

ГУАП

КАФЕДРА № 41

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ
ПРЕПОДАВАТЕЛЬ

ассистент		Н. И. Мирошниченко
_____ должность, уч. степень, звание	_____ подпись, дата	_____ инициалы, фамилия

ОТЧЕТ О ИТОГОВОМ ПРОЕКТЕ

по курсу: JavaScript, его библиотеки и фреймворки в Frontend-разработке

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ гр. №	4217		А.Н. Медянкина
	_____ подпись, дата		_____ инициалы, фамилия

Санкт-Петербург 2025

1. Требования к проекту

Разработать Single Page Application (SPA), выполненный на основе библиотеки Vue.js на согласованную с преподавателем тему, отвечающий требованиям адаптивности, кроссбраузерности, интерактивности. Допускается использование для разработки проекта библиотеки React.js в случае, если обучающийся самостоятельно в необходимом объеме освоил принципы работы с данной библиотекой.

Для проекта целесообразно выбирать тему, которая может быть достаточно полно раскрыта в объеме данной работы. Объем веб-сайта определяется требованием полноты представления информации по рассматриваемой теме. Можно выбрать тему, включенную в перечень рекомендуемых тем (см. табл. 1). Разрешается выбрать тему самостоятельно (можно использовать тематику, в соответствии с которой выполнялись практические задания по дисциплине).

2. Выбранная тема проекта

Веб-сайт «Протокол IPv6»

3. Введение

В современном мире, где технологии развиваются с невероятной скоростью, интернет стал неотъемлемой частью нашей жизни. Однако, с ростом числа устройств, подключенных к сети, возникла проблема нехватки IP-адресов. Протокол IPv4, который использовался с 1981 года, больше не мог удовлетворить потребности растущего интернета. В ответ на эту проблему был разработан новый протокол — IPv6. Наш проект посвящен именно этому протоколу и направлен на информирование пользователей о его преимуществах и особенностях.

4. Постановка задачи

Основной задачей проекта было создание Single Page Application (SPA) с использованием библиотеки Vue.js. Приложение должно было быть адаптивным, кроссбраузерным и интерактивным, чтобы обеспечить удобство использования на различных устройствах и в разных браузерах. Была выбрана тема «Протокол IPv6», так как она актуальна и востребована в современном мире технологий.

В рамках проекта предстояло решить несколько ключевых задач. Во-первых, необходимо было разработать макет сайта, который бы включал все основные разделы, посвященные IPv6. Во-вторых, требовалось реализовать реактивность сайта с использованием Vue.js, чтобы пользователи могли легко и удобно взаимодействовать с контентом. В-третьих, нужно было обеспечить адаптивность и кроссбраузерность, чтобы сайт корректно отображался на всех устройствах и в различных браузерах. И, наконец, важной задачей было тестирование и отладка, чтобы убедиться в корректной работе всех функций и элементов сайта.

Состав команды (группа 4217):

Алёна Лавелина – проектирование структуры сайта.

Анастасия Медянкина – разработка макета и стилей (CSS).

Дмитрий Никитин – реализация логики на Vue.js (маршрутизация, обработка форм, слайдер).

Лилия Мирошниченко – наполнение контентом и тестирование.

5. Проектирование сайта

Для реализации проекта были выбраны современные и надежные технологии. Vue.js был выбран для создания интерактивных компонентов, так как он позволяет легко и эффективно управлять состоянием приложения и обеспечивает реактивность интерфейса. Bootstrap использовался для обеспечения адаптивности и стилизации, что позволило создать удобный и привлекательный дизайн. HTML и CSS были использованы для структуры и оформления страниц, чтобы обеспечить их корректное отображение на различных устройствах.

5.1 Макет сайта

Макет сайта был разработан с учетом всех требований и пожеланий. Он включает несколько основных разделов, каждый из которых посвящен определенной теме, связанной с IPv6.

Раздел «Обзор» предоставляет информацию о истории IPv6, его основных характеристиках и ключевых отличиях от IPv4 (см. рис. 1).



Рисунок 1 – Раздел сайта «Обзор»

В разделе «Преимущества» подробно рассматриваются основные преимущества IPv6, такие как большой адресный пул, встроенная поддержка IPsec, улучшенная маршрутизация и другие (см. рис. 2).

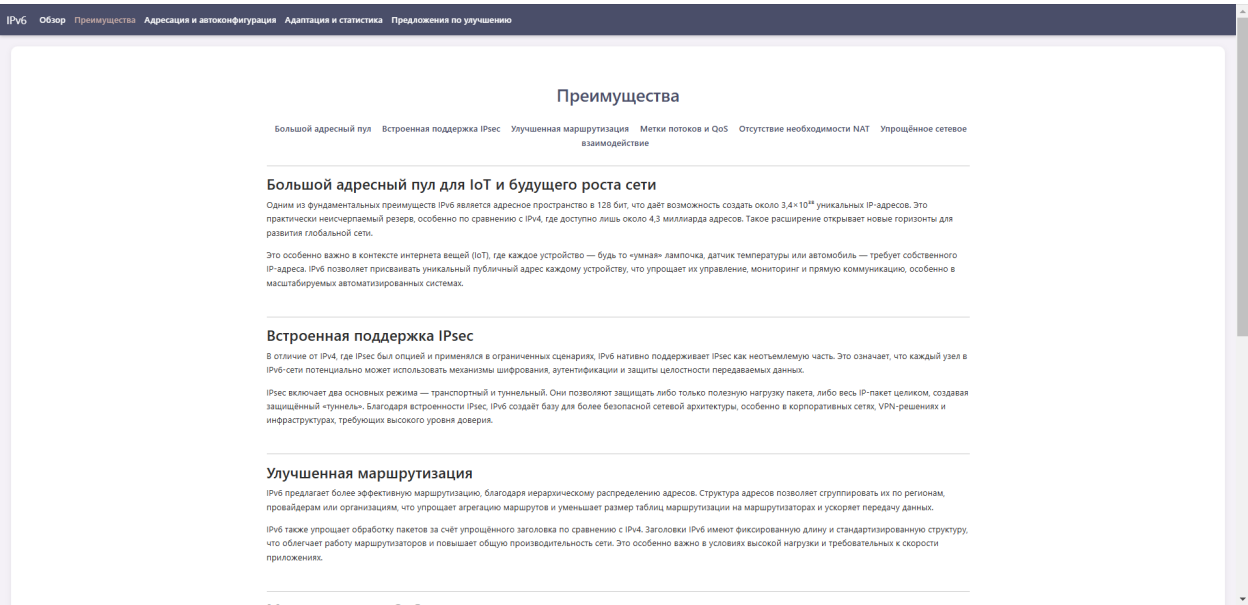


Рисунок 2 – Раздел сайта «Преимущества»

Раздел «Адресация и автоконфигурация» посвящен различным аспектам адресации и автоконфигурации в IPv6, включая SLAAC, EUI-64, локальные адреса и другие (см. рис. 3).

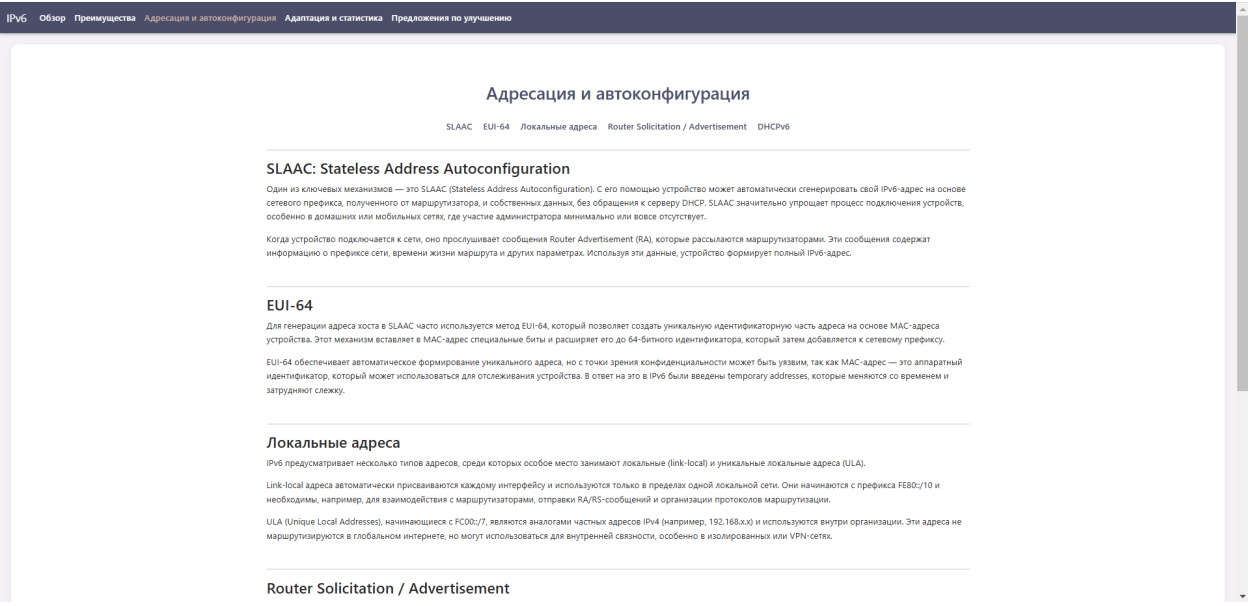


Рисунок 3 – Раздел сайта «Адресация и автоконфигурация»

В разделе «Адаптация и статистика» представлена информация о

глобальном проценте IPv6-пользователей, крупнейших странах-пионерах и вехах внедрения (см. рис. 4).

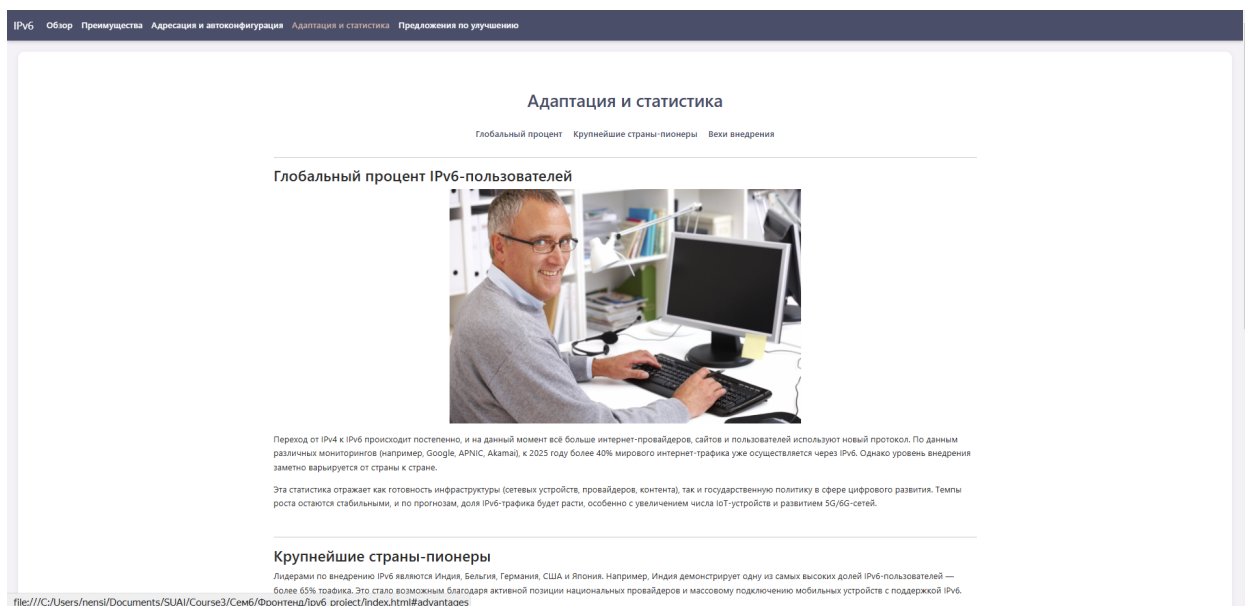


Рисунок 4 – Раздел сайта «Адаптация и статистика»

Наконец, раздел «Предложения по улучшению» содержит форму для сбора предложений от пользователей, что позволит нам получить обратную связь и улучшить сайт (см. рис. 5).

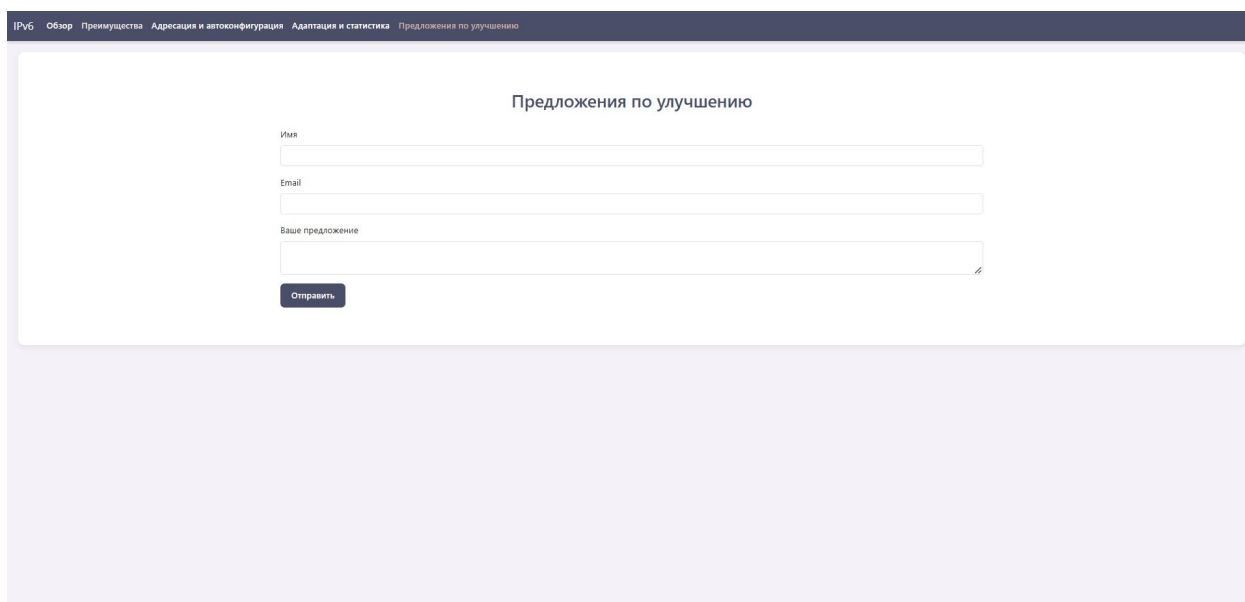


Рисунок 5 – Раздел сайта «Предложения по улучшению»

5.2 Реактивность сайта с использованием Vue.js.

Одной из ключевых задач проекта было обеспечение реактивности сайта. Vue.js оказался идеальным решением для этой задачи. С помощью Vue.js удалось создать динамические и интерактивные компоненты, которые позволяют пользователям легко и удобно взаимодействовать с контентом. Например, было реализовано переключение между разделами с использованием директив `v-if` и `v-show`, что позволило динамически обновлять контент без перезагрузки страницы. Также были обработаны события кликов для навигации по разделам, что позволило сделать использование сайта более интуитивным и удобным.

В нашем случае за переключение между разделами сайта отвечает метод `setTab`. Он принимает аргумент `tab`, который представляет собой идентификатор вкладки. При вызове этого метода значение `activeTab` обновляется, что приводит к изменению активной вкладки. Также выполняется прокрутка страницы вверх с плавным поведением. Далее будет представлен фрагмент кода на рисунке 6.

```

1  const { createApp } = Vue;
2
3  createApp({
4    data() {
5      return {
6        activeTab: 'overview',
7        currentSlide: 0,
8        slides: 5,
9        form: {
10         name: '',
11         email: '',
12         proposal: ''
13       },
14       modal: {
15         title: '',
16         message: ''
17       }
18     },
19   },
20   methods: {
21     setTab(tab) {
22       this.activeTab = tab;
23       window.scrollTo({ top: 0, behavior: 'smooth' });
24     },
25     scrollTo(id) {
26       const element = document.getElementById(id);
27       if (element) {
28         element.scrollIntoView({ behavior: 'smooth' });

```

Рисунок 6 – Фрагмент кода с методом setTab

5.3 Описание взаимодействия с элементами на странице с использованием функционала Vue. (В данном подразделе рассмотреть, какие методы, компоненты, модули и хуки реализованы в данном SPA и с какой целью, привести фрагменты кода с необходимыми комментариями).

Управление активными вкладками

Одной из ключевых задач в нашем приложении было создание системы навигации по вкладкам, которая позволяла бы пользователям легко переключаться между различными разделами контента. Для этого за основу был взят Vue.js, который предоставляет мощные инструменты для управления состоянием приложения и реактивным обновлением интерфейса.

Основной механизм управления вкладками реализован через свойство `activeTab`, которое хранит идентификатор текущей активной вкладки. Это свойство определено в объекте данных Vue.js:

```
data() {  
  return {  
    activeTab: 'overview',  
    currentSlide: 0,  
    slides: 5,  
    form: {  
      name: '',  
      email: '',  
      proposal: ''  
    },  
  },  
}
```

Рисунок 7 – Код с реализацией `activeTab`

Здесь `activeTab` инициализируется значением «overview», что соответствует первой вкладке, которую пользователь видит при загрузке страницы.

Для переключения между вкладками был реализован метод `setTab`, который принимает идентификатор вкладки в качестве аргумента и обновляет значение `activeTab`:

```

methods: {
  setTab(tab) {
    this.activeTab = tab;
    window.scrollTo({ top: 0, behavior: 'smooth' });
  },
  scrollTo(id) {
    const element = document.getElementById(id);
    if (element) {
      element.scrollIntoView({ behavior: 'smooth' });
    }
  },
  prevSlide() {
    this.currentSlide = (this.currentSlide - 1 + this.slides) % this.slides;
  },
}

```

Рисунок 8 – Код с реализацией activeTab

Метод setTab не только обновляет активную вкладку, но и обеспечивает плавную прокрутку страницы вверх, что улучшает пользовательский опыт.

В шаблоне HTML метод setTab вызывается при клике на ссылки навигации (см. рис. 9).

```

<div class="collapse navbar-collapse" id="navbarNav">
  <ul class="navbar-nav">
    <li class="nav-item">
      <a class="nav-link" :class="{ active: activeTab === 'overview' }" href="#overview" @click.prevent="setTab('overview')">Обзор</a>
    </li>
  </ul>
</div>

```

Рисунок 9 – Использование метода setTab

Здесь используется директива @click.prevent, чтобы предотвратить стандартное поведение ссылки и вызвать метод setTab с аргументом «overview». Это позволяет переключаться между разделами сайта, обновляя контент динамически и обеспечивая удобный и интуитивный пользовательский опыт.

Условное отображение контента

Для условного отображения контента в зависимости от активной вкладки была использована директива v-if. Эта директива позволяет

показывать или скрывать элементы на странице в зависимости от условия. Например:

```
<div v-if="activeTab === 'overview'" id="overview" class="section">
  <div class="container">
    <h2>Обзор</h2>
    <div class="subnav">
      <a href="#history" @click.prevent="scrollTo('history')">История IPv6</a>
```

Рисунок 10 – Использование директивы v-if

В этом фрагменте кода раздел с id=«overview» будет отображаться только в том случае, если activeTab имеет значение «overview». Аналогично, другие разделы также используют директиву v-if для управления их видимостью:

```
<div v-if="activeTab === 'advantages'" id="advantages" class="section">
  <div class="container">
    <h2>Преимущества</h2>
    <div class="subnav">
      <a href="#address-pool" @click.prevent="scrollTo('address-pool')">Большой адресный пул</a>
```

Рисунок 11 – Использование директивы v-if

Это позволяет динамически обновлять контент на странице без необходимости перезагрузки, что делает взаимодействие с сайтом более плавным и приятным.

Обработка событий и взаимодействие с пользователем

Для обработки событий и взаимодействия с пользователем была использована директива v-on (или сокращенно @). Эта директива позволяет назначать обработчики событий на различные элементы страницы. Например, для обработки кликов по ссылкам навигации внутри разделов:

Здесь используется метод scrollTo, который обеспечивал плавную прокрутку до указанного элемента на странице:

```
<div class="subnav">
  <a href="#history" @click.prevent="scrollTo('history')">История IPv6</a>
  <a href="#characteristics" @click.prevent="scrollTo('characteristics')">Основные характеристики</a>
  <a href="#differences" @click.prevent="scrollTo('differences')">Ключевые отличия от IPv4</a>
</div>
```

Рисунок 12 – Использование метода scrollTo

Этот метод принимает идентификатор элемента и использует метод `scrollIntoView` для плавной прокрутки страницы до этого элемента. Это улучшает навигацию по разделу и делает взаимодействие с контентом более удобным.

```
scrollTo(id) {
  const element = document.getElementById(id);
  if (element) {
    element.scrollIntoView({ behavior: 'smooth' });
  }
},
```

Рисунок 13 – Описание метода scrollTo

Работа с формами

Для работы с формами и обработки пользовательского ввода была использована директива `v-model`, которая обеспечивает двустороннее связывание данных между элементами формы и данными в компоненте `Vue.js`. Например:

```

<form @submit.prevent="submitForm">
  <div class="mb-3">
    <label for="name" class="form-label">Имя</label>
    <input type="text" class="form-control" id="name" v-model="form.name" required>
  </div>
  <div class="mb-3">
    <label for="email" class="form-label">Email</label>
    <input type="email" class="form-control" id="email" v-model="form.email" required>
  </div>
  <div class="mb-3">
    <label for="proposal" class="form-label">Ваше предложение</label>
    <textarea class="form-control" id="proposal" v-model="form.proposal" required></textarea>
  </div>
  <button type="submit" class="btn btn-primary">Отправить</button>
</form>

```

Рисунок 14 –Директива v-model

На рисунке 15 v-model связывает значение поля ввода с свойством form.name в данных компонента.

```

data() {
  return {
    // activeTab: 'overview',
    // currentSlide: 0,
    // slides: 5,
    form: {
      name: '',
      email: '',
      proposal: ''
    },
  },
}

```

Рисунок 15 – Пример реализации связи v-model с свойством form.name

При отправке формы вызывается метод submitForm, который обрабатывает введенные данные. На рисунке 16 представлена реализация.

```

submitForm() {
  const emailRegex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;
  if (!this.form.name || !this.form.email || !this.form.proposal) {
    this.modal.title = 'Ошибка';
    this.modal.message = 'Все поля обязательны для заполнения.';
  } else if (!emailRegex.test(this.form.email)) {
    this.modal.title = 'Ошибка';
    this.modal.message = 'Пожалуйста, введите корректный email.';
  } else {
    this.modal.title = 'Успех';
    this.modal.message = 'Ваше предложение успешно отправлено!';
    this.form.name = '';
    this.form.email = '';
    this.form.proposal = '';
  }
  const modal = new bootstrap.Modal(document.getElementById('formModal'));
  modal.show();
}

```

Рисунок 16 – Реализация метода submitForm

Этот метод проверяет корректность введенных данных и отображает модальное окно с сообщением об успехе или ошибке. Это обеспечивает обратную связь с пользователем и улучшает взаимодействие с формой.

6. Заключение

В ходе выполнения проекта была проделана значительная работа по созданию Single Page Application (SPA), посвящённого протоколу IPv6. Использование библиотеки Vue.js позволило реализовать интерактивность, реактивность и удобную навигацию между разделами без перезагрузки страницы. Все поставленные задачи, включая разработку адаптивного и кроссбраузерного интерфейса, были успешно выполнены.

В процессе работы над проектом были успешно реализованы все поставленные задачи. Разработан макет сайта, включающий основные разделы, посвященные истории, преимуществам, адресации и автоконфигурации IPv6, а также адаптации и статистике его использования. Реализована реактивность сайта с использованием Vue.js, что обеспечило динамическое обновление контента без перезагрузки страницы и улучшило пользовательский опыт. Также реализованы адаптивность и кроссбраузерность, что позволяет сайту корректно отображаться на различных устройствах и в разных браузерах. Проведено тестирование и отладка, что гарантирует корректную работу всех функций и элементов сайта.

Проект готов к публикации в интернете и может быть полезен для пользователей, интересующихся современными сетевыми технологиями. Веб-сайт предоставляет полную и актуальную информацию о протоколе IPv6, его преимуществах и особенностях, что способствует повышению осведомленности и пониманию важности перехода на новый протокол.

С точки зрения сформированных компетенций в процессе обучения, проект позволил применить и закрепить знания, полученные в ходе изучения курса. Были развиты навыки работы с библиотекой Vue.js, что включает управление состоянием приложения, реактивное обновление интерфейса, работу с формами и обработку событий. Также были улучшены навыки работы с HTML и CSS для структурирования и оформления веб-страниц, а также использование Bootstrap для обеспечения адаптивности и стилизации.

В целом, проект оказался успешным и достиг поставленных целей. Он демонстрирует практическую значимость изученных технологий и библиотек, а также их эффективное применение для создания современных и интерактивных веб-приложений.

ПРИЛОЖЕНИЕ А. VUE-КОД

```
const { createApp } = Vue;

createApp({
  data() {
    return {
      activeTab: 'overview',
      currentSlide: 0,
      slides: 5,
      form: {
        name: '',
        email: '',
        proposal: ''
      },
      modal: {
        title: '',
        message: ''
      }
    };
  },
  methods: {
    setTab(tab) {
      this.activeTab = tab;
      window.scrollTo({ top: 0, behavior: 'smooth' });
    },
    scrollTo(id) {
      const element = document.getElementById(id);
      if (element) {
        element.scrollIntoView({ behavior: 'smooth' });
      }
    },
    prevSlide() {
      this.currentSlide = (this.currentSlide - 1 + this.slides)
% this.slides;
    },
    nextSlide() {
      this.currentSlide = (this.currentSlide + 1) % this.slides;
    },
    submitForm() {
      const emailRegex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;
      if (!this.form.name || !this.form.email ||
!this.form.proposal) {
        this.modal.title = 'Ошибка';
        this.modal.message = 'Все поля обязательны для
заполнения.';
      } else if (!emailRegex.test(this.form.email)) {
        this.modal.title = 'Ошибка';
        this.modal.message = 'Пожалуйста, введите корректный
email.';
      } else {

```

```

        this.modal.title = 'Успех';
        this.modal.message = 'Ваше предложение успешно
отправлено!';
        this.form.name = '';
        this.form.email = '';
        this.form.proposal = '';
    }
    const modal = new
bootstrap.Modal(document.getElementById('formModal'));
    modal.show();
  }
},
mounted() {
  window.addEventListener('hashchange', () => {
    const tab = window.location.hash.replace('#', '');
    if (['overview', 'advantages', 'addressing', 'adaptation',
'proposals'].includes(tab)) {
      this.setTab(tab);
    }
  });
}
}).mount('#app');

```

ПРИЛОЖЕНИЕ Б. HTML-КОД

```
<!DOCTYPE html>
<html lang="ru">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>IPv6: Современный интернет-протокол</title>
  <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.
min.css" rel="stylesheet">
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <div id="app">
    <nav class="navbar navbar-expand-lg">
      <div class="container-fluid">
        <a class="navbar-brand text-white"
href="#overview">IPv6</a>
        <button class="navbar-toggler" type="button" data-bs-
toggle="collapse" data-bs-target="#navbarNav" aria-
controls="navbarNav" aria-expanded="false" aria-label="Toggle
navigation">
          <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarNav">
          <ul class="navbar-nav">
            <li class="nav-item">
              <a class="nav-link" :class="{ active:
activeTab === 'overview' }" href="#overview"
@click.prevent="setTab('overview')">Обзор</a>
            </li>
            <li class="nav-item">
              <a class="nav-link" :class="{ active:
activeTab === 'advantages' }" href="#advantages"
@click.prevent="setTab('advantages')">Преимущества</a>
            </li>
            <li class="nav-item">
              <a class="nav-link" :class="{ active:
activeTab === 'addressing' }" href="#addressing"
@click.prevent="setTab('addressing')">Адресация и автоконфигурация</a>
            </li>
            <li class="nav-item">
              <a class="nav-link" :class="{ active:
activeTab === 'adaptation' }" href="#adaptation"
@click.prevent="setTab('adaptation')">Адаптация и статистика</a>
            </li>
            <li class="nav-item">
```

```

        <a class="nav-link" :class="{ active:
activeTab === 'proposals' }" href="#proposals"
@click.prevent="setTab('proposals')">Предложения по улучшению</a>
    </li>
</ul>
</div>
</div>
</div>
</nav>

<div v-if="activeTab === 'overview'" id="overview"
class="section">
    <div class="container">
        <h2>Обзор</h2>
        <div class="subnav">
            <a href="#history"
@click.prevent="scrollTo('history')">История IPv6</a>
            <a href="#characteristics"
@click.prevent="scrollTo('characteristics')">Основные
характеристики</a>
            <a href="#differences"
@click.prevent="scrollTo('differences')">Ключевые отличия от IPv4</a>
        </div>
        <div id="history" class="content-block">
            <hr>
            <h3>История IPv6</h3>
            
            <p>IPv6 (Internet Protocol version 6) был
разработан как ответ на исчерпание адресного пространства в предыдущей
версии протокола – IPv4. Протокол IPv4, появившийся в 1981 году,
использует 32-битные адреса, что дает около 4,3 миллиарда уникальных
IP-адресов. С развитием интернета, ростом количества пользователей,
мобильных устройств и IoT-устройств стало ясно, что этого объема
недостаточно для будущего масштабирования сети.</p>
            <p>Работа над IPv6 началась в 1994 году под
руководством IETF, а в 1998 году был опубликован официальный стандарт
– RFC 2460. Новый протокол использует 128-битные адреса, что позволяет
создать примерно  $3,4 \times 10^{38}$  уникальных адресов. Помимо огромного
адресного пространства, IPv6 включает улучшения в маршрутизации,
встроенную поддержку безопасности (IPsec) и устранение необходимости в
NAT – технологии, которая маскировала нехватку адресов в IPv4.</p>
            <p>Переход на IPv6 оказался постепенным и
технически сложным, так как существующие устройства и сети были
ориентированы на IPv4. Чтобы упростить миграцию, была внедрена модель
dual stack, при которой оба протокола работают параллельно. Знаковым
событием стало проведение World IPv6 Launch Day в 2012 году – тогда
ведущие интернет-компании начали постоянную поддержку IPv6.</p>
            <p>Сегодня внедрение IPv6 продолжается по всему
миру: одни страны и провайдеры продвинулись дальше, другие –
медленнее. Однако в целом IPv6 постепенно становится стандартом,

```

открывая путь к более устойчивому, масштабируемому и современному интернету.</p>

</div>

<div id="characteristics" class="content-block">

<hr>

<h3>Основные характеристики</h3>

<p>IPv6 представляет собой современный сетевой протокол, пришедший на смену IPv4, и обладает рядом важных особенностей, обеспечивающих его гибкость, масштабируемость и безопасность. Вот его ключевые характеристики:</p>

128-битная адресация:

Каждый IP-адрес состоит из 128 бит, что позволяет создать около $3,4 \times 10^{38}$ уникальных адресов. Это практически устраняет проблему нехватки IP-адресов в будущем.

Формат адреса: IPv6-адрес записывается как восемь групп из четырёх шестнадцатеричных цифр, разделённых двоеточиями (например, 2001:0db8:85a3:0000:0000:8a2e:0370:7334). Последовательности нулей могут быть сокращены для удобства.

Без NAT: В отличие от IPv4, IPv6 не требует использования NAT (Network Address Translation), поскольку адресного пространства достаточно, чтобы каждому устройству в сети можно было присвоить уникальный внешний адрес.

Автонастройка (Stateless Address Autoconfiguration): Устройства могут автоматически получать IPv6-адрес при подключении к сети, без необходимости использовать DHCP.

Встроенная поддержка безопасности: IPv6 изначально включает IPsec – набор протоколов для обеспечения шифрования, аутентификации и целостности передаваемых данных.

Упрощённая маршрутизация: Благодаря иерархической структуре адресов и более эффективной агрегации маршрутов, IPv6 снижает нагрузку на маршрутизаторы и ускоряет передачу данных.

Расширяемость протокола: IPv6 использует расширяемую систему заголовков, что позволяет легко добавлять новые функции без нарушения совместимости.

<p>Эти характеристики делают IPv6 не просто заменой IPv4, а более продвинутым решением, рассчитанным на долгосрочное развитие интернета и технологий, завязанных на глобальное сетевое взаимодействие.</p>

</div>

<div id="differences" class="content-block">

<hr>

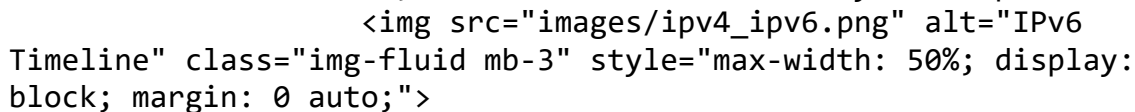
<h3>Ключевые отличия от IPv4</h3>

IPv6 был создан не только для решения проблемы нехватки адресов, но и как более совершенный и гибкий протокол. По сравнению с IPv4, он обладает рядом принципиальных отличий, отражающих изменения в архитектуре и подходах к работе с сетью.

Главное и самое очевидное отличие – это длина IP-адреса. IPv4 использует 32 бита, что ограничивает количество адресов до примерно 4,3 миллиарда. В IPv6 адреса состоят из 128 бит, что позволяет задать число адресов, фактически неисчерпаемое с точки зрения современных потребностей. Это обеспечивает возможность индивидуальной адресации для каждого устройства, без необходимости использовать NAT.

IPv6 также предлагает улучшенную систему конфигурации. Если в IPv4 часто требуется ручная настройка или использование DHCP-серверов, то в IPv6 предусмотрена функция автоконфигурации – устройство может самостоятельно получить адрес и подключиться к сети без дополнительного вмешательства. Кроме того, IPv6 избавляет от многих ограничений, связанных с маршрутизацией, и упрощает её за счёт иерархической структуры адресов.

Существенным новшеством стало включение механизмов безопасности уже на уровне протокола: IPv6 изначально поддерживает IPsec, обеспечивая возможность шифрования и аутентификации трафика. В IPv4 такие механизмы внедрялись дополнительно и далеко не всегда применялись. Также стоит отметить, что в IPv6 отсутствуют широковещательные сообщения (broadcast), что снижает нагрузку на сеть – вместо них используются более эффективные методы доставки данных, такие как multicast и anycast.

Timeline" class="img-fluid mb-3" style="max-width: 50%; display: block; margin: 0 auto;"/>

```
<div v-if="activeTab === 'advantages'" id="advantages"
class="section">
  <div class="container">
    <h2>Преимущества</h2>
    <div class="subnav">
      <a href="#address-pool"
@click.prevent="scrollTo('address-pool')">Большой адресный пул</a>
      <a href="#ipsec"
@click.prevent="scrollTo('ipsec')">Встроенная поддержка IPsec</a>
      <a href="#routing"
@click.prevent="scrollTo('routing')">Улучшенная маршрутизация</a>
      <a href="#qos"
@click.prevent="scrollTo('qos')">Метки потоков и QoS</a>
      <a href="#no-nat" @click.prevent="scrollTo('no-
nat')">Отсутствие необходимости NAT</a>
      <a href="#network"
@click.prevent="scrollTo('network')">Упрощённое сетевое
взаимодействие</a>
```

```
</div>
<div id="address-pool" class="content-block">
  <hr>
  <h3>Большой адресный пул для IoT и будущего роста
сети</h3>

  <p>Одним из фундаментальных преимуществ IPv6
является адресное пространство в 128 бит, что даёт возможность создать
около  $3,4 \times 10^{38}$  уникальных IP-адресов. Это практически неисчерпаемый
резерв, особенно по сравнению с IPv4, где доступно лишь около 4,3
миллиарда адресов. Такое расширение открывает новые горизонты для
развития глобальной сети.</p>

  <p>Это особенно важно в контексте интернета вещей
(IoT), где каждое устройство – будь то «умная» лампочка, датчик
температуры или автомобиль – требует собственного IP-адреса. IPv6
позволяет присваивать уникальный публичный адрес каждому устройству,
что упрощает их управление, мониторинг и прямую коммуникацию, особенно
в масштабируемых автоматизированных системах.</p>
</div>
<div id="ipsec" class="content-block">
  <hr>
  <h3>Встроенная поддержка IPsec</h3>
  <p>В отличие от IPv4, где IPsec был опцией и
применялся в ограниченных сценариях, IPv6 нативно поддерживает IPsec
как неотъемлемую часть. Это означает, что каждый узел в IPv6-сети
потенциально может использовать механизмы шифрования, аутентификации и
защиты целостности передаваемых данных.</p>

  <p>IPsec включает два основных режима –
транспортный и туннельный. Они позволяют защищать либо только полезную
нагрузку пакета, либо весь IP-пакет целиком, создавая защищённый
«туннель». Благодаря встроенности IPsec, IPv6 создаёт базу для более
безопасной сетевой архитектуры, особенно в корпоративных сетях, VPN-
решениях и инфраструктурах, требующих высокого уровня доверия.</p>
</div>
<div id="routing" class="content-block">
  <hr>
  <h3>Улучшенная маршрутизация</h3>
  <p>IPv6 предлагает более эффективную
маршрутизацию, благодаря иерархическому распределению адресов.
Структура адресов позволяет сгруппировать их по регионам, провайдерам
или организациям, что упрощает агрегацию маршрутов и уменьшает размер
таблиц маршрутизации на маршрутизаторах и ускоряет передачу
данных.</p>

  <p>IPv6 также упрощает обработку пакетов за счёт
упрощённого заголовка по сравнению с IPv4. Заголовки IPv6 имеют
фиксированную длину и стандартизированную структуру, что облегчает
работу маршрутизаторов и повышает общую производительность сети. Это
особенно важно в условиях высокой нагрузки и требовательных к скорости
приложениях.</p>
</div>
<div id="qos" class="content-block">
  <hr>
```

Метки потоков и QoS

Одним из новых элементов в заголовке IPv6 стал параметр Flow Label – метка потока. Она используется для идентификации трафика, принадлежащего к одному логическому потоку (например, видеозвонок, стриминг, игра), даже если пакеты передаются разными маршрутами. Это позволяет маршрутизаторам более точно и быстро обрабатывать такие потоки, обеспечивая стабильную доставку.

Функциональность меток потоков дополняется возможностями QoS (Quality of Service) – системой приоритезации трафика. Она позволяет выделять приоритетные пакеты (например, голосовой трафик) и обеспечивать им минимальные задержки, даже в условиях перегрузки сети. Это особенно важно для бизнес-приложений, онлайн-конференций, VoIP и других сервисов с критичными требованиями к задержкам и стабильности.

Отсутствие необходимости NAT

В IPv4 из-за ограниченности адресов широко применялась технология NAT (Network Address Translation), которая позволяла скрывать множество устройств за одним публичным IP-адресом. Однако NAT создаёт сложности в работе протоколов, требующих прямого соединения между устройствами, таких как VoIP, P2P и игровые приложения.

IPv6 устраняет эту проблему благодаря своему адресному пространству. Каждое устройство может получить уникальный глобальный IP-адрес, что позволяет устанавливать прямое соединение между узлами, упрощает настройку приложений и повышает прозрачность сетевого взаимодействия. Это также делает диагностику и мониторинг сетей более предсказуемыми и понятными.

Упрощённое сетевое взаимодействие

IPv6 был разработан с целью упрощения архитектуры сети, избавления от временных решений и упрощения конфигурации. Например, благодаря функции Stateless Address Autoconfiguration (SLAAC), устройства могут автоматически получать адреса, не требуя ручной настройки или наличия DHCP-сервера. Это особенно удобно в домашних и мобильных сетях.

Также IPv6 исключает широковещание (broadcast), который часто использовался в IPv4, но приводил к излишней нагрузке на сеть. Вместо него применяются более точные механизмы доставки, такие как multicast и anycast. В результате сети на IPv6 становятся более управляемыми, гибкими и надёжными, с меньшей зависимостью от костыльных решений и промежуточных технологий.


```

    <div v-if="activeTab === 'addressing'" id="addressing"
class="section">
    <div class="container">
        <h2>Адресация и автоконфигурация</h2>
        <div class="subnav">
            <a href="#slaac"
@click.prevent="scrollTo('slaac')">SLAAC</a>
            <a href="#eui-64" @click.prevent="scrollTo('eui-
64')">EUI-64</a>
            <a href="#local-addresses"
@click.prevent="scrollTo('local-addresses')">Локальные адреса</a>
            <a href="#router"
@click.prevent="scrollTo('router')">Router Solicitation /
Advertisement</a>
            <a href="#dhcpv6"
@click.prevent="scrollTo('dhcpv6')">DHCPv6</a>
        </div>
        <div id="slaac" class="content-block">
            <hr>
            <h3>SLAAC: Stateless Address
Autoconfiguration</h3>
            <p>Один из ключевых механизмов — это SLAAC
(Stateless Address Autoconfiguration). С его помощью устройство может
автоматически сгенерировать свой IPv6-адрес на основе сетевого
префикса, полученного от маршрутизатора, и собственных данных, без
обращения к серверу DHCP. SLAAC значительно упрощает процесс
подключения устройств, особенно в домашних или мобильных сетях, где
участие администратора минимально или вовсе отсутствует.</p>
            <p>Когда устройство подключается к сети, оно
прослушивает сообщения Router Advertisement (RA), которые рассылаются
маршрутизаторами. Эти сообщения содержат информацию о префиксе сети,
времени жизни маршрута и других параметрах. Используя эти данные,
устройство формирует полный IPv6-адрес.</p>
        </div>
        <div id="eui-64" class="content-block">
            <hr>
            <h3>EUI-64</h3>
            <p>Для генерации адреса хоста в SLAAC часто
используется метод EUI-64, который позволяет создать уникальную
идентификаторную часть адреса на основе MAC-адреса устройства. Этот
механизм вставляет в MAC-адрес специальные биты и расширяет его до 64-
битного идентификатора, который затем добавляется к сетевому
префиксу.</p>
            <p>EUI-64 обеспечивает автоматическое формирование
уникального адреса, но с точки зрения конфиденциальности может быть
уязвим, так как MAC-адрес — это аппаратный идентификатор, который
может использоваться для отслеживания устройства. В ответ на это в
IPv6 были введены temporary addresses, которые меняются со временем и
затрудняют слежку.</p>
        </div>
        <div id="local-addresses" class="content-block">

```

```
<hr>
<h3>Локальные адреса</h3>
<p>IPv6 предусматривает несколько типов адресов,
среди которых особое место занимают локальные (link-local) и
уникальные локальные адреса (ULA).</p>
<p>Link-local адреса автоматически присваиваются
каждому интерфейсу и используются только в пределах одной локальной
сети. Они начинаются с префикса FE80::/10 и необходимы, например, для
взаимодействия с маршрутизаторами, отправки RA/RS-сообщений и
организации протоколов маршрутизации.</p>
<p>ULA (Unique Local Addresses), начинающиеся с
FC00::/7, являются аналогами частных адресов IPv4 (например,
192.168.x.x) и используются внутри организации. Эти адреса не
маршрутизируются в глобальном интернете, но могут использоваться для
внутренней связности, особенно в изолированных или VPN-сетях.</p>
</div>
<div id="router" class="content-block">
<hr>
<h3>Router Solicitation / Advertisement</h3>
<p>Автоконфигурация в IPv6 строится на
взаимодействии между хостами и маршрутизаторами с помощью сообщений
Router Solicitation (RS) и Router Advertisement (RA). Когда новое
устройство появляется в сети, оно может отправить RS-запрос, чтобы
ускорить получение RA от маршрутизатора. RA-сообщение содержит
информацию о наличии SLAAC, адресах шлюзов, MTU и других
параметрах.</p>
<p>Это позволяет полностью автоматизировать
процесс подключения без необходимости настройки вручную или
использования дополнительных серверов. RA может также указывать,
требуется ли использование DHCPv6 для получения дополнительных
параметров, таких как адреса DNS-серверов.</p>
</div>
<div id="dhcpv6" class="content-block">
<hr>
<h3>DHCPv6</h3>
<p>Хотя SLAAC обеспечивает базовую
автоконфигурацию, в некоторых случаях может понадобиться получения
более точной информации – например, адресов DNS-серверов, доменных
имён или определённых политик. Для этого в IPv6 используется DHCPv6 –
расширенная версия DHCP, адаптированная под особенности нового
протокола.</p>
<p>DHCPv6 может работать в нескольких режимах:</p>
<ul>
<li><strong>Stateful</strong> – как и в IPv4,
сервер полностью управляет адресами и конфигурацией устройств.</li>
<li><strong>Stateless</strong> – сервер
предоставляет только дополнительные параметры (например, DNS), а адрес
хост получает через SLAAC.</li>
</ul>
```

<p>Это обеспечивает гибкость: администратор может комбинировать SLAAC и DHCPv6 в зависимости от архитектуры сети и требований безопасности или мониторинга.</p>

</div>

</div>

</div>

<div v-if="activeTab === 'adaptation'" id="adaptation" class="section">

<div class="container">

<h2>Адаптация и статистика</h2>

<div class="subnav">

@click.prevent="scrollTo('global')">Глобальный процент

@click.prevent="scrollTo('pioneers')">Крупнейшие страны-пионеры

@click.prevent="scrollTo('milestones')">Вехи внедрения

</div>

<div id="global" class="content-block">

<hr>

<h3>Глобальный процент IPv6-пользователей</h3>

<p>Переход от IPv4 к IPv6 происходит постепенно, и на данный момент всё больше интернет-провайдеров, сайтов и пользователей используют новый протокол. По данным различных мониторингов (например, Google, APNIC, Akamai), к 2025 году более 40% мирового интернет-трафика уже осуществляется через IPv6. Однако уровень внедрения заметно варьируется от страны к стране.</p>

<p>Эта статистика отражает как готовность инфраструктуры (сетевых устройств, провайдеров, контента), так и государственную политику в сфере цифрового развития. Темпы роста остаются стабильными, и по прогнозам, доля IPv6-трафика будет расти, особенно с увеличением числа IoT-устройств и развитием 5G/6G-сетей.</p>

</div>

<div id="pioneers" class="content-block">

<hr>

<h3>Крупнейшие страны-пионеры</h3>

<p>Лидерами по внедрению IPv6 являются Индия, Бельгия, Германия, США и Япония. Например, Индия демонстрирует одну из самых высоких долей IPv6-пользователей – более 65% трафика. Это стало возможным благодаря активной позиции национальных провайдеров и массовому подключению мобильных устройств с поддержкой IPv6.</p>

<p>В Бельгии переход произошёл на ранних этапах, чему способствовало техническое лидерство провайдера Telenet. В США крупные провайдеры, такие как Comcast и Verizon, также массово внедрились поддержку IPv6, особенно для мобильного интернета. В Германии

Deutsche Telekom обеспечил активную миграцию пользователей на новую адресацию.</p>

```
<div class="slider-container mt-5">
  <div class="slider" :style="{ transform: `translateX(-
${currentSlide * 100}%)` }">
    
    
    
    
    
  </div>

  <button class="slider-btn prev" @click="prevSlide"></button>
  <button class="slider-btn next" @click="nextSlide"></button>
</div>

</div>
<div id="milestones" class="content-block">
  <hr>
  <h3>Вехи внедрения</h3>
  <p>История перехода к IPv6 включает ряд значимых
вех. Одной из них стало официальное исчерпание адресного пространства
IPv4, объявленное IANA в 2011 году. Вслед за этим началась активная
подготовка операторов связи и крупных корпораций к переходу.</p>
  <p>Следующей вехой стал World IPv6 Launch Day (6
июня 2012 года), когда крупнейшие интернет-компании, такие как Google,
Facebook, Yahoo и Cisco, включили поддержку IPv6 на постоянной основе.
Это событие стало глобальным сигналом к действию и дало толчок к
ускоренному принятию протокола.</p>
  <p>С тех пор внедрение шло поэтапно – сначала в
ядре магистральных сетей, затем у операторов доступа, хостинг-
провайдеров и, наконец, у конечных пользователей. Сегодня многие
государственные инициативы включают поддержку IPv6 как обязательную
часть цифровых стратегий.</p>
</div>

</div>
</div>

<div v-if="activeTab === 'proposals'" id="proposals"
class="section">
  <div class="container">
    <h2>Предложения по улучшению</h2>
    <form @submit.prevent="submitForm">
      <div class="mb-3">
        <label for="name" class="form-
label">Имя</label>
```

```

        <input type="text" class="form-control"
id="name" v-model="form.name" required>
    </div>
    <div class="mb-3">
        <label for="email" class="form-
label">Email</label>
        <input type="email" class="form-control"
id="email" v-model="form.email" required>
    </div>
    <div class="mb-3">
        <label for="proposal" class="form-label">Ваше
предложение</label>
        <textarea class="form-control" id="proposal"
v-model="form.proposal" required></textarea>
    </div>
    <button type="submit" class="btn btn-
primary">Отправить</button>
    </form>
</div>
</div>

    <div class="modal fade" id="formModal" tabindex="-1" aria-
hidden="true">
        <div class="modal-dialog">
            <div class="modal-content">
                <div class="modal-header">
                    <h5 class="modal-title">{{ modal.title }}</h5>
                    <button type="button" class="btn-close" data-
bs-dismiss="modal" aria-label="Close"></button>
                </div>
                <div class="modal-body">
                    <p>{{ modal.message }}</p>
                </div>
                <div class="modal-footer">
                    <button type="button" class="btn btn-
secondary" data-bs-dismiss="modal">Закрыть</button>
                </div>
            </div>
        </div>
    </div>
</div>

    <script
src="https://cdn.jsdelivr.net/npm/vue@3.2.47/dist/vue.global.prod.js">
</script>
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bu
ndle.min.js"></script>
    <script src="app.js"></script>
</body>
</html>

```

ПРИЛОЖЕНИЕ В. CSS-КОД

```
body {
    font-family: 'Segoe UI', 'Roboto', sans-serif;
    background-color: #f4f1f7;
    color: #2c2c2c;
    line-height: 1.6;
}

/* Навбар */
.navbar {
    background-color: #4a4e69;
    position: sticky;
    top: 0;
    z-index: 1000;
    box-shadow: 0 2px 6px rgba(0, 0, 0, 0.2);
}

.nav-link {
    color: #fff !important;
    font-weight: 500;
    transition: color 0.3s ease;
}

.nav-link:hover,
.nav-link.active {
    color: #c9ada7 !important;
}

/* Разделы */
.section {
    padding: 70px 0;
    border-bottom: 1px solid #ddd;
    background-color: #ffffff;
    border-radius: 12px;
    margin: 20px;
    box-shadow: 0 3px 10px rgba(0, 0, 0, 0.06);
}

.section h2 {
    color: #4a4e69;
    margin-bottom: 30px;
    font-size: 2rem;
    text-align: center;
}

/* Поднавигация */
.subnav {
    margin-bottom: 30px;
    text-align: center;
}
```

```

.subnav a {
    color: #4a4e69;
    font-weight: 500;
    text-decoration: none;
    margin: 0 10px;
    padding-bottom: 4px;
    border-bottom: 2px solid transparent;
    transition: all 0.3s;
}

.subnav a:hover {
    border-color: #c9ada7;
    color: #c9ada7;
}

/* Слайдер */
.slider-container {
    position: relative;
    max-width: 850px;
    margin: 40px auto;
    border-radius: 12px;
    overflow: hidden;
    box-shadow: 0 6px 15px rgba(0, 0, 0, 0.1);
}

.slider {
    display: flex;
    transition: transform 0.5s ease-in-out;
}

.slide-img {
    flex: 0 0 100%;
    width: 100%;
    object-fit: cover;
}

/* Кнопки слайдера */
.slider-btn {
    position: absolute;
    top: 50%;
    transform: translateY(-50%);
    background-color: rgba(74, 78, 105, 0.8);
    color: #fff;
    border: none;
    padding: 12px 16px;
    cursor: pointer;
    font-size: 24px;
    border-radius: 50%;
    transition: background-color 0.3s;
}

```

```

.slider-btn:hover {
    background-color: rgba(201, 173, 167, 0.9);
}

.slider-btn.prev {
    left: 15px;
}

.slider-btn.next {
    right: 15px;
}

/* Модальные окна */
.modal-content {
    border-radius: 10px;
    background-color: #fff;
    border: 1px solid #ccc;
}

/* Контентные блоки */
.content-block {
    margin-bottom: 50px;
}

/* Форма */
input,
textarea {
    background-color: #fdfdfd;
    border: 1px solid #ccc;
    border-radius: 6px;
    padding: 10px;
    width: 100%;
    margin-bottom: 15px;
    transition: border-color 0.3s;
}

input:focus,
textarea:focus {
    border-color: #c9ada7;
    outline: none;
}

button, .btn {
    background-color: #4a4e69;
    color: #fff;
    border: none;
    padding: 10px 20px;
    border-radius: 8px;
    font-weight: 500;
    transition: background-color 0.3s ease;
}

```



```
}

button:hover,
.btn:hover {
    background-color: #c9ada7;
    color: #fff;
}

/* Адаптивность */
@media (max-width: 1200px) {
    .section {
        padding: 50px 20px;
    }

    .slider-container {
        max-width: 650px;
    }
}

@media (max-width: 800px) {
    .navbar-nav {
        text-align: center;
    }

    .section {
        padding: 40px 20px;
    }

    .slider-container {
        max-width: 100%;
    }
}

@media (max-width: 550px) {
    .navbar-brand {
        font-size: 1.3rem;
    }

    .section h2 {
        font-size: 1.4rem;
    }

    .subnav a {
        display: block;
        margin: 8px 0;
    }
}
```