
кафедра

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ
ПРЕПОДАВАТЕЛЬ

должность, уч. степень, звание

подпись, дата

инициалы, фамилия

ОТЧЕТ О ПРАКТИКО-ОРИЕНТИРОВАННОМ ПРОЕКТЕ

по курсу: ИТ-модуль «JavaScript, его библиотеки и фреймворки в Frontend-разработке»

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. №

подпись, дата

инициалы, фамилия

Санкт-Петербург 2024

Введение

Сайт 'SmartRozetki' посвящен технологии умных розеток, которые являются ключевым элементом современных систем 'умного дома'. В условиях быстро развивающихся технологий и возрастающего интереса к автоматизации домашних услуг, эта тема приобретает особую актуальность. Проект представляет собой удобный веб-интерфейс, предоставляющий информацию о функциях, преимуществах и применении умных розеток в повседневной жизни. Цель проекта - облегчить пользователям доступ к информации об умных розетках и продемонстрировать возможности их использования в различных сценариях домашнего обихода.

Постановка задачи

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!)

Выбранный вариант:

6. Веб-сайт «Умные розетки для систем интернета вещей»

Состав команды исполнителей. Перечень и распределение решаемых задач:

1 – Захаров Андрей Сергеевич 4133К

Изучение и подготовка: Изучение основ веб-разработки, включая HTML, CSS, JavaScript и Vue.js. Подготовка необходимого программного обеспечения и инструментов для разработки.

Проектирование и планирование: Определение структуры и дизайна сайта, планирование функциональности и пользовательского интерфейса.

Разработка: Непосредственное создание веб-сайта, включая написание кода для различных компонентов (например, навигационной панели, формы обратной связи, слайдера изображений и т.д.), стилизацию и добавление интерактивности с использованием Vue.js.

Тестирование и отладка: Проверка работоспособности сайта, исправление обнаруженных ошибок и улучшение производительности.

Запуск и развертывание: Запуск и настройка локального сервера.

Проектирование сайта

Выбор программных средств и технологий для решения поставленных задач:

Программные Средства:

Visual Studio Code: Использован в качестве основной интегрированной среды разработки (IDE) для написания и редактирования кода.

Браузеры Safari и Chrome: Safari использовался в качестве основного браузера для разработки и тестирования сайта, а Chrome служил дополнительным инструментом для кросс-браузерной проверки.

Аппаратные Средства:

4K Монитор: Использование монитора с высоким разрешением (3840x2160 пикселей) позволило детально оценить визуальные аспекты сайта и убедиться в корректности отображения элементов интерфейса. Но есть вероятность недочетов при маленьком разрешении окна сайта.

Технологии Разработки:

HTML5, CSS3, и JavaScript: Основа создания веб-сайта. HTML5 использовался для структурирования контента, CSS3 – для стилизации, а JavaScript – для реализации интерактивных функций.

Vue.js: Фреймворк JavaScript, который обеспечивал реактивность интерфейса и упрощал управление состоянием приложения.

Тестирование и Отладка:

Различные инструменты разработчика в браузерах (например, DevTools в Safari) использовались для тестирования, отладки.

Макет сайта:

Общая Концепция и Дизайн:

Сайт разработан с использованием современных веб-стандартов, включая HTML5, CSS3 и JavaScript.

Основной цветовой палитрой сайта являются нейтральные тона, создающие ощущение технологичности и современности.

Дизайн сайта минималистичен, что способствует легкости восприятия информации пользователями.

Структура и Навигация:

Сайт состоит из нескольких основных разделов, доступ к которым обеспечивается через навигационное меню.

Главная страница включает информацию о преимуществах умных розеток, их функциональности и способах использования.

Раздел "Обратная связь" позволяет пользователям оставлять сообщения и комментарии.

Визуальное Оформление:

Используется адаптивный дизайн, обеспечивающий корректное отображение сайта на различных устройствах и экранах.

Включены высококачественные изображения умных розеток, которые демонстрируют их в интерьере.

Применяется современная типографика, улучшающая читаемость текста.

Интерактивность и Функциональность:

Пользователи могут взаимодействовать с элементами сайта, такими как формы обратной связи и слайдеры изображений.

Сайт оптимизирован для быстрой загрузки и плавной работы на всех современных браузерах.

Реактивность сайта с использованием Vue.js

Vue.js — это прогрессивный JavaScript-фреймворк, используемый для создания интерактивных пользовательских интерфейсов. Особенно эффективен он оказался в создании одностраничных приложений и интерфейсов, где требуется высокая реактивность и взаимодействие с пользователем. Vue.js облегчает создание динамических приложений, позволяя легко изменять и обновлять данные без необходимости перезагрузки страницы.

В разработанном сайте Vue.js использовался для создания динамического и интерактивного интерфейса. Каждый компонент сайта, такой как навигационная панель, форма обратной связи, слайдер изображений и информационные разделы, был создан как отдельный компонент Vue. Это позволило упростить управление состоянием и логикой каждой части сайта, делая его более модульным и легким для обслуживания.

Функциональность реактивности Vue.js проявляется в нескольких аспектах сайта:

Двустороннее связывание данных (Two-way Data Binding): На сайте использовалась директива v-model, обеспечивающая синхронизацию данных между пользовательским интерфейсом и моделью данных. Это особенно важно в формах, где ввод пользователя непосредственно отражается в данных приложения.

Реактивное обновление элементов интерфейса: При изменении данных в компонентах Vue, связанный пользовательский интерфейс автоматически обновляется. Это позволяет

пользователям видеть изменения в реальном времени, например, при вводе текста в форме обратной связи или при переключении между слайдами в слайдере изображений.

Компонентный подход: Каждый элемент сайта разбит на отдельные компоненты Vue, что облегчает управление, тестирование и повторное использование кода. Например, слайдер изображений и форма обратной связи являются отдельными компонентами, которые могут быть легко интегрированы в другие части сайта или даже в другие проекты.

Использование директив и методов Vue: На сайте активно использовались встроенные директивы Vue, такие как `v-for` для рендеринга списков, `v-if` и `v-show` для условного рендеринга элементов интерфейса, а также пользовательские методы для обработки событий, таких как отправка формы.

Таким образом, использование Vue.js в разработке сайта позволило создать гибкий, интерактивный и легко поддерживаемый пользовательский интерфейс.

Описание взаимодействия с элементами на странице с использованием функционала Vue

В разработанном SPA (Single Page Application) с использованием Vue.js реализованы различные методы, компоненты, модули и хуки, которые обеспечивают динамическое взаимодействие с элементами на странице и повышают общую реактивность приложения. Ниже представлены примеры их использования с комментариями:

Компонент AppNavbar (файл AppNavbar.vue):

- Этот компонент создает навигационное меню сайта.
- Используется Vue Router для маршрутизации и переключения между разными частями сайта без перезагрузки страницы.
- Код включает слоты для ссылок, обеспечивающих навигацию.

Компонент FeedbackForm (файл FeedbackForm.vue):

- Этот компонент используется для обработки форм обратной связи.
- Использует двустороннюю привязку данных (`v-model`) для сбора ввода пользователя и локальные данные (`data()`) для хранения этих вводов.
- Реализует методы для валидации введенных данных и их отправки.

Компонент ImageSlider (файл ImageSlider.vue):

- Этот компонент создает слайдер изображений.
- Использует цикл `v-for` для отображения каждого изображения из массива и `v-if` для управления отображением модального окна.
- Включает методы для переключения слайдов и управления модальным окном.

Компонент SmartPlugInfo (файл SmartPlugInfo.vue):

- Этот компонент предоставляет информацию об умных розетках.
- Использует HTML-шаблоны и Vue-стили для представления содержимого.

Главный файл App.vue и маршрутизация (router/index.js):

- App.vue служит корневым компонентом, который объединяет все другие компоненты.
- Маршрутизация в router/index.js управляет переходами между страницами, поддерживая SPA-архитектуру.

Хуки жизненного цикла Vue.js:

- В разных компонентах используются хуки жизненного цикла Vue.js (например, created, mounted), чтобы выполнить определенные действия в ключевые моменты жизни компонента.

Реактивность данных:

- Vue.js используется для создания реактивного интерфейса, где изменения в данных автоматически отражаются в пользовательском интерфейсе без необходимости явного обновления.

В компоненте FeedbackForm.vue:

// Метод для отправки формы

```
submitForm() {  
  
  if (!this.formData.name || !this.formData.email || !this.formData.message) {  
  
    this.errorMessage = 'Пожалуйста, заполните все поля.';  
  
    return;  
  
  }  
  
  // ... дальнейшая логика ...  
  
}
```

В компоненте ImageSlider.vue:

// Метод для переключения слайда

```
nextSlide() {  
  
  this.currentSlide = (this.currentSlide + 1) % this.images.length;  
  
}
```

Эти примеры демонстрируют, как Vue.js используется для создания интерактивных, реактивных и модульных компонентов, улучшая пользовательский опыт и облегчая разработку и обслуживание кода.

Скриншоты, иллюстрирующие внешний вид и работоспособность разработанной страницы

На рисунках 1-8 представлен общий вид страницы с разными разрешениями

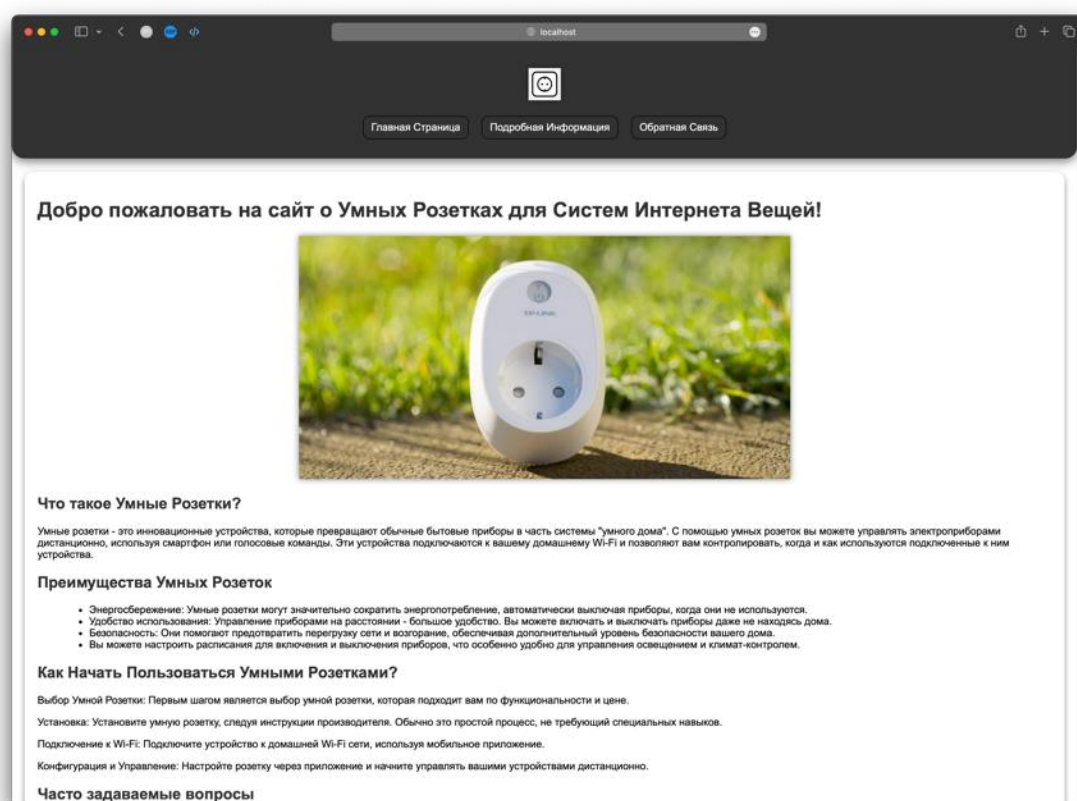


Рисунок 1 – общий вид страницы, шапка сайта

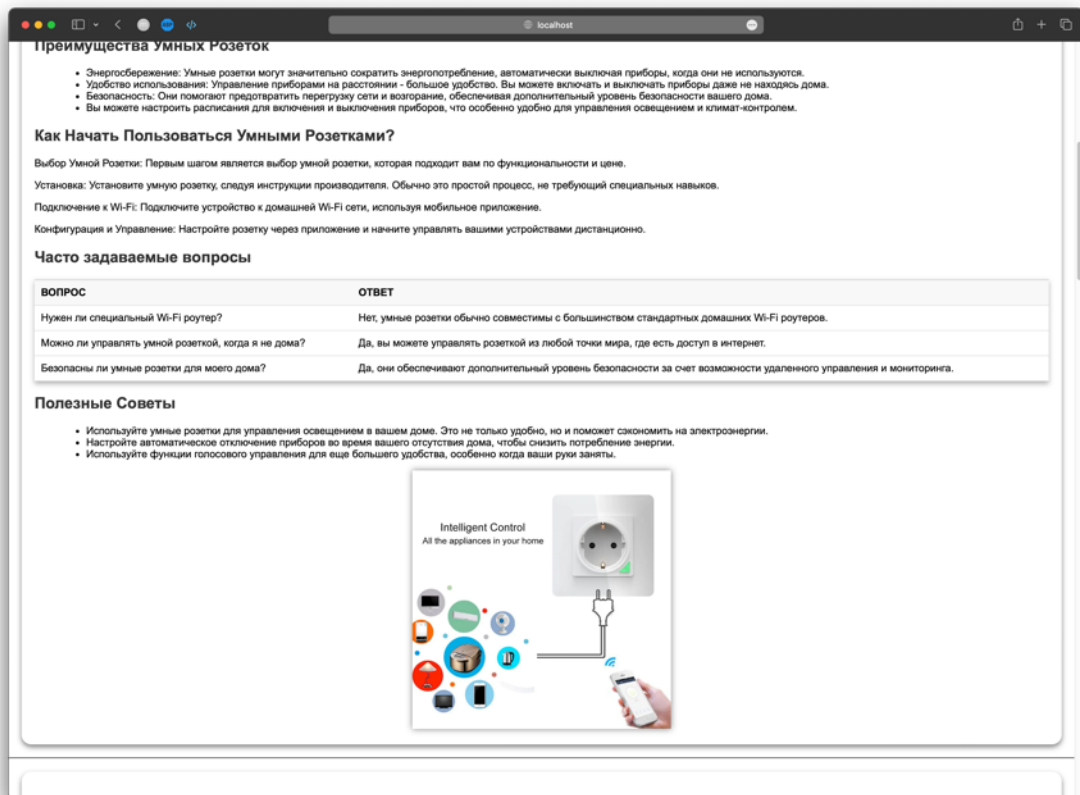


Рисунок 2 – общий вид страницы, таблица

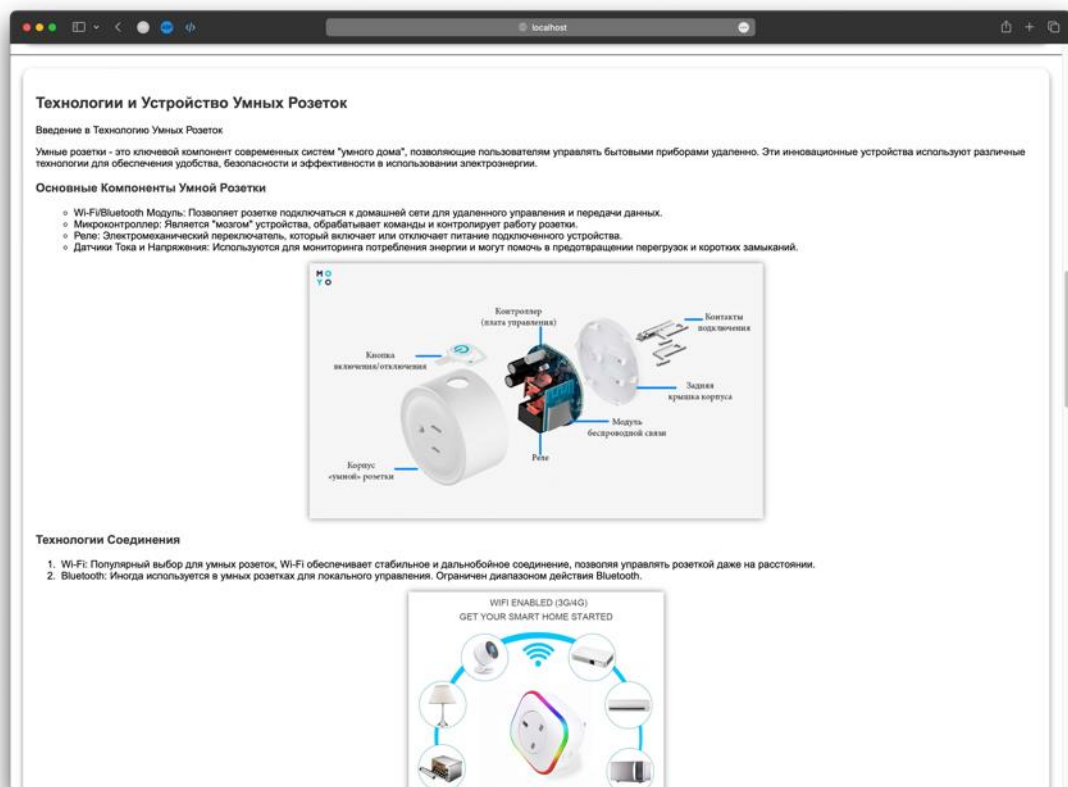


Рисунок 3 – общий вид страницы, списки

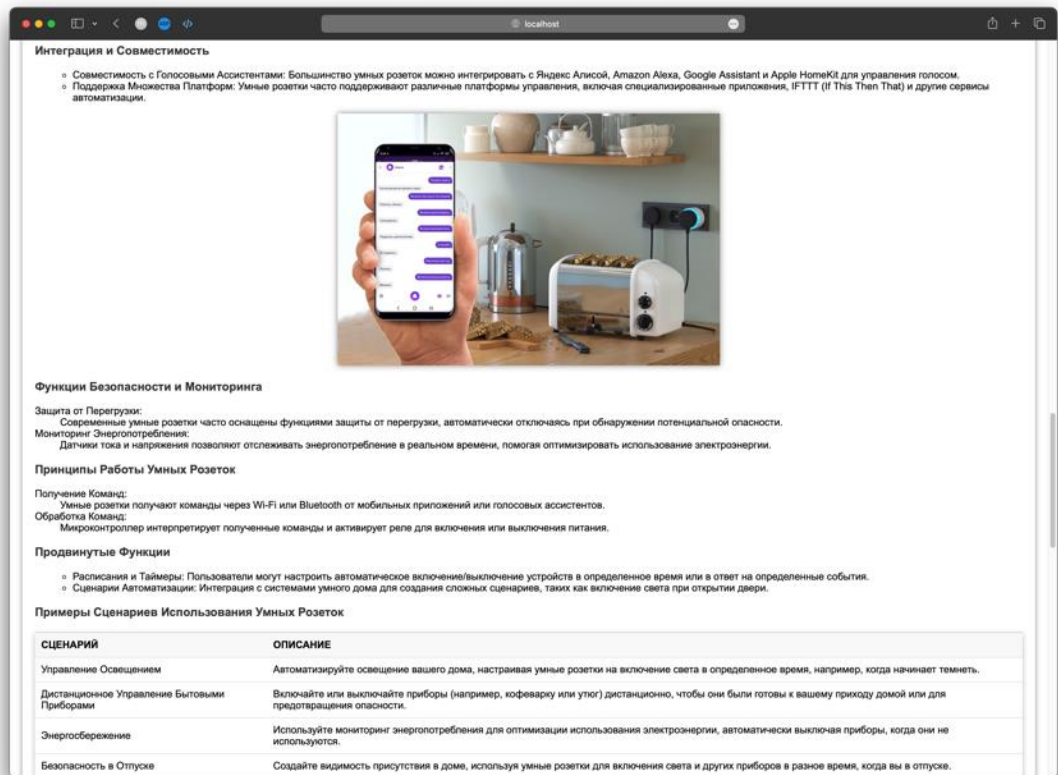


Рисунок 4 – общий вид страницы, блок с изображением, списком, таблицей

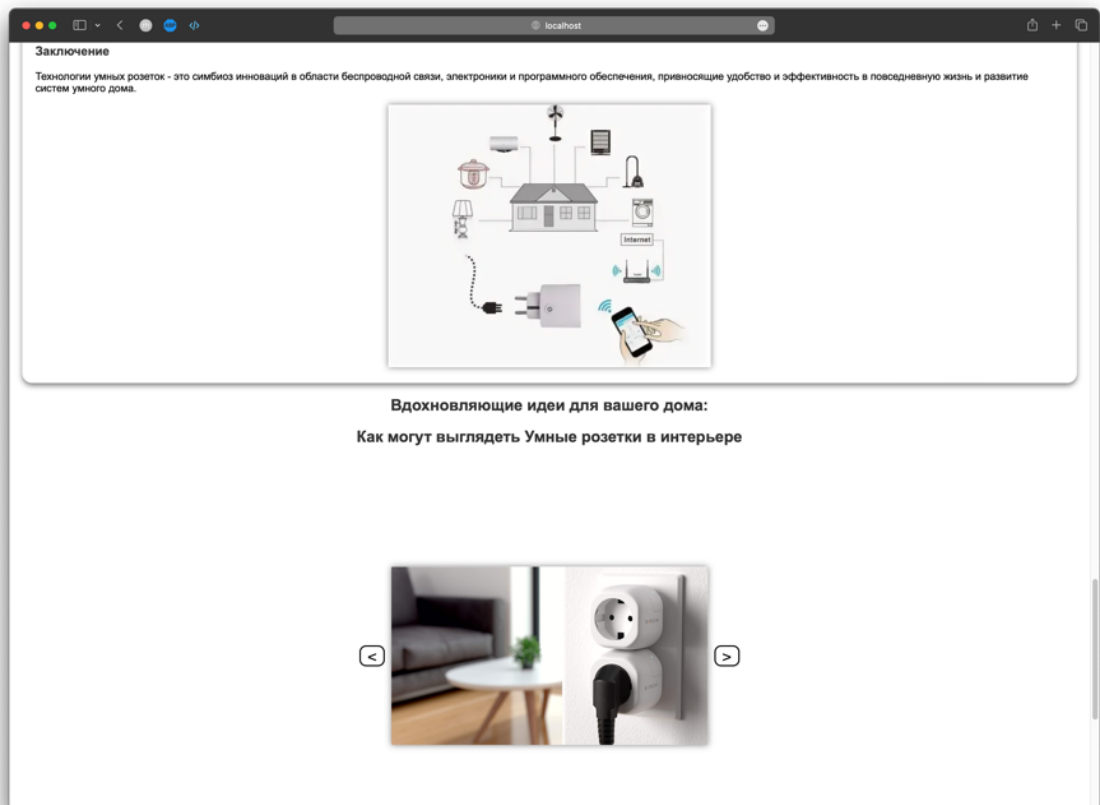


Рисунок 5 – вид слайдера изображений в галерее

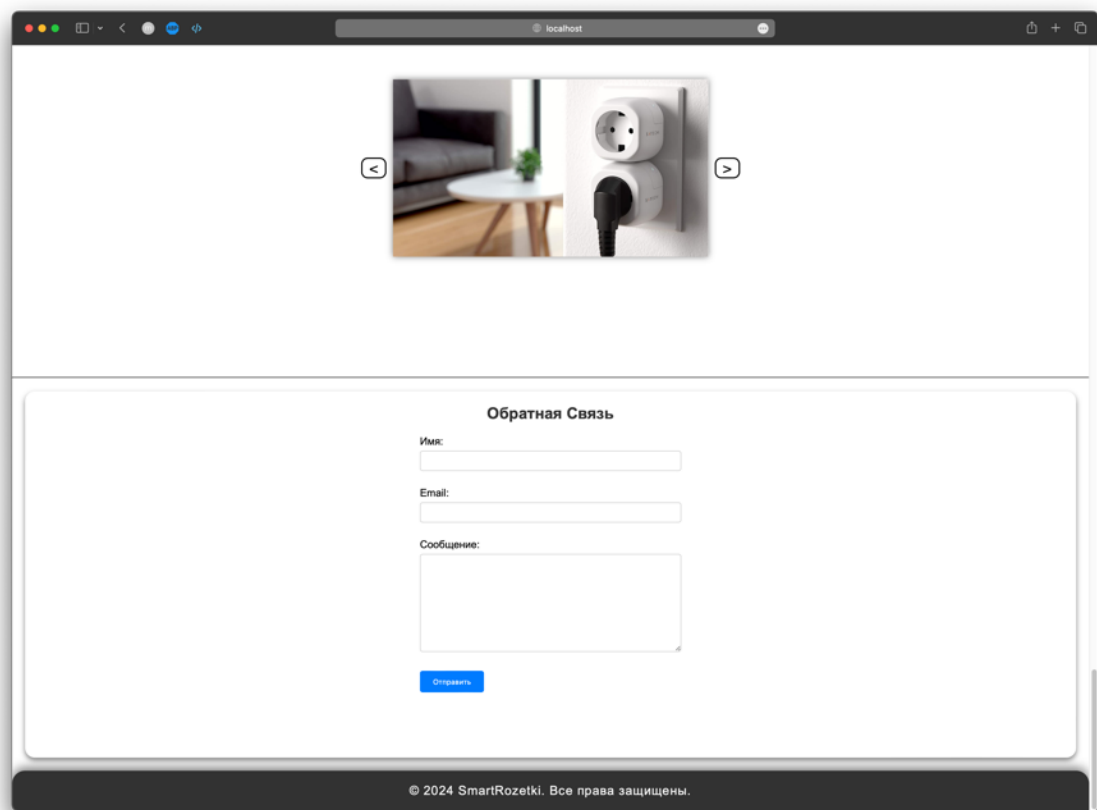


Рисунок 6 – вид подвала сайта и блока обратной связи

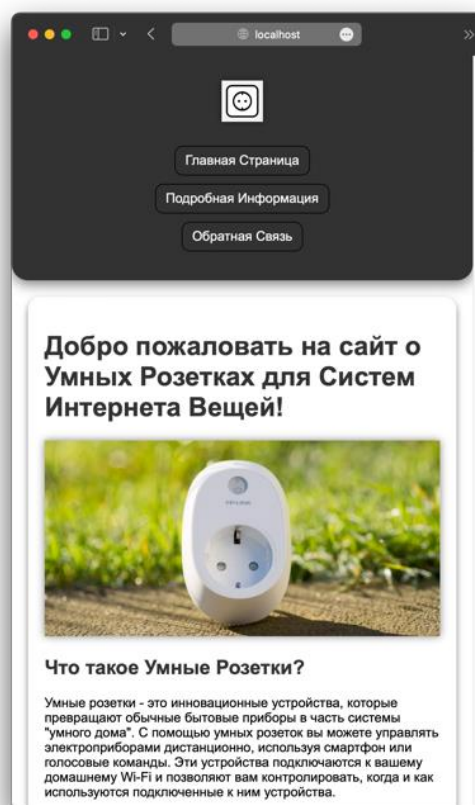


Рисунок 7 – вид страницы в разрешении для мобильных устройств, шапка сайта

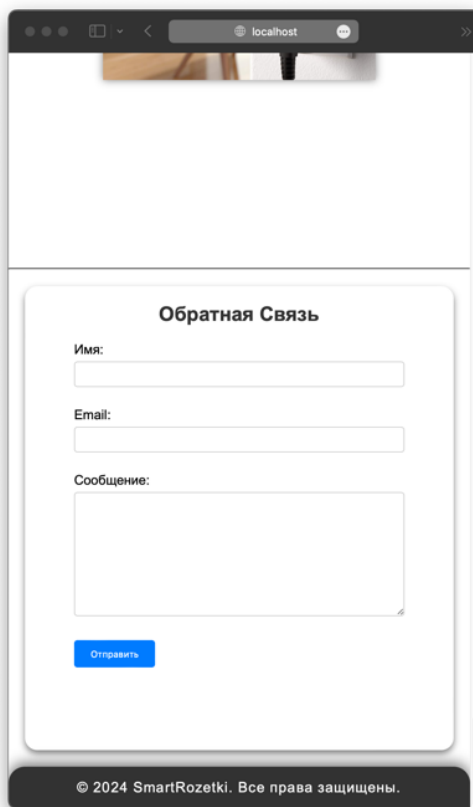


Рисунок 8 - вид страницы в разрешении для мобильных устройств, подвал сайта

На рисунках 9-14 представлены решения для оформления шапки сайта. При наведении на элемент происходит увеличение, а у текста так же смена цвета

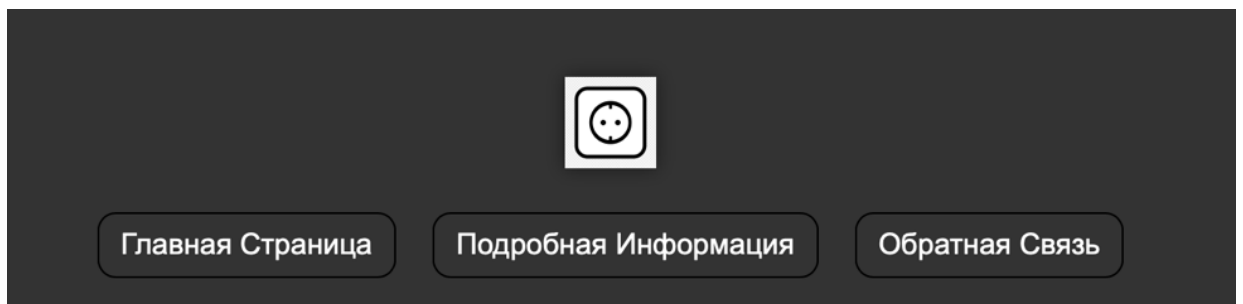


Рисунок 9 – изначальный вид шапки

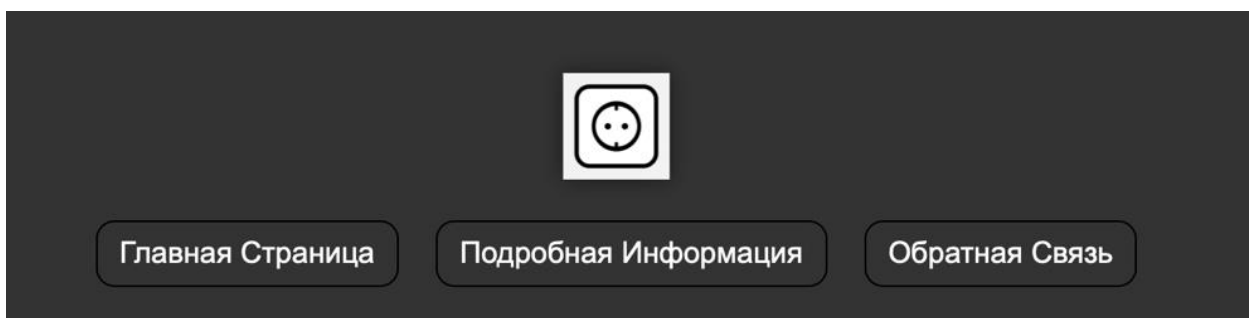


Рисунок 10 – вид логотипа при наведении на него (происходит увеличение)

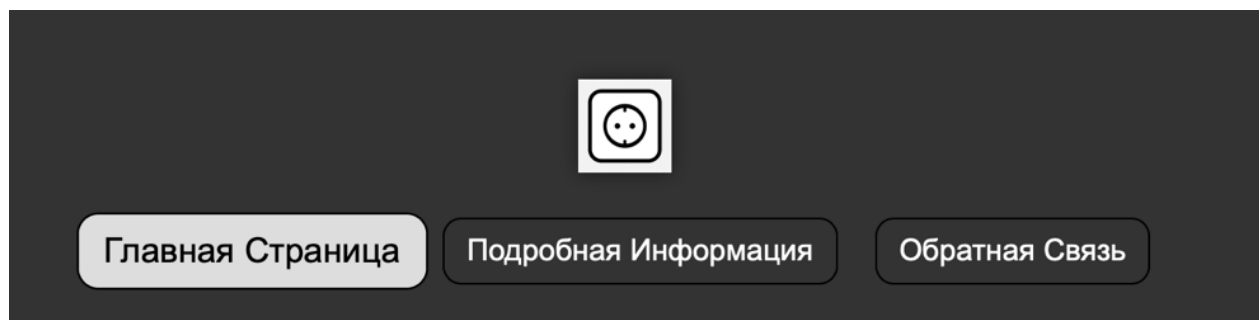


Рисунок 11 – наведение на кнопку Главная Страница – Увеличение, инверсия цветов

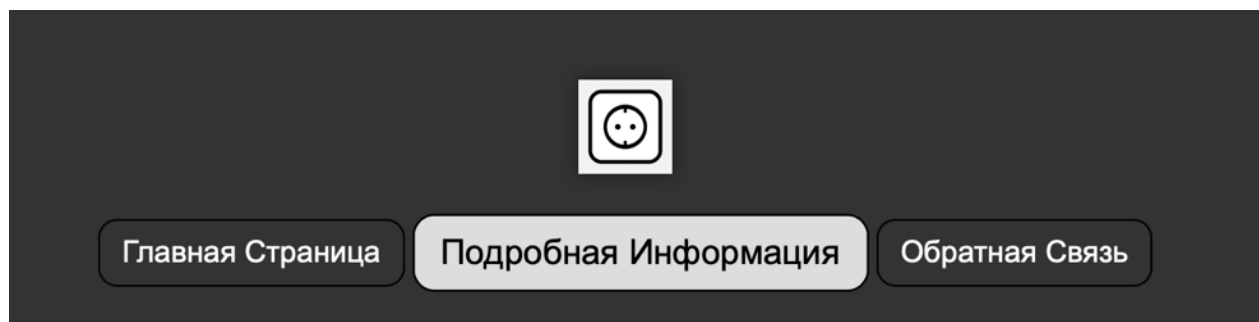


Рисунок 12 - наведение на кнопку Подробная Информация – Увеличение, инверсия цветов

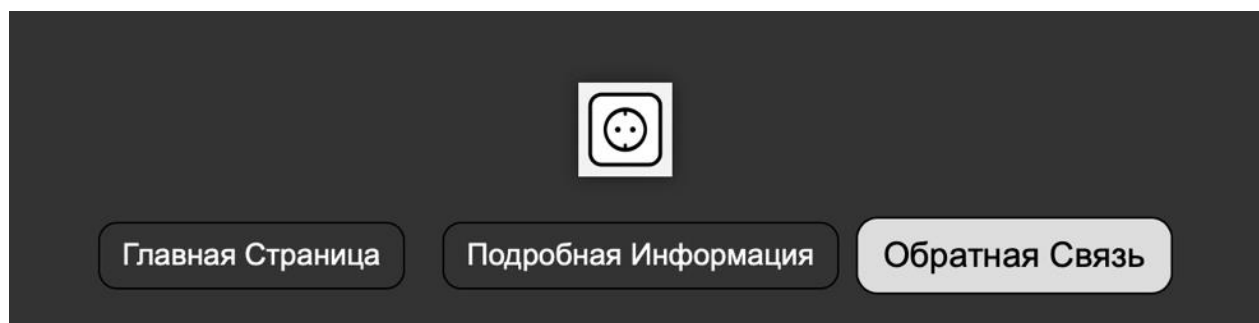


Рисунок 13 - наведение на кнопку Обратная Связь – Увеличение, инверсия цветов

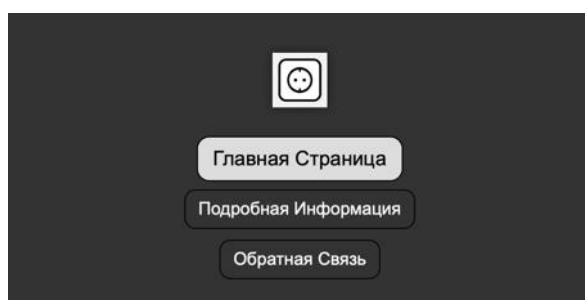


Рисунок 14 – вид шапки сайта на мобильном устройстве, при наведении кнопки обрабатывают так же, как и с большим разрешением

На рисунке 15 продемонстрировано как меняет свой стиль кнопка при наведении. У всех изображений добавлена «тень» для объёмного эффекта. «Тень» так же добавлена на отдельных блоках сайта.



Рисунок 15 – наведение курсора на кнопку, происходит изменение стиля кнопки

На рисунках 16-17 показана анимация перехода между изображениями

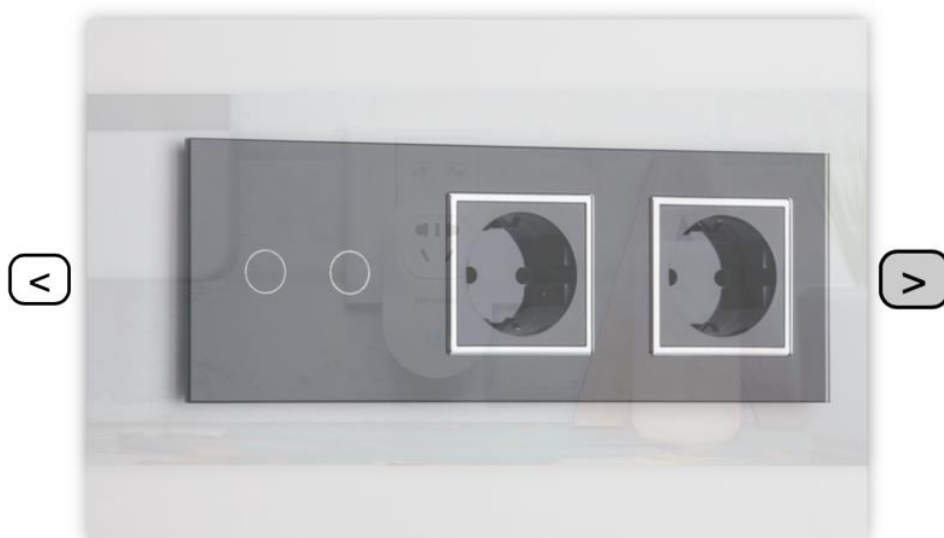


Рисунок 16 – при перелистывании «вперед» текущее изображение затухает и проявляется следующее



Рисунок 17 - при перелистывании «назад» текущее изображение затухает и проявляется следующее

Обратная Связь

Имя:

Email:

Сообщение:

Отправить

Рисунок 18 – изначальный вид формы для обратной связи

На рисунке 19 показана ошибка, которая выводится при отправке пустой формы. Выводится только уведомление под формой, а не модальное окно. Это сделано специально - для удобства, чтобы пользователь мог сразу внести изменения в форму.

The screenshot shows a feedback form titled "Обратная Связь". It contains three input fields: "Имя:" (Name), "Email:", and "Сообщение:" (Message). The "Имя:" and "Email:" fields are empty. The "Сообщение:" field is a large text area, also empty. Below the fields is a blue button labeled "Отправить" (Send). Below the button, a red error message is displayed: "Пожалуйста, заполните все поля." (Please fill out all fields.).

Рисунок 19 – демонстрация отображения ошибки

На рисунках 20-21 показано как после успешного заполнения формы появляется окно об отправке сообщения и очищается форма.

The screenshot shows the same feedback form titled "Обратная Связь", but now it is filled out. The "Имя:" field contains "Андрей", the "Email:" field contains "123@gmail.com", and the "Сообщение:" text area contains "Тест". The blue "Отправить" (Send) button is still present below the fields.

Рисунок 20 – пример верного заполнения формы

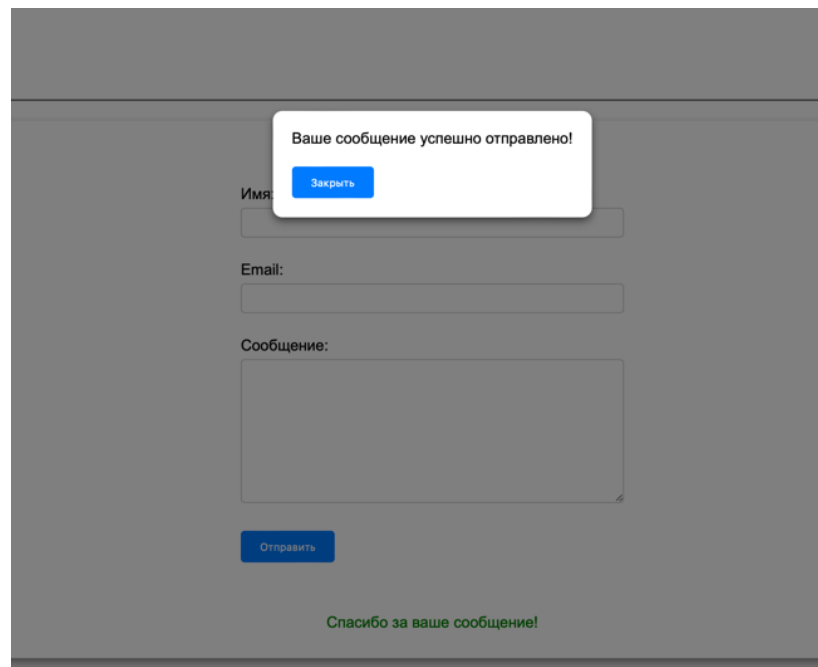


Рисунок 21 – уведомлении об успешной отправке сообщения

На рисунке 22 показано куда именно переведет кнопка «Главная страница» с шапки сайта. Так же откроется эта часть страницы, если нажать на логотип сайта.

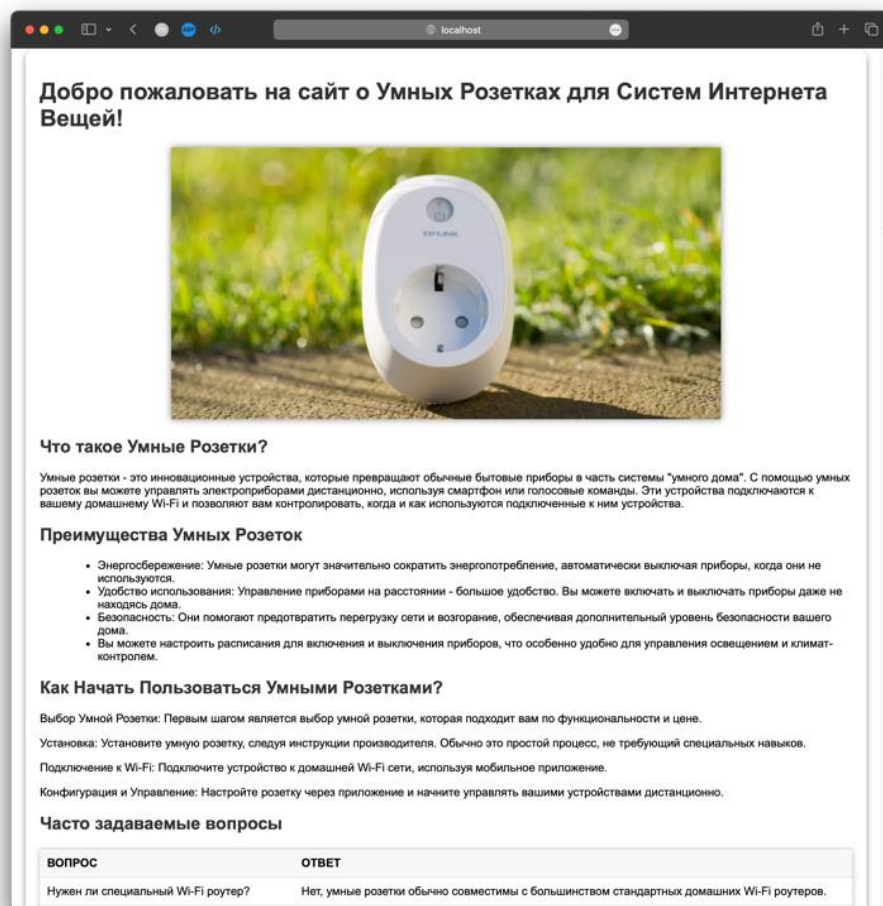


Рисунок 22 – переадресация на первый блок сайта

На рисунке 23 показано куда именно переведет кнопка «Подробная информация» с шапки сайта.

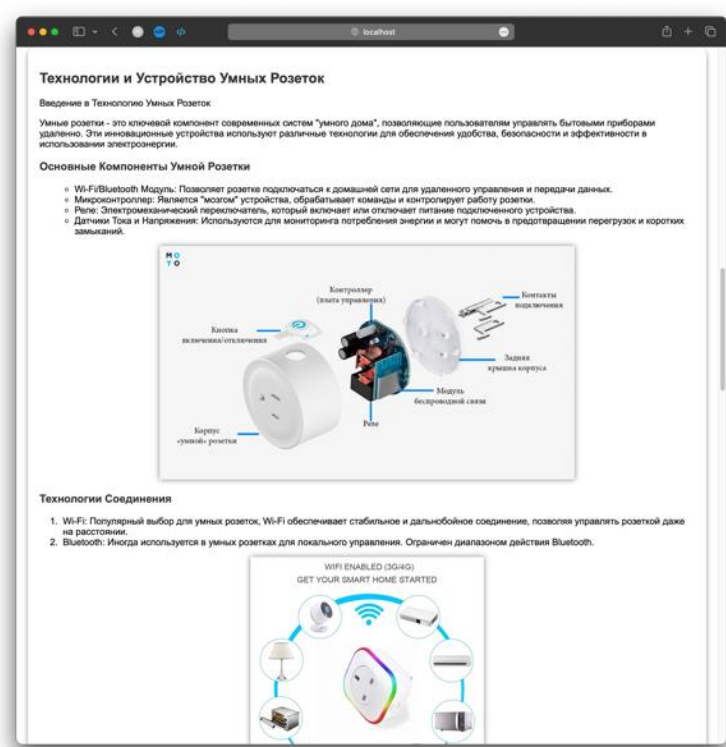


Рисунок 23 – переадресация на блок сайта о технологиях умных розеток

На рисунке 24 показано куда именно переведет кнопка «Обратная связь» с шапки сайта.

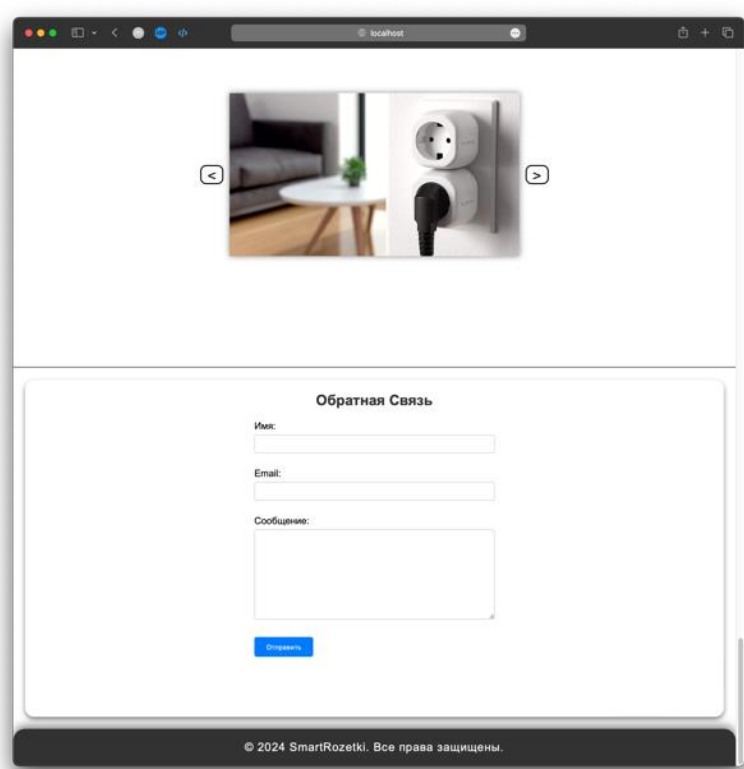


Рисунок 24 - переадресация на блок сайта с обратной связью

Заключение

В заключении данного проекта по созданию одностраничного приложения (SPA) с использованием Vue.js можно отметить, что все поставленные задачи были выполнены успешно. Проект включал в себя разработку веб-интерфейса для сайта, посвященного умным розеткам, с использованием современных технологий веб-разработки и реактивных возможностей Vue.js.

Сайт был разработан с учетом принципов дизайна и пользовательского опыта, обеспечивая удобную и понятную навигацию, а также интерактивность элементов интерфейса. Особое внимание было уделено реактивности и динамичности взаимодействия пользователя с сайтом, что достигнуто благодаря использованию Vue.js.

Компоненты сайта, такие как навигационное меню, форма обратной связи, слайдер изображений и информационные блоки, были тщательно спланированы и реализованы, демонстрируя гибкость и мощь Vue.js в создании интерактивных веб-приложений.

С точки зрения готовности к публикации в Интернете, сайт полностью функционален и готов к развертыванию. Он адаптирован к различным экранам и устройствам, что делает его доступным для широкой аудитории пользователей.

Практическая значимость данной работы заключается в получении и закреплении навыков в области современной веб-разработки, а именно работы с Vue.js, HTML5, CSS3 и JavaScript. Эти навыки являются важными для профессионального развития в области веб-разработки и могут быть применены в будущих проектах.

Таким образом, данный проект не только успешно решает поставленные задачи, но и способствует развитию компетенций в области веб-разработки, что имеет большое значение для дальнейшей профессиональной деятельности.

Приложение:

Листинг файлов:

app.vue

```
<template>
  <div id="app">
    <AppNavbar/>
    <HomePage/>
    <SmartPlugInfo/>
    <ImageSlider/>
    <FeedbackForm/>
    <AppFooter/>
  </div>
</template>

<script>
import AppNavbar from './components/AppNavbar.vue'
import HomePage from './components/HomePage.vue'
import SmartPlugInfo from './components/SmartPlugInfo.vue'
import ImageSlider from './components/ImageSlider.vue'
import FeedbackForm from './components/FeedbackForm.vue'
import AppFooter from './components/AppFooter.vue'

export default {
  name: 'App',
  components: {
    AppNavbar,
    HomePage,
    SmartPlugInfo,
    ImageSlider,
    FeedbackForm,
    AppFooter
  }
}
</script>

<style>
/* Импорт глобальных стилей */
@import './assets/main.css';
</style>
```

appnavbar.vue

```
<template>
  <nav class="app-navbar">
    <div class="navbar-content">
```

```
<!-- Логотип -->
<div class="navbar-logo">
  <a href="#home-page" class="logo-link"> <!-- Добавляем класс для ссылки -->
     <!-- Добавляем класс
для изображения -->
  </a>
</div>

<!-- Меню -->
<ul>
  <li><a href="#home-page">Главная Страница</a></li>
  <li><a href="#smart-plug-info">Подробная Информация</a></li>
  <li><a href="#feedback">Обратная Связь</a></li>
</ul>
</div>
</nav>
</template>
```

```
<script>
export default {
  name: 'AppNavbar'
}
</script>
```

```
<style scoped>
body, html {
  margin: 0;
  padding: 0;
  width: 100%;
  height: 100%;
}
```

```
.app-navbar {
  background-color: #333;
  z-index: 1000;
  width: 100%;
  position: relative; /* Фиксированное позиционирование */
  top: 0; /* Устранение отступа от верха страницы */
  left: 0;
  z-index: 1000; /* Убедиться, что шапка всегда наверху */
  box-shadow: 0 5px 15px rgba(0, 0, 0, 0.5); /* Тень для добавления глубины */
  padding: 10px 0; /* Вертикальные отступы */
  border-bottom-left-radius: 20px; /* Закругление нижнего левого угла */
  border-bottom-right-radius: 20px; /* Закругление нижнего правого угла */
}
```

```
.navbar-content {
  display: flex;
  flex-direction: column;
  align-items: center;
  padding: 20px 0; /* Отступ сверху и снизу внутри шапки */
}
```

```

}

.navbar-logo img {
  height: 50px; /* Высота логотипа */
  margin-bottom: 20px; /* Отступ от логотипа до меню */
}

.app-navbar ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
  display: flex;
  justify-content: center;
}

.app-navbar li {
  margin: 0 10px;
}

.app-navbar li a {
  display: block;
  color: white;
  text-align: center;
  padding: 8px 12px;
  text-decoration: none;
  transition: background-color 0.3s, color 0.3s, transform 0.3s; /* Плавный переход для цвета,
фона и трансформации */
  border-radius: 10px; /* Закругление углов */
  border: 1px solid black; /* Тонкая черная обводка */
}

.app-navbar li a:hover {
  background-color: #ddd;
  color: black;
  transform: scale(1.15); /* Увеличение при наведении */
}

@media (max-width: 800px) {
  .app-navbar ul {
    flex-direction: column; /* Ставим элементы навигации вертикально */
    align-items: center;
  }
  .app-navbar {
    height: auto; /* или задайте фиксированную высоту, если она известна */
  }
  .app-navbar li {
    margin: 5px 0; /* Увеличиваем вертикальные отступы между элементами навигации */
  }

  .content-section {
    padding: 15px; /* Уменьшаем отступы в основных секциях */
    padding-top: 120px; /* Увеличенный отступ, когда меню вертикальное */
  }

```

```

    }
  }
  /* Стили для анимации логотипа */
  .logo-link {
    display: inline-block; /* Делаем ссылку блочно-строчным элементом */
    transition: transform 0.3s ease; /* Плавный переход для трансформации */
  }

  .logo-link:hover {
    transform: scale(1.15); /* Увеличение при наведении */
  }

  .logo-image {
    height: 50px; /* Высота логотипа */
    /* Другие стили для изображения */
  }

</style>

```

homepage.vue

```

<template>
  <div id="home-page" class="content-section">
    <!-- Ваш новый текст -->
    <h1>Добро пожаловать на сайт о Умных Розетках для Систем Интернета Вещей!</h1>
    <div class="home-page-image-container">
      
    </div>
    <h2>Что такое Умные Розетки?</h2>
    <p>Умные розетки - это инновационные устройства, которые превращают обычные бытовые приборы в часть системы "умного дома". С помощью умных розеток вы можете управлять электроприборами дистанционно, используя смартфон или голосовые команды. Эти устройства подключаются к вашему домашнему Wi-Fi и позволяют вам контролировать, когда и как используются подключенные к ним устройства.</p>

    <!-- Преимущества умных розеток в виде списка -->
    <h2>Преимущества Умных Розеток</h2>
    <ul>
      <li>Энергосбережение: Умные розетки могут значительно сократить энергопотребление, автоматически выключая приборы, когда они не используются.</li>
      <li>Удобство использования: Управление приборами на расстоянии - большое удобство. Вы можете включать и выключать приборы даже не находясь дома.</li>
      <li>Безопасность: Они помогают предотвратить перегрузку сети и возгорание, обеспечивая дополнительный уровень безопасности вашего дома.</li>
      <li>Вы можете настроить расписания для включения и выключения приборов, что особенно удобно для управления освещением и климат-контролем.</li>
    </ul>
    <h2>Как Начать Пользоваться Умными Розетками?</h2>
    <p>Выбор Умной Розетки: Первым шагом является выбор умной розетки, которая подходит вам по функциональности и цене.</p>

```

<p>Установка: Установите умную розетку, следуя инструкции производителя. Обычно это простой процесс, не требующий специальных навыков.</p>

<p>Подключение к Wi-Fi: Подключите устройство к домашней Wi-Fi сети, используя мобильное приложение.</p>

<p>Конфигурация и Управление: Настройте розетку через приложение и начните управлять вашими устройствами дистанционно.</p>

<!-- Таблица с часто задаваемыми вопросами -->

<h2>Часто задаваемые вопросы</h2>

<table>

<tr>

<th>Вопрос</th>

<th>Ответ</th>

</tr>

<tr>

<td>Нужен ли специальный Wi-Fi роутер?</td>

<td>Нет, умные розетки обычно совместимы с большинством стандартных домашних Wi-Fi роутеров.</td>

</tr>

<tr>

<td>Можно ли управлять умной розеткой, когда я не дома?</td>

<td>Да, вы можете управлять розеткой из любой точки мира, где есть доступ в интернет.</td>

</tr>

<tr>

<td>Безопасны ли умные розетки для моего дома?</td>

<td>Да, они обеспечивают дополнительный уровень безопасности за счет возможности удаленного управления и мониторинга.</td>

</tr>

</table>

<h2>Полезные Советы</h2>

Используйте умные розетки для управления освещением в вашем доме. Это не только удобно, но и поможет сэкономить на электроэнергии.

Настройте автоматическое отключение приборов во время вашего отсутствия дома, чтобы снизить потребление энергии.

Используйте функции голосового управления для еще большего удобства, особенно когда ваши руки заняты.

<div class="home-page-image-container">

</div>

</div>

<hr>

</template>

<script>

export default {

name: 'HomePage'

}

</script>

<style scoped>

/* Устанавливаем сероватый фон для всей страницы */

```
body {  
  background-color: #f5f5f5;  
}
```

/* Стили для раздела с текстом */

```
.content-section {  
  background-color: white; /* Белый фон для раздела */  
  padding: 20px;  
  border-radius: 15px; /* Закругленные углы */  
  box-shadow: 0 4px 8px rgba(0, 0, 0, 0.5); /* Небольшая тень для эффекта "выпуклости" */  
  margin: 20px; /* Добавляем отступ, чтобы отделить раздел от краев экрана */  
}
```

/* Стили для заголовков и списка */

```
.content-section h2 {  
  margin-top: 20px;  
}
```

```
.content-section ul {  
  list-style-type: disc;  
  margin-left: 40px;  
}
```

```
.content-section {  
  padding: 20px;  
}
```

```
.content-section h2 {  
  margin-top: 20px;  
}
```

```
.content-section ul {  
  list-style-type: disc;  
  margin-left: 40px;  
}
```

/* Стили для таблицы */

```
table {  
  width: 100%;  
  border-collapse: separate;  
  border-spacing: 0;  
  margin-top: 20px;  
  box-shadow: 0 4px 8px rgba(0, 0, 0, 0.3); /* Добавляем тень для объемного эффекта */  
}
```

```

th, td {
  padding: 10px;
  text-align: left;
  border-bottom: 1px solid #ddd; /* Мало заметная граница */
}

th {
  background-color: #f8f8f8; /* Нежный цвет фона для заголовков */
  border-top: 2px solid #ddd; /* Верхняя граница для заголовков */
  text-transform: uppercase; /* Заглавные буквы для текста заголовков */
}

td {
  background-color: #fff; /* Белый фон для ячеек */
}

/* Стил для строк таблицы при наведении */
tr:hover {
  background-color: #f1f1f1; /* Светло-серый фон при наведении */
}

.home-page-image-container {
  text-align: center; /* Центрирование содержимого */
}

.home-page-image {
  max-width: 100%;
  height: auto;
  max-height: 400px; /* Максимальная высота изображения */
}

@media (max-width: 800px) {
  .home-page-image {
    max-height: 300px; /* Адаптивная высота изображения для средних экранов */
  }
}

@media (max-width: 550px) {
  .home-page-image {
    max-height: 200px; /* Адаптивная высота изображения для малых экранов */
  }
}

</style>

```

smartpluginfo.vue

```

<template>
  <div id="smart-plug-info" class="content-section">

```

<h2>Технологии и Устройство Умных Розеток</h2>

<p>Введение в Технологию Умных Розеток</p>

<p>Умные розетки - это ключевой компонент современных систем "умного дома", позволяющие пользователям управлять бытовыми приборами удаленно. Эти инновационные устройства используют различные технологии для обеспечения удобства, безопасности и эффективности в использовании электроэнергии.</p>

<h3>Основные Компоненты Умной Розетки</h3>

Wi-Fi/Bluetooth Модуль: Позволяет розетке подключаться к домашней сети для удаленного управления и передачи данных.

Микроконтроллер: Является "мозгом" устройства, обрабатывает команды и контролирует работу розетки.

Реле: Электромеханический переключатель, который включает или отключает питание подключенного устройства.

Датчики Тока и Напряжения: Используются для мониторинга потребления энергии и могут помочь в предотвращении перегрузок и коротких замыканий.

<div class="smart-plug-image-container">

</div>

<h3>Технологии Соединения</h3>

Wi-Fi: Популярный выбор для умных розеток, Wi-Fi обеспечивает стабильное и дальнбойное соединение, позволяя управлять розеткой даже на расстоянии.

Bluetooth: Иногда используется в умных розетках для локального управления. Ограничен диапазоном действия Bluetooth.

<div class="smart-plug-image-container">

</div>

<h3>Интеграция и Совместимость</h3>

Совместимость с Голосовыми Ассистентами: Большинство умных розеток можно интегрировать с Яндекс Алисой, Amazon Alexa, Google Assistant и Apple HomeKit для управления голосом.

Поддержка Множества Платформ: Умные розетки часто поддерживают различные платформы управления, включая специализированные приложения, IFTTT (If This Then That) и другие сервисы автоматизации.

<div class="smart-plug-image-container">

</div>

<h3>Функции Безопасности и Мониторинга</h3>

<dl>

<dt>Защита от Перегрузки:</dt>

<dd>Современные умные розетки часто оснащены функциями защиты от перегрузки, автоматически отключаясь при обнаружении потенциальной опасности.</dd>

<dt>Мониторинг Энергопотребления:</dt>

<dd>Датчики тока и напряжения позволяют отслеживать энергопотребление в реальном времени, помогая оптимизировать использование электроэнергии.</dd>

</dl>

<h3>Принципы Работы Умных Розеток</h3>

<dl>

<dt>Получение Команд:</dt>

<dd>Умные розетки получают команды через Wi-Fi или Bluetooth от мобильных приложений или голосовых ассистентов.</dd>

<dt>Обработка Команд:</dt>

<dd>Микроконтроллер интерпретирует полученные команды и активирует реле для включения или выключения питания.</dd>

</dl>

<h3>Продвинутые Функции</h3>

Расписания и Таймеры: Пользователи могут настроить автоматическое включение/выключение устройств в определенное время или в ответ на определенные события.

Сценарии Автоматизации: Интеграция с системами умного дома для создания сложных сценариев, таких как включение света при открытии двери.

<h3>Примеры Сценариев Использования Умных Розеток</h3>

<table>

<tr>

<th>Сценарий</th>

<th>Описание</th>

</tr>

<tr>

<td>Управление Освещением</td>

<td>Автоматизируйте освещение вашего дома, настраивая умные розетки на включение света в определенное время, например, когда начинает темнеть.</td>

</tr>

<tr>

<td>Дистанционное Управление Бытовыми Приборами</td>

<td>Включайте или выключайте приборы (например, кофеварку или утюг) дистанционно, чтобы они были готовы к вашему приходу домой или для предотвращения опасности.</td>

</tr>

<tr>

<td>Энергосбережение</td>

<td>Используйте мониторинг энергопотребления для оптимизации использования электроэнергии, автоматически выключая приборы, когда они не используются.</td>

</tr>

<tr>

<td>Безопасность в Отпуске</td>

<td>Создайте видимость присутствия в доме, используя умные розетки для включения света и других приборов в разное время, когда вы в отпуске.</td>

</tr>

<tr>

<td>Голосовое Управление</td>

<td>Интегрируйте умные розетки с голосовым ассистентом для удобного управления приборами, не вставая с места.</td>

</tr>

</table>

<h3>Заключение</h3>

<p>Технологии умных розеток - это симбиоз инноваций в области беспроводной связи, электроники и программного обеспечения, привносящие удобство и эффективность в повседневную жизнь и развитие систем умного дома.</p>

```
<div class="smart-plug-image-container">
  
</div>
</div>
</template>
```

```
<script>
export default {
  name: 'SmartPlugInfo'
}
</script>
```

```
<style scoped>
/* Стили для раздела с информацией об умных розетках */
.content-section {
  background-color: white; /* Белый фон для раздела */
  padding: 20px;
  border-radius: 15px; /* Закругленные углы */
  box-shadow: 0 4px 8px rgba(0, 0, 0, 0.5); /* Небольшая тень для эффекта "выпуклости" */
  margin: 20px; /* Отступ, чтобы отделить раздел от краев экрана */
}
```

```
.content-section ul {
  list-style-type: circle; /* Стилль списка */
  margin-left: 20px;
}
.content-section {
  padding: 20px;
}
```

```
.content-section ul {
  list-style-type: circle;
  margin-left: 20px;
}
```

```
/* Стили для таблицы */
table {
  width: 100%;
  border-collapse: separate;
  border-spacing: 0;
  margin-top: 20px;
  box-shadow: 0 4px 8px rgba(0, 0, 0, 0.3); /* Добавляем тень для объемного эффекта */
}
```

```
th, td {
  padding: 10px;
  text-align: left;
  border-bottom: 1px solid #ddd; /* Малоаметная граница */
}
```

```

}

th {
  background-color: #f8f8f8; /* Нежный цвет фона для заголовков */
  border-top: 2px solid #ddd; /* Верхняя граница для заголовков */
  text-transform: uppercase; /* Заглавные буквы для текста заголовков */
}

td {
  background-color: #fff; /* Белый фон для ячеек */
}

/* Стилль для строк таблицы при наведении */
tr:hover {
  background-color: #f1f1f1; /* Светло-серый фон при наведении */
}

.smart-plug-image-container {
  text-align: center; /* Центрирование содержимого */
}

.smart-plug-image {
  max-width: 100%;
  height: auto;
  max-height: 400px; /* Максимальная высота изображения */
}

@media (max-width: 800px) {
  .smart-plug-image {
    max-height: 300px; /* Адаптивная высота изображения для средних экранов */
  }
}

@media (max-width: 550px) {
  .smart-plug-image {
    max-height: 200px; /* Адаптивная высота изображения для малых экранов */
  }
}

</style>

```

imageslide.vue

```

<template>
  <div class="image-slider-container">
    <div class="slider-header">
      <h2>Вдохновляющие идеи для вашего дома:</h2>
      <h2>Как могут выглядеть Умные розетки в интерьере</h2>
    </div>

```

```

<div class="image-slider">
  <button @click="prevSlide" class="slider-btn prev-btn">&#60;</button>
  <div class="slides">
    <transition-group name="fade" tag="div">
      
    </transition-group>
  </div>
  <button @click="nextSlide" class="slider-btn next-btn">&#62;</button>
</div>
</template>

```

```

<script>
export default {
  name: 'ImageSlider',
  data() {
    return {
      currentSlide: 0,
      images: [
        require('@/assets/photo6.jpg'),
        require('@/assets/photo7.jpg'),
        require('@/assets/photo8.jpg'),
        require('@/assets/photo9.jpg'),
        require('@/assets/photo10.jpg'),
        require('@/assets/photo11.jpg'),
        require('@/assets/photo12.jpg'),
        require('@/assets/photo13.jpg'),
        require('@/assets/photo14.jpg'),
        require('@/assets/photo15.jpg')
      ]
    }
  },
  methods: {
    nextSlide() {
      this.currentSlide = (this.currentSlide + 1) % this.images.length;
    },
    prevSlide() {
      this.currentSlide = this.currentSlide - 1 < 0 ? this.images.length - 1 : this.currentSlide - 1;
    }
  }
};
</script>

```

```

<style scoped>

```

```

.image-slider-container {

```

```

text-align: center; /* Центрирование текста */
margin-bottom: 20px; /* Отступ перед слайдером */
}

.slider-header h2 {
font-size: 24px; /* Размер шрифта для заголовка */
color: #333; /* Цвет текста */
margin: 20px 0; /* Вертикальные отступы */
}

.image-slider {
width: 600px;
height: 600px;
position: relative;
margin: auto;
overflow: hidden;
display: flex;
align-items: center;
justify-content: space-between; /* Распределение пространства */
}

@media (max-width: 800px) {
.image-slider {
width: 80%;
min-height: 300px;
}
}

@media (max-width: 550px) {
.image-slider {
width: 70%;
min-height: 200px;
}
}

.slides {
width: 100%;
height: 100%;
display: flex; /* Использование Flexbox */
justify-content: center;
align-items: center;
position: relative;
}

.slides img {
max-width: 100%; /* Максимальная ширина изображения */
max-height: 100%; /* Максимальная высота изображения */
object-fit: contain; /* Это свойство гарантирует, что изображения полностью помещаются в
контейнер без обрезки */
position: absolute;
top: 50%;
left: 50%;
transform: translate(-50%, -50%); /* Центрирование изображения */
}

```



```

    transition: opacity 0.5s ease;
  }
  /* Стили для анимации исчезновения и появления */
  .fade-enter-active, .fade-leave-active {
    transition: opacity 0.5s ease;
  }

  .fade-enter, .fade-leave-to {
    opacity: 0;
  }

  /* Добавление промежуточного состояния для плавного перехода */
  .fade-enter-to, .fade-leave {
    opacity: 1;
  }

  .slider-btn {
    background-color: #fff;
    border: 2px solid black;
    color: black;
    padding: 0px 10px;
    cursor: pointer;
    transition: background-color 0.3s, color 0.3s, transform 0.3s; /* Плавный переход для цвета,
фона и трансформации */
    z-index: 10;
    font-size: 24px;
    margin: 0 10px; /* Добавление отступов для кнопок */
    border-radius: 10px; /* Закругление краев */
  }

  .slider-btn:hover {
    background-color: #ccc; /* Более темный цвет при наведении */
    transform: scale(1.15); /* Увеличение при наведении */
  }
</style>

```

feedbackform.vue

```

<template>
  <hr>
  <div id="feedback" class="content-section">
    <h2>Обратная Связь</h2>
    <form @submit.prevent="submitForm">
      <div class="form-group">
        <label for="name">Имя:</label>
        <input type="text" id="name" v-model="formData.name">
      </div>

```

```

<div class="form-group">
  <label for="email">Email:</label>
  <input type="email" id="email" v-model="formData.email">
</div>
<div class="form-group">
  <label for="message">Сообщение:</label>
  <textarea id="message" v-model="formData.message"></textarea>
</div>
<div class="message-container">
  <p v-if="errorMessage" class="error-message">{{ errorMessage }}</p>
  <p class="submitted-message" :class="{ active: submitted }">Спасибо за ваше
сообщение!</p>
</div>
<button type="submit">Отправить</button>
</form>
<div v-if="showModal" class="modal-overlay" @click="closeModal">
  <div class="modal-content" @click.stop>
    <p>Ваше сообщение успешно отправлено!</p>
    <button @click="closeModal">Закрыть</button>
  </div>
</div>
</div>
</template>

<script>
export default {
  name: 'FeedbackForm',
  data() {
    return {
      formData: {
        name: "",
        email: "",
        message: ""
      },
      submitted: false,
      errorMessage: "",
      showModal: false
    };
  },
  methods: {
    submitForm() {
      // Валидация формы
      if (!this.formData.name || !this.formData.email || !this.formData.message) {
        this.errorMessage = 'Пожалуйста, заполните все поля.';
        return;
      }

      if (!/^[S+@S+\.S+/.test(this.formData.email)) {
        this.errorMessage = 'Пожалуйста, введите корректный адрес электронной почты.';
        return;
      }
    }
  }
}

```

```

        console.log('Отправлено:', this.formData);
        this.submitted = true;
        this.errorMessage = "";
        this.clearForm();

        this.showModal = true; // Открываем модальное окно

    },

    closeModal() {
        this.showModal = false;
    },

    clearForm() {
        this.formData.name = "";
        this.formData.email = "";
        this.formData.message = "";
    }
}
</script>

<style scoped>

/* Стили для формы */
.content-section {
    background-color: white; /* Белый фон для раздела */
    padding: 0px;
    border-radius: 15px; /* Закругленные углы */
    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.5); /* Небольшая тень для эффекта "выпуклости" */
    display: flex;
    flex-direction: column;
    align-items: center;
    margin-bottom: 60px; /* Дополнительное место под формой */
    margin: 20px; /* Отступ, чтобы отделить раздел от краев экрана */
    position: relative; /* Для абсолютного позиционирования внутри */
    padding-bottom: 100px; /* Дополнительный отступ снизу для сообщений */
}
.content-section ul {
    list-style-type: circle; /* Стилль списка */
    margin-left: 20px;
}
.form-group {
    margin-bottom: 15px;
    width: 100%;
    max-width: 600px; /* Ограничение ширины */
}

label {

```

```
display: block;
margin-bottom: 5px;
text-align: left;
}
```

```
input, textarea {
width: 100%;
border: 1px solid #ccc;
border-radius: 4px;
box-sizing: border-box;
}
```

```
input {
padding: 8px;
margin-bottom: 10px;
}
```

```
textarea {
padding: 8px;
margin-bottom: 10px;
height: 150px; /* Увеличиваем высоту поля ввода сообщения */
width: 400px;
}
```

```
button {
padding: 10px 20px;
border: none;
border-radius: 4px;
background-color: #007bff;
color: white;
cursor: pointer;
transition: background-color 0.3s;
align-self: center; /* Выравнивание кнопки по центру */
}
```

```
button:hover {
background-color: #0056b3;
}
```

```
/* Стили для блоков сообщений */
.message-container {
position: absolute;
bottom: 10px; /* Положение в нижней части формы */
left: 20px;
right: 20px;
}
```

```
.error-message, .submitted-message {
text-align: center;
height: 20px;
}
```

```

.error-message {
  color: red;
}

.submitted-message {
  color: green;
  visibility: hidden;
}

.submitted-message.active, .error-message.active {
  visibility: visible;
}

```

```

/* Стили для модального окна */

```

```

.modal-overlay {
  position: fixed;
  top: 0;
  left: 0;
  width: 100%;
  height: 100%;
  background-color: rgba(0, 0, 0, 0.5);
  display: flex;
  justify-content: center;
  align-items: center;
}

.modal-content {
  background-color: white;
  padding: 20px;
  border-radius: 10px;
  box-shadow: 0 4px 8px rgba(0, 0, 0, 0.5);
}

.modal-content p {
  margin: 0;
  margin-bottom: 20px;
}

```

```

</style>

```

appfooter.vue

```

<template>
  <footer class="app-footer">
    <!-- Содержимое футера -->
    <p>&copy; {{ currentYear }} SmartRozetki. Все права защищены.</p>
  </footer>
</template>

```

```

<script>
export default {
  name: 'AppFooter',
  data() {
    return {
      currentYear: new Date().getFullYear()
    };
  }
}
</script>

<style scoped>
.app-footer {
  background-color: #333; /* Темный фон */
  color: white; /* Белый цвет текста */
  text-align: center; /* Выравнивание текста по центру */
  padding: 20px 0; /* Вертикальный отступ */
  position: relative;
  bottom: 0;
  width: 100%;
  border-top-left-radius: 20px; /* Закругление левого верхнего угла */
  border-top-right-radius: 20px; /* Закругление правого верхнего угла */
  box-shadow: 0 -5px 15px rgba(0, 0, 0, 0.5); /* Тень для добавления глубины */
  font-family: 'Arial', sans-serif; /* Шрифт */
}

/* Стиль для текста в футере */
.app-footer p {
  margin: 0; /* Убрать стандартный отступ у параграфа */
  font-size: 1.2em; /* Увеличить размер шрифта */
  font-weight: 500; /* Полужирное начертание для лучшей видимости */
  letter-spacing: 1px; /* Расстояние между буквами для улучшения читаемости */
}

/* Адаптивные стили */
@media (max-width: 800px) {
  .app-footer {
    font-size: 0.9em; /* Меньший размер шрифта на мобильных устройствах */
    padding: 15px 0; /* Меньший отступ на мобильных устройствах */
  }
}
</style>

```

index.html

```

<!DOCTYPE html>
<html lang="ru">
<head>
  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width,initial-scale=1.0">

```

```

<meta name="description" content="Описание сайта">
<meta name="keywords" content="умный дом, умные розетки, розетки">
<link rel="icon" href="/assets/logo.png">
<title>SmartRozetki</title>
</head>
<body>
  <noscript>
    <strong>Для полной функциональности этого сайта необходимо включить
JavaScript.</strong>
  </noscript>
  <div id="app"></div>
</body>
</html>

```

main.css

```

body {
  font-family: Arial, sans-serif;
  margin: 0;
  padding: 0;
  box-sizing: border-box;
}

h1, h2, h3, h4, h5, h6 {
  color: #333;
}

a {
  color: #007bff;
  text-decoration: none;
}

a:hover {
  text-decoration: underline;
}

/* Адаптивность */
@media (max-width: 1200px) {
  body {
    font-size: 16px; /* Немного уменьшаем размер шрифта */
  }

  .app-navbar li a {
    padding: 10px 14px; /* Уменьшаем отступы в навигационной панели */
  }
}

@media (max-width: 800px) {
  .app-navbar ul {
    flex-direction: column; /* Ставим элементы навигации вертикально */
  }
}

```

```

    align-items: center;
  }
  .app-navbar {
    height: auto; /* или задайте фиксированную высоту, если она известна */
  }
  .app-navbar li {
    margin: 20px 0; /* Уменьшаем отступы между элементами навигации */
  }

  .content-section {
    padding: 15px; /* Уменьшаем отступы в основных секциях */
    padding-top: 120px; /* Увеличенный отступ, когда меню вертикальное */
  }
}

@media (max-width: 550px) {
  h1, h2, h3 {
    font-size: 1.2em; /* Уменьшаем размеры заголовков */
  }
  .app-navbar {
    height: auto; /* или задайте фиксированную высоту, если она известна */
  }
  .app-navbar li a {
    font-size: 0.9em; /* Уменьшаем размер текста в навигации */
  }

  .content-section {
    padding-top: 120px; /* Увеличенный отступ, когда меню вертикальное */
  }
}

img {
  box-shadow: 0px 0px 10px rgba(0, 0, 0, 0.5); /* Создание тени вокруг изображений */
}

```

Main.js

```

import { createApp } from 'vue'
import App from './App.vue'

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

```

babel.config.js

```

module.exports = {
  presets: [
    '@vue/cli-plugin-babel/preset'
  ]
}

```



```
}
```

jsonconfig.json

```
{
  "compilerOptions": {
    "target": "es5",
    "module": "esnext",
    "baseUrl": "./",
    "moduleResolution": "node",
    "paths": {
      "@/*": [
        "src/*"
      ]
    },
    "lib": [
      "esnext",
      "dom",
      "dom.iterable",
      "scripthost"
    ]
  }
}
```

Package.json

```
{
  "name": "my-smart-plug-website",
  "version": "0.1.0",
  "private": true,
  "scripts": {
    "serve": "vue-cli-service serve",
    "build": "vue-cli-service build",
    "lint": "vue-cli-service lint"
  },
  "dependencies": {
    "core-js": "^3.8.3",
    "vue": "^3.2.13"
  },
  "devDependencies": {
    "@babel/core": "^7.12.16",
    "@babel/eslint-parser": "^7.12.16",
    "@vue/cli-plugin-babel": "~5.0.0",
    "@vue/cli-plugin-eslint": "~5.0.0",
    "@vue/cli-service": "~5.0.0",
    "eslint": "^7.32.0",
    "eslint-plugin-vue": "^8.0.3"
  },
  "eslintConfig": {
    "root": true,
    "env": {
      "node": true
    },
    "extends": [
      "plugin:vue/vue3-essential",
      "eslint:recommended"
    ],
    "parserOptions": {
      "parser": "@babel/eslint-parser"
    },
    "rules": {}
  },
  "browserslist": [
```

```
"> 1%",  
"last 2 versions",  
"not dead",  
"not ie 11"  
}]}
```