ГУАП

КАФЕДРА № 42

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| канд. техн. наук, доцент |  |  |  | О. И. Красильникова |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ПОЯСНИТЕЛЬНАЯ ЗАПИСКА К ПРАКТИКО-ОРИЕНТИРОВАННОМУ ПРОЕКТУ |
| ТОПОЛОГИИ ЛОКАЛЬНЫХ СЕТЕЙ |
| по курсу: |
|  |
| ИТ-МОДУЛЬ "ОСНОВЫ FRONTEND-РАЗРАБОТКИ" |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | 1245 |  | 31.05.2024 |  | Г. С. Куранов |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2025

1. Введение

* Тема проекта

Разработанный одностраничный веб-сайт (SPA) посвящен изучению топологий локальных сетей: звезда, кольцо, шина и их смешанных вариантов. Проект предназначен для студентов IT-специальностей, системных администраторов и всех, кто интересуется сетевыми технологиями.

* Актуальность

1. Локальные сети — основа корпоративной и домашней инфраструктуры.
2. Понимание топологий критично для проектирования надежных сетей.
3. Интерактивный формат подачи материала улучшает усвоение информации.
4. Постановка задачи

* *Описание проекта и требования*  
  Тема 12.  
  Веб-сайт «Топологии локальных сетей»

Цель: cоздать одностраничное приложение (SPA) с интерактивным интерфейсом, адаптивным дизайном и функциональными компонентами.

* *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).

Проект должен представлять собой одностраничный веб-сайт. Обязательными элементами интерфейса должны быть несколько вкладок (табов), оформленных как навигационное меню, в соответствии с которыми контент сайта должен быть разделен на несколько частей.

Навигационное меню должно быть фиксированным и не прокручиваться вместе со страницей, а нажатие на ссылки должны вести на соответствующий компонент без перезагрузки страницы (вкладки должны открываться без перезагрузки страницы).

В работе должны быть использованы современные технологии разработки web-сайтов, которые были изучены в процессе теоретического обучения по данному ИТ-модулю. Обязательным является использование HTML5, технологии CSS3, а также программирования на JavaScript, Vue,js.

При обеспечении адаптивности необходимо предусмотреть 3 контрольные точки (например, 1200px, 800px, 550px).

Следует обратить внимание на продуманность дизайна веб-страницы, ее архитектуры, на удобство навигации, привлекательность подачи информации.

На одну из вкладок следует добавить форму с необходимыми элементами (для отдельных полей должна быть предусмотрена обязательность заполнения). При нажатии на кнопку отправки должно появляться модальное окно с отображением успеха отправки (условие отправки можно не выполнять, т.е. страница не должна перезагружаться), либо с указанием ошибки при заполнении (например, пустое поле) через использование регулярных выражений JavaScript.

Еще раз обращаем внимание на то, что любая перезагрузка страницы с использованием переходов, отправок форм строго запрещена! В проекте должна существовать только одна веб-страница!

Весь функционал и реактивность страницы должны быть описаны с использованием методов и хук Vue.js.

На странице обязательно должен присутствовать компонент «слайдер изображений в галерее». Для смены картинок необходимо наличие кнопок (вправо/влево или </> и т.п.). Кнопки должны иметь стилевое оформление. Анимация перехода приветствуется, но не обязательна.

При выполнении работы разрешается использование библиотеки Bootstrap 5-ой версии.

Дизайн сайта можно придумать самому, но приветствуется также и использование бесплатных макетов Figma (верстка исключительно через position: absolute запрещена!)

* *Состав команды исполнителей. Перечень и распределение решаемых задач.*

Исполнители: студенты 1245 Куранов Григорий, Лапин Алексей, Лычагин Василий, Семенов Вадим.  
Решаемые задачи и распределение:

1. Архитектор проекта: Куранов Григорий

* Создание структуры проекта (Vue CLI, настройка роутинга).
* Распределение задач между участниками.
* Проверка кода на соответствие требованиям.
* Развертка на сервере

2. Frontend-разработчик: Лапин Алексей(Vue.js, UX/UI)

* Разработка фиксированного меню и шаблона страниц.
* Создание адаптивных layout (3 контрольные точки).
* Реализация слайдера-галереи с кнопками переключения.

3. Frontend-разработчик (Vue Router и формы): Лычагин Василий

* Настройка Vue Router (вкладки без перезагрузки).

4. Тестировщик / Документатор: Семенов Вадим

* Проверка кроссбраузерности (Chrome, Firefox, Edge).
* Тестирование адаптивности на разных устройствах.
* Создание инструкции по запуску проекта.

1. Проектирование сайта

3.1 Выбор программных средств и технологий для решения поставленных задач

|  |  |
| --- | --- |
| Технология | Назначение |
| Vue.js 3 | Реактивность, компонентный подход. |
| Vue Router | Навигация без перезагрузки. |
| CSS3 (Flex/Grid) | Адаптивный дизайн. |
| JavaScript | Логика слайдера и формы. |

Макет сайта:

* Header: Логотип + фиксированное меню.  
    
  Изображение выглядит как текст, снимок экрана

  Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 1 – Главное меню

* Main: Контент по вкладкам (Главная, Топологии, Галерея, Контакты).  
    
  Изображение выглядит как текст, снимок экрана, веб-страница, программное обеспечение

  Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 2 – Вкладка топологии(виды)

* Footer: Информация об авторе.  
    
  Изображение выглядит как текст, программное обеспечение, дизайн, снимок экрана

  Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 3 – footer и слайдер

1. Реактивность с Vue.js

Роль Vue.js в проекте

* Компонентный подход: Разделение на Header, Footer, Gallery, Contacts.
* Роутинг: Переключение вкладок через <router-view>.
* Состояние: Управление данными (например, текущий слайд в галерее).
* ***Вот пример кода, в котором настраивались роутинги:***

import { createRouter, createWebHistory } from 'vue-router';

import Home from './views/Home.vue';

const routes = [

{ path: '/', component: Home },

// ...другие маршруты

];

const router = createRouter({

history: createWebHistory(),

routes,

});

* ***Далее приведен код для слайдера (Composition API)***

data() {

return {

images: [

{ file: busImage, alt: 'Топология шина', caption: 'Топология шина: все устройства подключены к одному кабелю.' },

{ file: starImage, alt: 'Топология звезда', caption: 'Топология звезда: устройства подключены к центральному концентратору или коммутатору.' },

{ file: ringImage, alt: 'Топология кольцо', caption: 'Топология кольцо: устройства соединены в кольцевую структуру.' }

],

currentSlide: 0

}

},

methods: {

nextSlide() {

this.currentSlide = (this.currentSlide + 1) % this.images.length

},

prevSlide() {

this.currentSlide = (this.currentSlide - 1 + this.images.length) % this.images.length

}

}  
  
Изображение выглядит как снимок экрана, диаграмма, дизайн

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 4 – пример работы слайдера

* ***Для валидации код был написан следующий код***  
  methods: {

validateEmail() {

const emailRegex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;

if (!emailRegex.test(this.email)) {

this.emailError = 'Введите корректный email (например: user@example.com)';

return false;

}

this.emailError = '';

return true;

},

submitForm() {

// Проверяем валидность email перед отправкой

if (!this.validateEmail()) {

this.modalMessage = this.emailError;

this.showModal = true;

return;

}

if (!this.name || !this.email || !this.message) {

this.modalMessage = 'Заполните все поля!';

} else {

this.modalMessage = 'Сообщение отправлено!';

}

this.showModal = true;

}

}

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 5 – сообщение о неправильном заполнении почты при попытки уйти с неправильно введенного поля

1. Заключение

В ходе разработки SPA-приложения "Топологии локальных сетей" на Vue.js были успешно реализованы все поставленные задачи. Проект представляет собой полноценное одностраничное приложение с клиентской маршрутизацией через Vue Router, что обеспечивает плавные переходы между разделами без перезагрузки страницы. Особое внимание было уделено адаптивному дизайну - реализованы три контрольные точки (1200px, 800px и 550px), что гарантирует корректное отображение на различных устройствах.

Ключевые функциональные элементы, такие как слайдер-галерея с изображениями топологий сетей и форма обратной связи с комплексной валидацией, были тщательно проработаны. Форма включает проверку обязательных полей, валидацию email с использованием регулярных выражений и отображение соответствующих сообщений об ошибках. Для улучшения пользовательского опыта реализовано модальное окно, информирующее о результатах отправки формы.

В процессе работы были приобретены ценные практические навыки работы с современными фронтенд-технологиями, включая Vue 3 (Composition API), Vue Router, а также методы создания адаптивных интерфейсов. Полученный опыт охватывает все этапы разработки - от проектирования архитектуры приложения до тестирования и оптимизации финального продукта.

Разработанное SPA-приложение успешно решает поставленные образовательные задачи, предоставляя пользователям удобный и интерактивный способ изучения топологий локальных сетей. Проект может служить основой для дальнейшего развития - добавления новых функций, таких как анимации переходов, тестирование с в Jest/Vitest, или интеграции с другими backend-сервисами. Полученные в ходе работы знания и навыки открывают широкие возможности для создания сложных веб-приложений профессионального уровня.