

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 № 8

«Применение библиотек web-приложений в ИС»

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	7
Вывод	7
Приложение	12

Цели и задачи

В ходе данной работы были поставлены следующая цель:

Изучить базовые принципы работы с библиотекой Vue.js для разработки динамических web-интерфейсов. В ходе выполнения задания необходимо подключить Vue к HTML-странице, реализовать связывание данных, отобразить список элементов, применить условную отрисовку и добавить обработку пользовательских действий. Это позволит закрепить основы реактивного программирования и компонентной структуры, применяемых при создании современных web-приложений.

Для из реализации нужно выполнить следующие задачи:

1. Подключите к имеющейся странице библиотеку web-приложения (vue.js). Можно создать новую страницу ИС.
2. Свяжите несколько элементов (разными типами связывания).
3. Выведите массив данных (не менее 10 шт.) приложением. Например, список, набор изображений, новости и т.д.
4. Используйте на странице условную отрисовку.
5. Добавьте обработчик события приложением.

Теоретическая справка

Vue.js — это гибкая и легковесная JavaScript-библиотека, предназначенная для разработки интерактивных пользовательских интерфейсов. Её главное преимущество — это реактивная система, при которой изменения в данных автоматически отображаются в интерфейсе без дополнительной работы с DOM. В данной работе рассматриваются основы работы с Vue: связывание данных, отображение массивов, работа с условиями и событиями.

1. Как происходит добавление библиотеки?

Чтобы использовать Vue.js на странице, достаточно подключить её через внешний источник (CDN). Это делается путём добавления тега `<script>` в HTML-документ, например:

```
<script src="https://unpkg.com/vue@3/dist/vue.global.js"></script>
```

После этого можно инициализировать приложение Vue, вызвав `Vue.createApp({...}).mount('#app')`, где `#app` — id элемента, к которому будет привязано приложение.

2. Что такое связывание? Какое бывает?

Связывание (binding) — это способ автоматической синхронизации данных приложения с DOM. Vue обеспечивает два основных типа привязки:

- Односторонняя: значение передаётся из данных в интерфейс, например, с помощью `{{ message }}` или директивы `v-bind`.

- Двусторонняя: используется `v-model` и позволяет данным и элементу HTML взаимодействовать взаимно, что особенно полезно для ввода текста в формах.

3. Как происходит отрисовка списка?

Vue предоставляет директиву `v-for` для циклического отображения элементов массива. Она создаёт повторяющиеся DOM-элементы для каждого элемента списка. Пример:

```
<ul>

  <li v-for="(item, index) in items" :key="index">{{ item }}</li>

</ul>
```

В этом примере `items` — это массив, элементы которого последовательно выводятся в виде списка ``.

4. Что такое условная отрисовка? Какие бывают?

Условная отрисовка позволяет отображать элементы только при выполнении определённых условий. В Vue это реализуется через следующие директивы:

`v-if` — элемент отображается только если условие истинно; если нет — элемент полностью удаляется из DOM.

`v-else-if` и `v-else` — создают цепочку условий.

`v-show` — элемент остаётся в DOM, но его видимость регулируется через CSS.

Пример использования:

```
<p v-if="user">Добро пожаловать, {{ user }}!</p>

<p v-else>Гость, пожалуйста, войдите.</p>
```

5. Как происходит обработка события?

Vue позволяет обрабатывать события с помощью директивы `v-on`, которую можно записывать сокращённо как `@`. Это позволяет вызывать методы при наступлении событий, таких как нажатие кнопки:

```
<button @click="handleClick">Кликни</button>
```

Здесь `handleClick` — это метод, определённый внутри раздела `methods` Vue-приложения.

6. Что такое директивы?

Директивы — это специальные конструкции Vue, добавляющие динамическое поведение к HTML. Они начинаются с префикса `v-` и предоставляют различные возможности управления DOM на основе данных.

Наиболее распространённые директивы:

`v-bind` — привязка атрибутов.

`v-model` — двусторонняя синхронизация формы и данных.

`v-for` — отображение списка.

`v-if`, `v-show` — отображение по условию.

`v-on` — обработка событий.

Директивы являются одним из ключевых элементов, упрощающих работу с DOM и логикой интерфейса в Vue.

Реализация

Для выполнения задания по изучению Vue.js была модифицирована структура существующей HTML-страницы, добавлены элементы, работающие на реактивной системе Vue, а также реализованы основные функции — привязка данных, рендеринг списка, условная отрисовка и работа с событиями.

1. Подключение Vue.js

В шапке HTML-документа была добавлена строка подключения Vue через CDN:

```
<script src="https://unpkg.com/vue@3/dist/vue.global.js"></script>
```

2. Разметка для Vue-приложения

В теле документа создан отдельный блок с `id="app"`, в который встраивается Vue-приложение. Внутри него:

- Заголовок с односторонним связыванием (`{{ title }}`),
- Поле ввода с двусторонней привязкой (`v-model`),
- Кнопка добавления элемента с обработчиком события (`@click`),
- Условное сообщение об отсутствии элементов (`v-if`),
- Список с `v-for` и кнопками удаления элементов.

На рисунке 2 представлен блок с Vue в DOM — до добавления элементов (отображается "Список пуст").

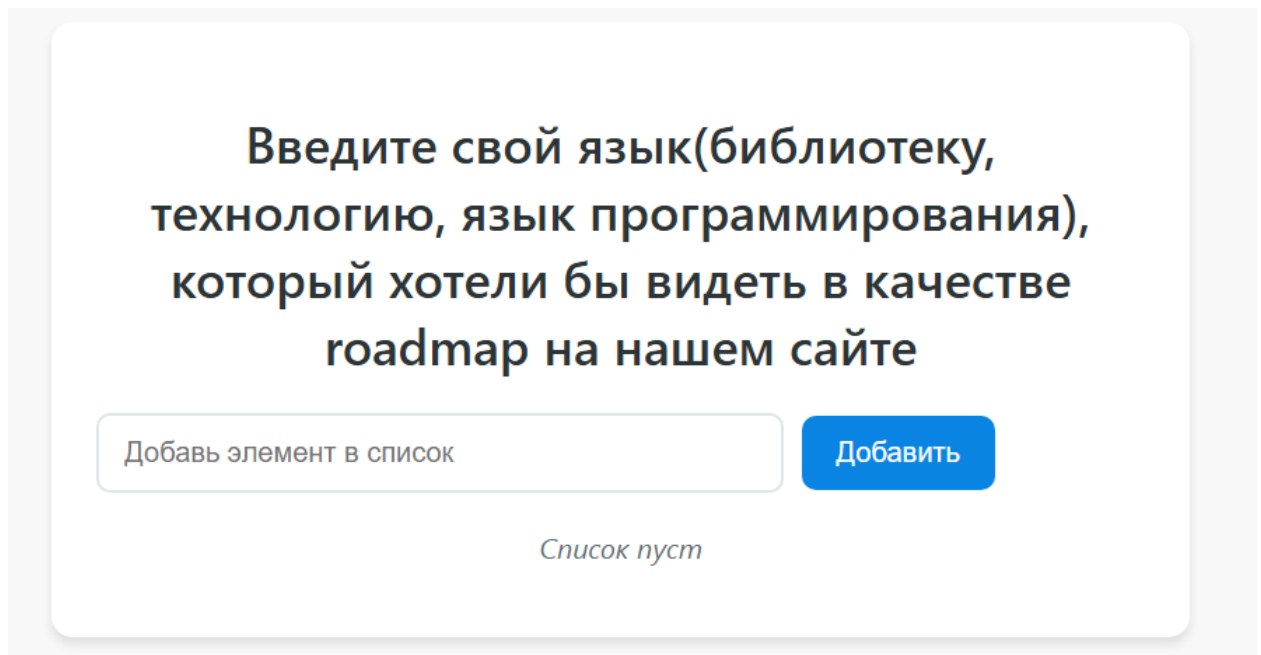


Рисунок 1 – Пустой блок

На рисунке 2 представлен блок с Vue — после добавления нескольких пунктов в список.

**Введите свой язык(библиотеку,
технологию, язык программирования),
который хотели бы видеть в качестве
roadmap на нашем сайте**

Добавь элемент в список

Добавить

1. HTML

Удалить

2. CSS

Удалить

3. JavaScript

Удалить

4. Vue.js

Удалить

5. React

Удалить

6. Node.js

Удалить

7. Express

Удалить

8. MongoDB

Удалить

9. Git

Удалить

10. Webpack

Удалить

Рисунок 1 – Блок со списком

3. Логика приложения на Vue

Внизу HTML-файла описано само Vue-приложение. Используется функция `createApp()` для инициализации. Внутри компонента:

- В `data()` хранится состояние: заголовок, текущий ввод и массив элементов.
- Методы `addItem()` и `removeItem()` позволяют управлять содержимым списка (добавлять и удалять пункты).

4. Взаимодействие с пользователем

Пользователь может добавлять новые элементы в список, вводя их в текстовое поле и нажимая кнопку "Добавить". Каждый добавленный элемент появляется с кнопкой "Удалить", которая при нажатии убирает его из массива и, соответственно, из интерфейса. Так же была добавлена стилизация.

Таким образом, были продемонстрированы базовые возможности Vue: реактивное отображение данных, управление списками и событиями, а также динамическое изменение интерфейса без ручной работы с DOM.

Вывод

В результате выполнения работы была достигнута поставленная цель — освоены базовые приёмы работы с библиотекой Vue.js для создания интерактивных web-интерфейсов. Подключение Vue к HTML-странице, реализация различных видов привязки данных, вывод массива, условная отрисовка элементов и обработка действий пользователя позволили на практике понять принципы реактивности и компонентного подхода. Полученные знания формируют фундамент для дальнейшего освоения современных технологий фронтенд-разработки.

Приложение

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Javista</title>
  <link rel="stylesheet" href="CSS/main.css" />
  <script src="Java Script/six1.js" defer></script>
  <script src="Java Script/seventh.js" defer></script>
  <!-- 8 lab -->
  <script src="https://unpkg.com/vue@3/dist/vue.global.js"></script>
  <!-- 8 lab -->
</head>
<body>
  <a href="mainPage.html">
    <div class="image-up"></div>
  </a>
  <header id="main-header" class="site-header" data-role="navigation">
    <nav>
      <a href="courses.html" title="View Courses">Courses</a>
      <a href="infoPage.html" title="View Info">Info</a>
    </nav>
  </header>

  <section id="about" class="about-section">
    <h1 id="about-title">What's Javista all about?</h1>
    <div class="features">
      <div class="feature" data-category="info">
        <h3>All on one place.</h3>
        <p>Everything you need, in one secure platform.</p>
      </div>
      <div class="feature" data-category="alerts">
        <h3>Get daily alerts.</h3>
        <p>Stay updated with real-time notifications.</p>
      </div>
      <div class="feature" data-category="security">
        <h3>Safe and secure.</h3>
        <p>Your data is protected with top encryption.</p>
      </div>
    </div>
  </section>

  <div id="data-container"></div>

  <section id="track" class="track-section" data-role="automation">
    <div class="track-section">
      

      <div class="track-section-text">
        <h2> Set, forget, and then track.</h2>
        <p>Choose a direction, study step by step and achieve your
goals.</p>
        <ul class="custom-list">
          <li>
            
            Understand which technologies are important
          </li>
          <li>
            
            Follow proven learning paths
          </li>
        </ul>
      </div>
    </div>
  </section>
</body>
</html>
```

```

        <li>
            
            Master skills without chaos and wasted time
        </li>
    </ul>
</div>
</div>
</section>

<section id="automation" class="automation-section">
    <h2>Your tasks, automated.</h2>
    <div class="task-grid">
        <div class="task" data-feature="options">
            
            <h3>Learn your options.</h3>
            <p>Understand what works best for you.</p>
        </div>
        <div class="task" data-feature="updates">
            
            <h3>Stay informed.</h3>
            <p>Get the latest updates and insights.</p>
        </div>
        <div class="task" data-feature="automation">
            
            <h3>Automate it all.</h3>
            <p>Let our platform handle the tasks.</p>
        </div>
    </div>
</section>

<!-- 8 lab -->

<section id="vue-app">
    <div id="app">
        <!-- Одностороннее связывание -->
        <h2>{{ title }}</h2>

        <!-- Двустороннее связывание -->
        <input v-model="newItem" placeholder="Добавь элемент в список" />
        <button @click="addItem">Добавить</button>

        <!-- Условная отрисовка -->
        <p v-if="items.length === 0">Список пуст</p>

        <!-- Вывод массива данных -->
        <ul v-else>
            <li v-for="(item, index) in items" :key="index">
                {{ index + 1 }}. {{ item }}
                <!-- А тут клик -->
                <button @click="removeItem(index)">Удалить</button>
            </li>
        </ul>
    </div>
</section>

<!-- 8 lab -->

<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>

<!-- 8 lab -->
<script>
  const { createApp } = Vue;

  createApp({
    data() {
      return {
        title: 'Введите свой язык(библиотеку, технологию, язык
программирования), который хотели бы видеть в качестве roadmap на нашем сайте ',
        newItem: '',
        items: [
          'HTML',
          'CSS',
          'JavaScript',
          'Vue.js',
          'React',
          'Node.js',
          'Express',
          'MongoDB',
          'Git',
          'Webpack'
        ]
      }
    },
    methods: {
      addItem() {

```

```

        if (this.newItem.trim() !== '') {
            this.items.push(this.newItem.trim());
            this.newItem = '';
        }
    },
    removeItem(index) {
        this.items.splice(index, 1);
    }
}
}).mount('#app');
</script>
<!-- 8 lab -->

</body>
</html>

```

Main.css

```

/* Основные стили секции */
#vue-app {
    font-family: 'Segoe UI', Roboto, sans-serif;
    max-width: 600px;
    margin: 20px auto;
    padding: 25px;
    background: #ffffff;
    border-radius: 12px;
    box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);
}

/* Заголовок */
#vue-app h2 {
    color: #2d3436;
    margin-bottom: 20px;
    font-size: 1.8rem;
    font-weight: 600;
    text-align: center;
}

/* Поле ввода и кнопка Добавить */
#vue-app input {
    padding: 12px 15px;
    border: 2px solid #dfe6e9;
    border-radius: 8px;
    width: 60%;
    font-size: 1rem;
    margin-right: 10px;
}

#vue-app button {
    padding: 12px 20px;
    background-color: #0984e3;
    color: white;
    border: none;
    border-radius: 8px;
    font-size: 1rem;
    font-weight: 500;
    cursor: pointer;
    transition: all 0.3s ease;
}

```

```

/* Сообщение о пустом списке */
#vue-app p {
    color: #636e72;
    text-align: center;
    margin-top: 20px;
    font-style: italic;
}

/* Список элементов */
#vue-app ul {
    list-style: none;
    padding: 0;
    margin-top: 25px;
}

#vue-app li {
    display: flex;
    justify-content: space-between;
    align-items: center;
    padding: 15px 20px;
    margin-bottom: 10px;
    background-color: #f5f6fa;
    border-radius: 8px;
    transition: transform 0.2s ease;
}

/* Кнопка Удалить */
#vue-app li button {
    padding: 8px 12px;
    background-color: #ff7675;
    color: white;
    border: none;
    border-radius: 6px;
    font-size: 0.9rem;
    cursor: pointer;
    transition: background-color 0.2s ease;
}

```