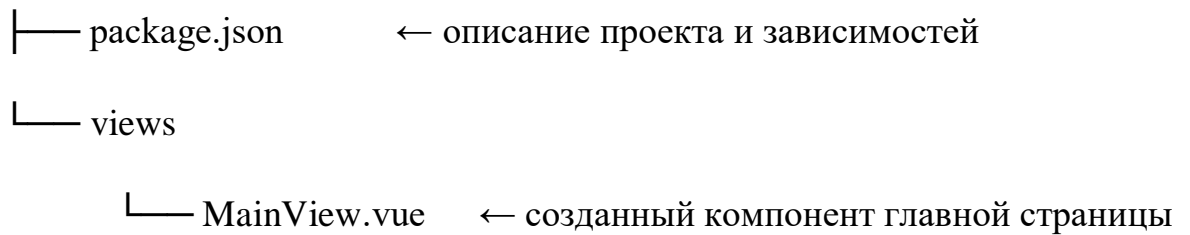
|— src/

| └─ assets/ ← изображения, иконки

| └─ components/ ← Vue-компоненты

| └─ App.vue ← корневой компонент

| └─ main.js ← точка входа в приложение



Эта структура облегчает навигацию по проекту и поддержку кода.

4. Главная страница как компонент

Основной интерфейс был реализован в отдельном компоненте `MainView.vue`, находящемся в папке `src/views/`. Компонент включает несколько логических блоков страницы: заголовок, раздел добавления мест, список популярных направлений и футер.

5. Запуск

```
nmp run dev
```

Сервер запускается на локальном хосте и автоматически обновляется при изменении файлов.

Запуск представлен на рисунке 2.

```
> vue-project@0.0.0 dev
> vite

VITE v6.3.5 ready in 729 ms
→ Local:   http://localhost:5173/
→ Network: use --host to expose
→ Vue DevTools: Open http://localhost:5173/__devtools__/ as a separate window
→ Vue DevTools: Press Alt(⌘)+Shift(⇧)+D in App to toggle the Vue DevTools
→ press h + enter to show help
19:40:58 [vite] (client) Pre-transform error: Failed to resolve import "../components/TheHeader.vue" from "src/App.vue".
```

Рисунок 2 – Запуск

Вывод


В ходе работы была достигнута поставленная цель: создана локальная среда разработки на Vue.js, организована структура проекта и реализована главная страница в виде компонента. Для этого выполнены ключевые задачи: установка Node.js, инициализация Vue-приложения, настройка файловой структуры и разработка компонента главной страницы. В результате подготовлена основа для дальнейшего расширения функционала веб-приложения.

Приложение

```
<script setup>
  import TheHeader from "@/components/TheHeader.vue"
  import TheFooter from "@/components/TheFooter.vue"
</script>

<template>

  <TheHeader />

  <main>
    <section class="course-details">
      <h2>Java Course</h2>
      
      <p class="course-description">
        This course is designed for those who want to master one of the most
        popular programming languages  Java.
        You will learn the basics of the language, object-oriented programming,
        working with databases, multithreading, and much more.
      </p>
      <ul class="course-topics">
        <li>Introduction to Java</li>
        <li>Syntax and Program Structure</li>
        <li>Object-Oriented Programming</li>
        <li>Working with Collections</li>
        <li>Multithreading</li>
        <li>Working with Databases (JDBC)</li>
        <li>Creating Web Applications (Spring Framework)</li>
      </ul>
      <button class="enroll-button">Enroll in the Course</button>
    </section>
  </main>

  <TheFooter />

</template>

<style scoped>
  /* Стили для основного контента */
  main {
    padding: 20px;
  }
}
```

```
.course-details {  
    background-color: #fff;  
    padding: 20px;  
    border-radius: 8px;  
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);  
    max-width: 800px;  
    margin: 0 auto;  
}
```

```
.course-details h2 {  
    margin-top: 0;  
}
```

```
.course-image {  
    max-width: 100%;  
    height: auto;  
    display: block;  
    margin: 20px 0;  
    border-radius: 8px;  
}
```

```
.course-description {  
    line-height: 1.6;  
}
```

```
.course-topics {  
    list-style-type: disc;  
    padding-left: 20px;  
}
```

```
.course-topics li {  
    margin-bottom: 10px;  
}
```

```
.enroll-button {  
    background-color: #28a745;  
    color: #fff;  
    border: none;  
    padding: 10px 20px;  
    font-size: 16px;  
    border-radius: 5px;  
    cursor: pointer;  
    display: block;
```

```

margin: 20px auto 0;
margin-bottom: 35px;
}

.enroll-button:hover {
  background-color: #218838;
}
</style>

<script setup>
  import { RouterLink, RouterView } from 'vue-router'
</script>

<template>
  <RouterLink to = "/">
    <div class="image-up"></div>
  </RouterLink>
  <header id="main-header" class="site-header" data-role="navigation">
    <nav>
      <RouterLink to = "/courses">View Courses</RouterLink>
      <RouterLink to = "/info">View Info</RouterLink>
    </nav>
  </header>
  <RouterView/>
</template>

<style scoped>
  .site-header {
    position: absolute;
    height: 75px;
    width: 100%;
    text-align: center;
    display: flex;
    justify-content: center;
    align-items: center;
    z-index: 10;
    top: 0;
  }

  .site-header nav a {
    margin-right: 15px;
    text-decoration: none;
  }

```

```

        color: #000;
        font-family: 'Inter', sans-serif;
        font-weight: 400;
        font-size: 15px;
        line-height: 24px;
    }

    .image-up {
        width: 222px;
        height: 186px;
        background-image: url('@assets/logo.png');
        background-size: contain;
        background-repeat: no-repeat;
        background-position: center;
        z-index: 2;
    }
</style>

<script>
    import { RouterLink, RouterView } from 'vue-router'
</script>

<template>
    <footer class="footer">
        <div class="footer__section footer__brand">
            <p>Built by <a href="https://vk.com/roubeq">Artem Blinov</a></p>
            
            <p>Javista</p>
        </div>

        <div class="footer__section footer__info">
            <h3>Info</h3>
            <ul>
                <li><a href="#">Features</a></li>
                <li><a href="#">Pricing</a></li>
                <li><a href="#">Support</a></li>
            </ul>
        </div>

        <div class="footer__section footer__admin">
            <h3>Admin</h3>
            <ul>

```

```

        <li><a href="#">Style Guide</a></li>
        <li><a href="#">Licenses</a></li>
        <li><a href="#">Instructions</a></li>
    </ul>
</div>

<div class="footer__section footer__newsletter">
    <h3>Newsletter</h3>
    <p>Sign up for the latest news, company insights, and Whirl updates.</p>
    <form class="feedback-form">
        <label for="first-name">Имя:</label>
        <input type="text" id="first-name" name="first-name" required>

        <label for="last-name">Фамилия:</label>
        <input type="text" id="last-name" name="last-name" required>

        <label for="review">ОТЗЫВ:</label>
        <textarea id="review" name="review" rows="4" required></textarea>

        <button type="submit">ОТПРАВИТЬ</button>
    </form>

</div>
</footer>
<footer class="all-rights">
    <p>&copy; 2023 All rights reserved.</p>
</footer>
</template>

<style scoped>
    /* Футер */
    .footer {
        background: #f8f8f8;
        width: 100%;
        max-width: 100vw;
        display: flex;
        justify-content: center;
        flex-wrap: wrap;
        text-align: center;
        padding: 20px;
        margin-top: auto;
    }

    a {

```

```

    text-decoration: none;
}

.footer__section {
    flex: 1 1 200px;
    max-width: 300px;
    margin-bottom: 20px;
}

.footer__brand img {
    width: 200px;
    margin-bottom: 10px;
}

.footer__section h3 {
    font-size: 18px;
    margin-bottom: 10px;
}

.footer__section ul {
    list-style: none;
    padding: 0;
}

.footer__section ul li a {
    text-decoration: none;
    color: #333;
    transition: color 0.3s;
}

.footer__section ul li a:hover {
    color: #007bff;
}

/* Форма обратной связи */
.feedback-form {
    display: flex;
    flex-direction: column;
    gap: 10px;
    max-width: 400px;
    margin: auto;
}

.feedback-form input,

```



```
.feedback-form textarea {
  border: 1px solid #ccc;
  border-radius: 5px;
  width: 100%;
}

.feedback-form button {
  background: #2b7db7;
  color: white;
  border: none;
  padding: 10px;
  cursor: pointer;
}

.feedback-form button:hover {
  background: #1a5a85;
}

.all-rights {
  background-color: #333;
  color: #fff;
  text-align: center;
  position: fixed;
  margin-top: 100px;
  bottom: 0;
  width: 100%;
  z-index: 1;
}
</style>
```