**Введение:**

Сайт создан для людей, которые планируют арендовать сервер. При помощи данной страницы они смогут получить информацию об аренде серверов и связаться с арендодателем.

1. **Постановка задачи:**

**1.1. Требования к проекту**

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 данной дисциплине и на лабораторном практикуме. Обязательным является использование HTML5, технологии CSS3, а также программирования на JavaScript, Vue,js.

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 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.

**1.2. Состав команды исполнителей**

Павлов Сергей Александрович (программист):

* Разработка и реализация функциональности проекта на основе требований и спецификаций.
* Создание компонентов и модулей на основе выбранного фреймворка.
* Отладка, тестирование и оптимизация кода.
* Управление зависимостями и сборка проекта с использованием инструментов сборки.

Шагдуров Баир Баирович (дизайнер):

* Разработка дизайна интерфейса пользователя, включая внешний вид, компоновку и графические элементы.
* Создание макетов и прототипов страниц, визуализация концепции и навигации.
* Разработка цветовой схемы, типографии и стилей элементов интерфейса.
* Создание графических ресурсов, таких как иллюстрации, иконки и фоны.
* Взаимодействие с программистом для обеспечения соответствия дизайна техническим возможностям и требованиям проекта.

1. **Проектирование сайта:**

**2.1. Выбор программных средств и технологий для решения поставленных задач**

Выбор программных средств и технологий для решения поставленных задач основывается на следующих соображениях:

Vue.js: Был выбран Vue.js для разработки SPA (Single Page Application) веб-сайта, так как Vue.js является одним из популярных и мощных фреймворков JavaScript. Он обеспечивает эффективное управление состоянием приложения, реактивность данных и модульность компонентов, что позволяет легко создавать интерактивные пользовательские интерфейсы.

Vue CLI: Использование Vue CLI облегчает процесс разработки, предоставляя инструменты для быстрой настройки и сборки проекта, а также автоматическую конфигурацию и генерацию компонентов.

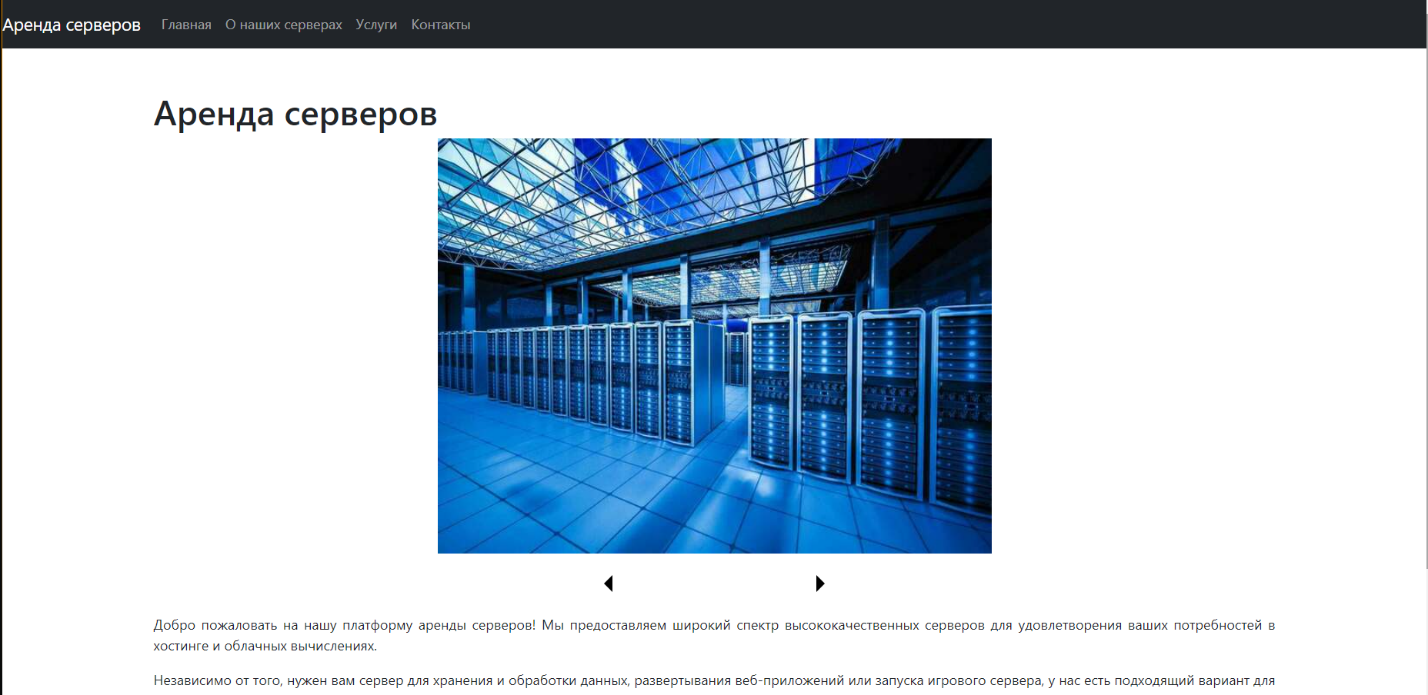
HTML5, CSS3 и Bootstrap: HTML5 и CSS3 являются основными языками разметки и стилей веб-страниц. Они позволяют создавать семантическую разметку и применять стили к элементам страницы. Для упрощения стилизации и создания адаптивного дизайна был выбран Bootstrap, который предоставляет готовые компоненты и сетку.

JavaScript: JavaScript используется для добавления интерактивности и функциональности на стороне клиента. Он позволяет управлять событиями, выполнять валидацию форм, а также реагировать на действия пользователя.

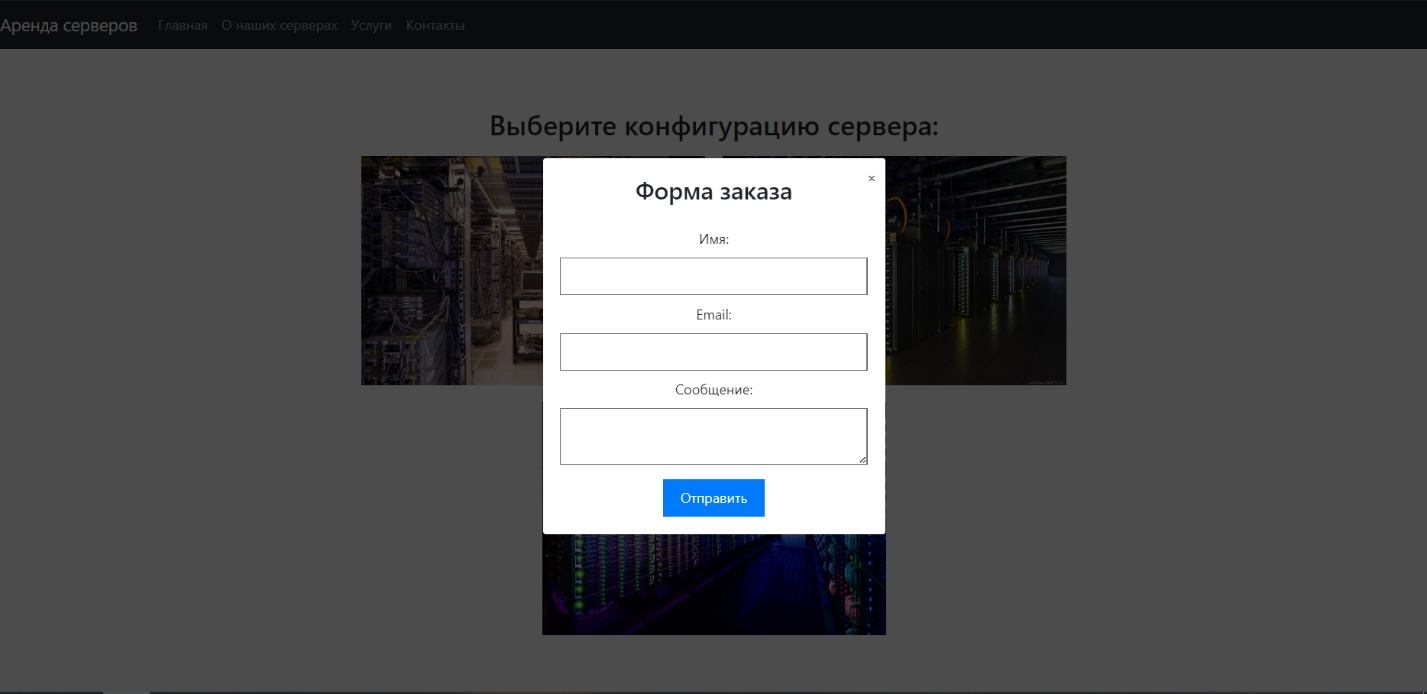
Vue Router: Vue Router используется для реализации навигационного меню с переключением между вкладками без перезагрузки страницы. Он обеспечивает маршрутизацию внутри приложения и управление состоянием при переключении между компонентами.

Выбор этих программных средств и технологий обусловлен их популярностью, эффективностью, гибкостью и способностью ускорить процесс разработки, обеспечивая удобную работу с данными, интерактивностью и адаптивным дизайном.

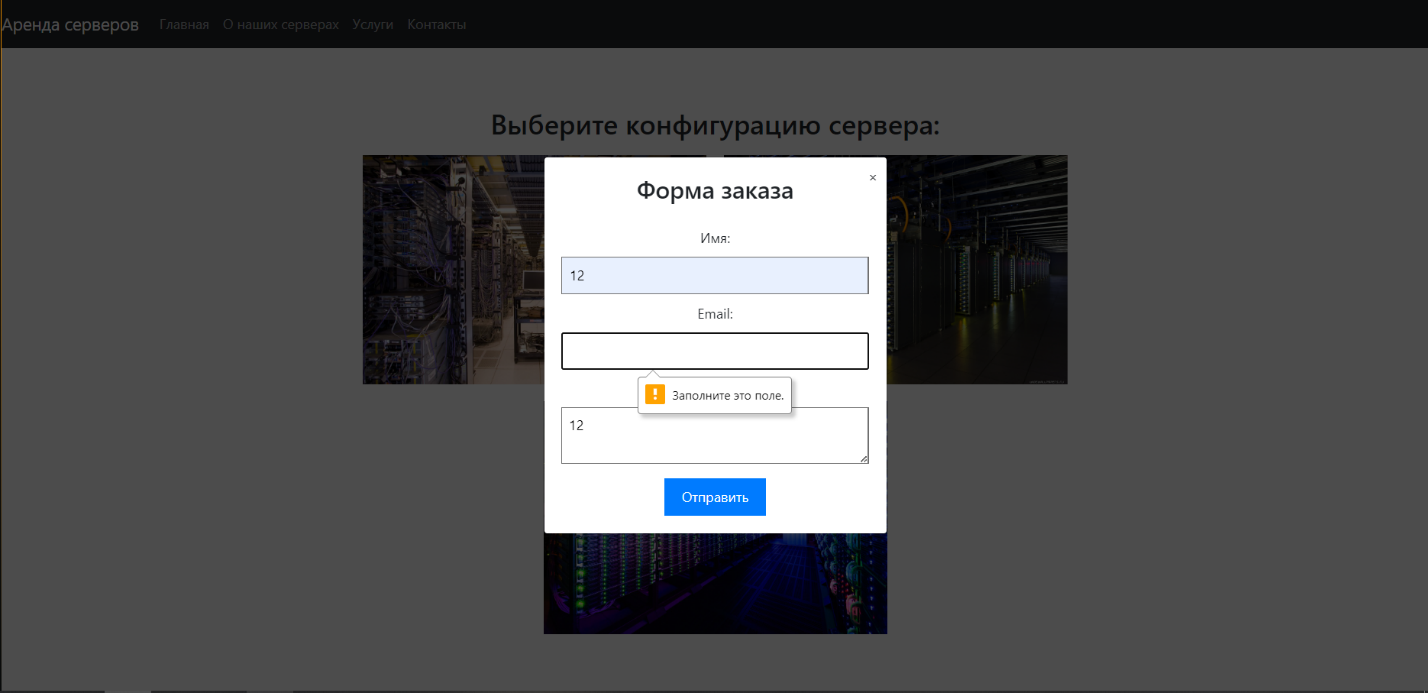
**2.2 Макет сайта**



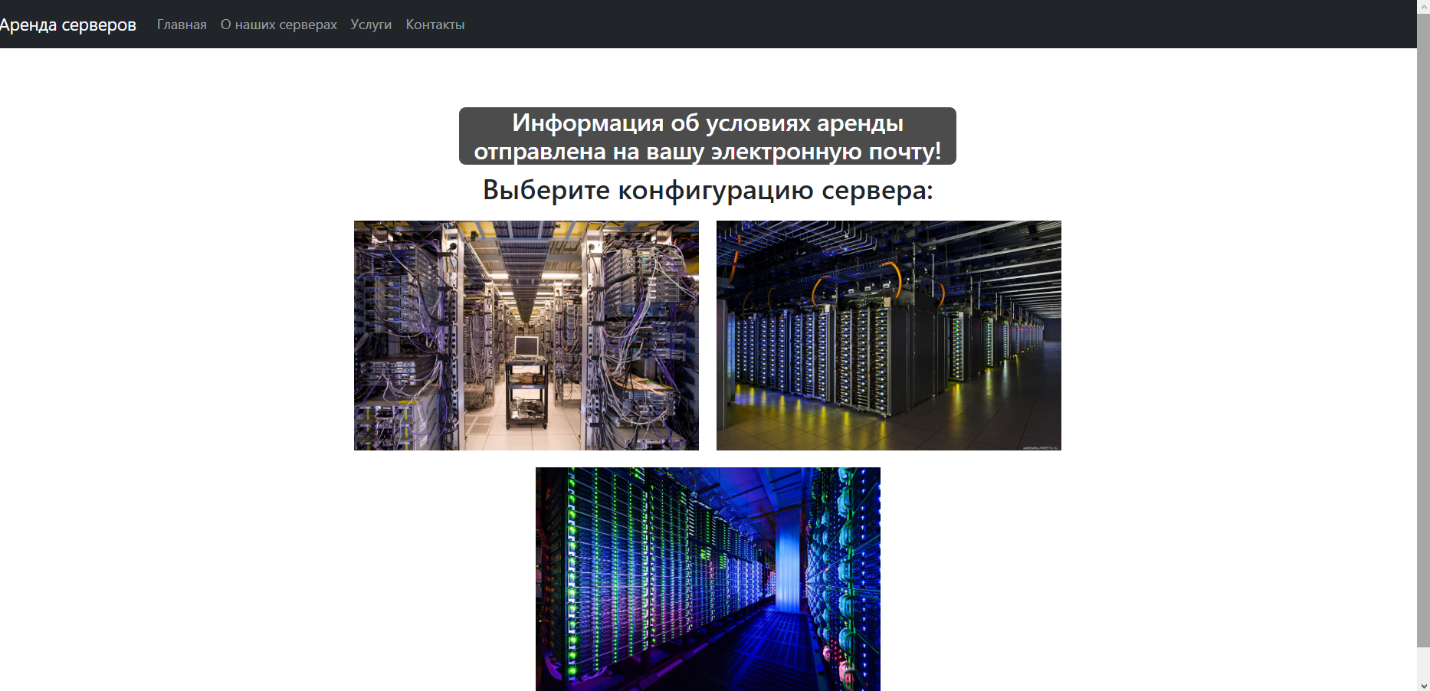
*Рис. 1. Домашняя страница сайта.*



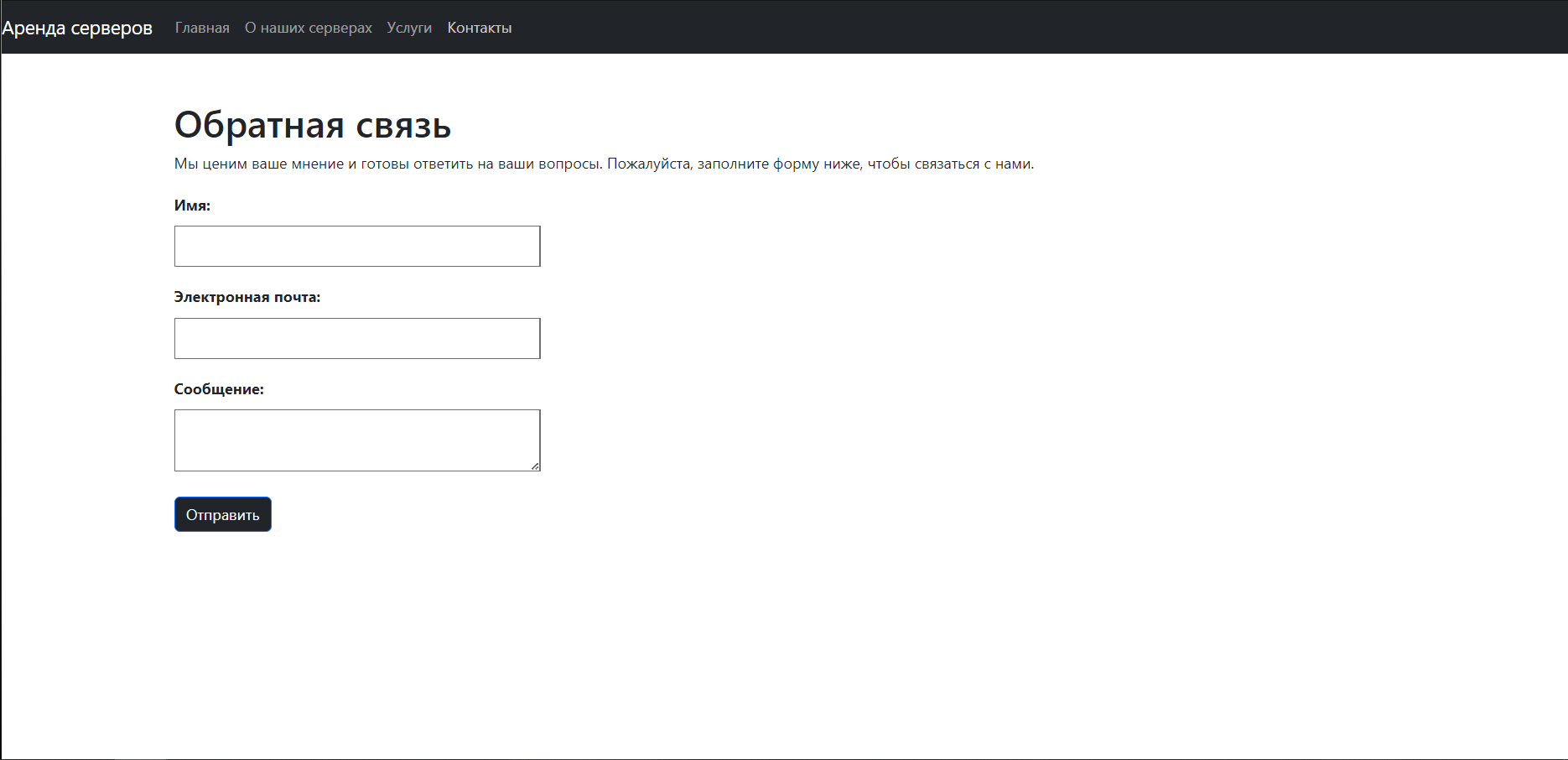
*Рис. 2. Форма заказа аренды серверов.*



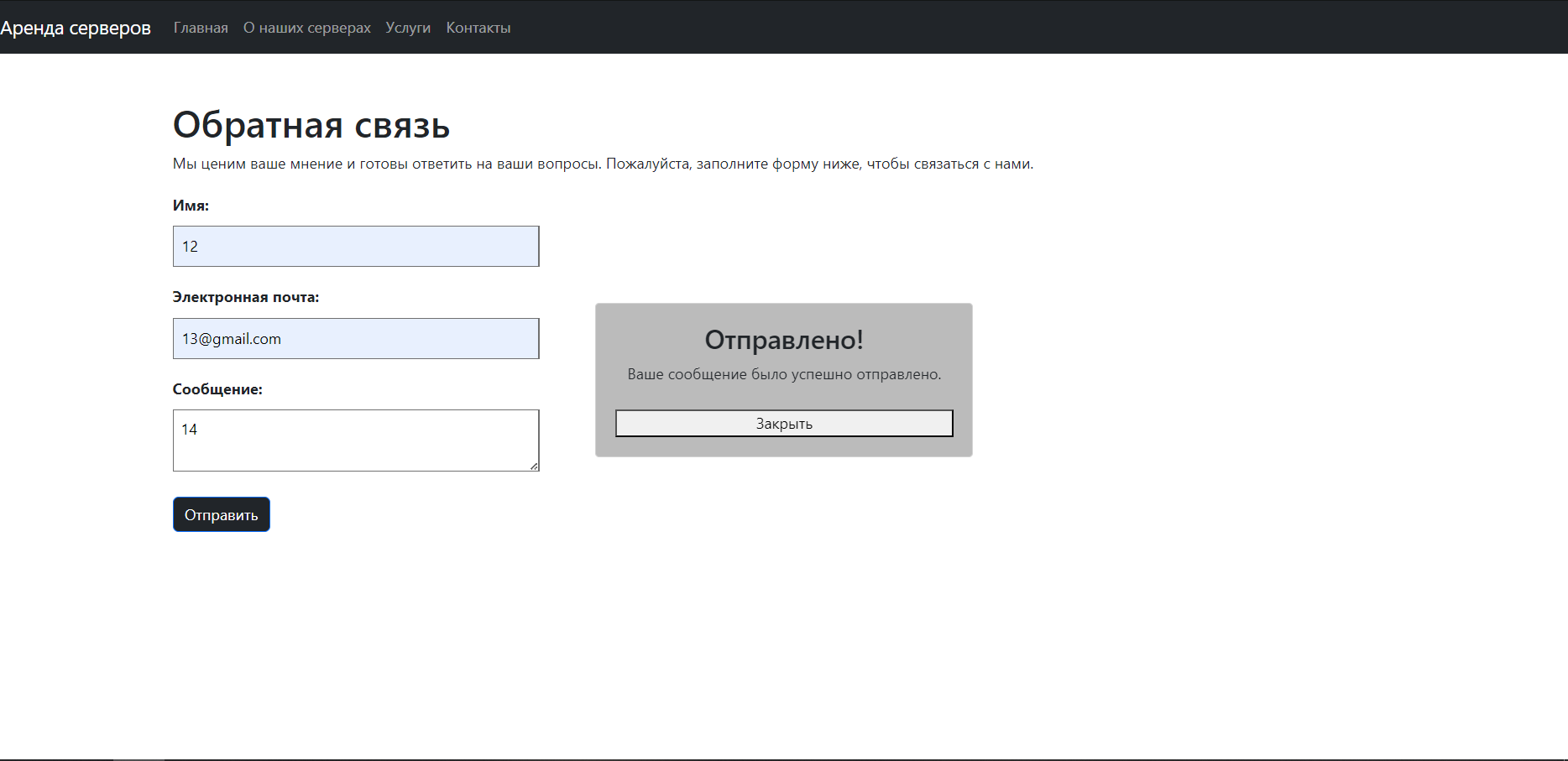
*Рис. 3. Ошибка отправки формы при не заполненном поле.*



*Рис. 4. Страница, показывающая уведомление об успешной отправке формы заказа.*



*Рис. 5. Страница с формой обратной связи.*



*Рис. 6. Страница с модальным окном и успешной отправкой формы.*

**2.3 Реактивность сайта с использованием Vue.js**

Взаимодействие с элементами на странице с использованием функционала Vue осуществляется через использование директив, событий и реактивных данных. Вот примеры основных способов взаимодействия:

Директивы:

v-model: Используется для создания связи между данными и элементами формы. Например, v-model="formData.name" связывает значение элемента <input> с переменной name.

События:

@click: Используется для прослушивания события клика на элементе и выполнения определенных действий. Например, @click="closeModal" вызывает метод closeModal при клике на элемент.

Реактивные данные:

data: Определение реактивных данных в компоненте Vue, которые могут быть использованы для отображения и управления состоянием на странице. Пример:

data () {

return {

sliderImages: [

'images/server1.jpg',

'images/server2.jpg',

'images/server3.jpg'

]

Отображает изображения на странице.

Пример использования взаимодействия с элементами в компоненте Vue:

<template>

<div class="modal" :class="{ 'show': showModal }">

<div class="modal-dialog">

<div class="modal-content">

<div class="modal-header">

<h5 class="modal-title">{{ title }}</h5>

<button type="button" class="btn-close" @click="closeModal"></button>

</div>

<div class="modal-body">

{{ message }}

</div>

<div class="modal-footer">

<button type="button" class="btn btn-primary" @click="closeModal">Закрыть</button>

</div>

</div>

</div>

</div>

</template>

<script>

export default {

props: {

title: String,

message: String

},

computed: {

showModal () {

return !!this.title || !!this.message

}

},

methods: {

closeModal () {

this.$emit('closeModal')

}

}

}

</script>

В данном примере при вводе данных в форме вызывается модальное окно успеха. При нажатии на кнопку вызывается метод CloseModal, закрывающий это окно.

**2.4 Описание взаимодействия с элементами на странице с использованием функционала Vue**

Для заказа аренды сервера пользователь переходит на вкладку оказываемых услуг и заказывает аренду сервера через форму заказа, после заполнения которой появляется сообщение об успешной отправке формы. Условия аренды отправляются пользователю на почту.

При взаимодействии с сайтом, пользователь рассматривает страницу, переходит на вкладку обратной связи и переходит к заполнению формы по ссылке-якорю, в этот момент активируется функция в JS, где пользователь задает вопросы арендодателю.

submitOrderForm () {

this.orderForm.name = ''

this.orderForm.email = ''

this.orderForm.message = ''

this.closeOrderModal()

this.showSuccessMessage = true

setTimeout(() => {

this.showSuccessMessage = false

}, 8000)

}

Далее при заполнении формы и нажатии кнопки, для отправки данных, идёт проверка на то что все поля заполнены и соглашения принят, это реализовано с помощью библиотеки Vue.js.

<div>

<label for="name">Имя:</label>

<input type="text" id="name" v-model="formData.name" required>

</div>

<div>

<label for="email">Электронная почта:</label>

<input type="email" id="email" v-model="formData.email" required>

</div>

<div>

<label for="message">Сообщение:</label>

<textarea id="message" v-model="formData.message" required></textarea>

</div>

<button class = "btn btn-primary bg-dark" type="submit">Отправить</button>

</form>

**Заключение**

Данная страница полностью готова для публикации в интернете, реализован адаптивный режим сайта (корректное отображение для всех устройств). Страница полностью удовлетворяет всем требованиям указанным в ТЗ.

**Приложение**

**App.vue**

<template>

<div>

<NavigationMenu />

<div class="container mt-5">

<router-view />

</div>

<Modal v-if="showModal" :title="modalTitle" :message="modalMessage" @closeModal="closeModal" />

</div>

</template>

<script>

import NavigationMenu from './components/NavigationMenu.vue'

import Modal from './components/TheModal.vue'

export default {

components: {

NavigationMenu,

Modal

},

data () {

return {

showModal: false,

modalTitle: '',

modalMessage: ''

}

},

methods: {

closeModal () {

this.showModal = false

this.modalTitle = ''

this.modalMessage = ''

}

}

}

</script>

<style>

@import './css/styles.css';

</style>

**NavigationMenu.vue**

<template>

<nav class="navbar navbar-expand-md navbar-dark bg-dark">

<router-link class="navbar-brand" to="/">Аренда серверов</router-link>

<button

class="navbar-toggler"

type="button"

data-toggle="collapse"

data-target="#navbarNav"

aria-controls="navbarNav"

aria-expanded="false"

aria-label="Toggle navigation"

@click="toggleMobileMenu"

>

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbarNav" :class="{ show: isMobileMenuOpen }">

<ul class="navbar-nav">

<li class="nav-item">

<router-link class="nav-link" to="/">Главная</router-link>

</li>

<li class="nav-item">

<router-link class="nav-link" to="/about">О наших серверах</router-link>

</li>

<li class="nav-item">

<router-link class="nav-link" to="/services">Услуги</router-link>

</li>

<li class="nav-item">

<router-link class="nav-link" to="/contact">Контакты</router-link>

</li>

</ul>

</div>

</nav>

</template>

<script>

export default {

data () {

return {

isMobileMenuOpen: false

}

},

methods: {

toggleMobileMenu () {

this.isMobileMenuOpen = !this.isMobileMenuOpen

}

}

}

</script>

**TheSlider.vue**

<template>

<div class="slider-container">

<transition name="slide">

<div class="slider-slide" :style="{ transform:'translateX(-${currentIndex \* 100}%)'}">

<img :src="require('@/assets/' + images[currentIndex])" :alt="'Image ' + (currentIndex + 1)" class="slider-image" />

</div>

</transition>

<div class="slider-controls">

<div class="slider-arrow left" @click="previousImage">

<i class="fas fa-chevron-left"></i>

</div>

<div class="slider-arrow right" @click="nextImage">

<i class="fas fa-chevron-right"></i>

</div>

</div>

</div>

</template>

<script>

export default {

props: {

images: {

type: Array,

required: true

}

},

data () {

return {

currentIndex: 0

}

},

methods: {

previousImage () {

if (this.currentIndex > 0) {

this.currentIndex--

} else {

this.currentIndex = 7

}

},

nextImage () {

if (this.currentIndex < this.images.length - 1) {

this.currentIndex++

} else {

this.currentIndex = 0

}

},

startAutoSlide () {

this.autoSlideInterval = setInterval(() => {

this.nextImage()

}, 2000)

},

stopAutoSlide () {

clearInterval(this.autoSlideInterval)

}

},

mounted () {

// Запуск автоматической прокрутки слайдов при загрузке компонента

this.startAutoSlide()

},

beforeDestroy () {

// Остановка автоматической прокрутки слайдов перед удалением компонента

this.stopAutoSlide()

}

}

</script>

**TheForm.vue**

<template>

<div>

<h2></h2>

<form @submit.prevent="submitForm">

<div>

<label for="name">Имя:</label>

<input type="text" id="name" v-model="formData.name" required>

</div>

<div>

<label for="email">Электронная почта:</label>

<input type="email" id="email" v-model="formData.email" required>

</div>

<div>

<label for="message">Сообщение:</label>

<textarea id="message" v-model="formData.message" required></textarea>

</div>

<button class = "btn btn-primary bg-dark" type="submit">Отправить</button>

</form>

<!-- Модальное окно успеха отправки -->

<div v-if="showSuccessModal" class="modal">

<div class="modal-content">

<h3>Отправлено!</h3>

<p>Ваше сообщение было успешно отправлено.</p>

<button @click="closeModal">Закрыть</button>

</div>

</div>

</div>

</template>

<script>

export default {

data () {

return {

formData: {

name: '',

email: '',

message: ''

},

showSuccessModal: false

}

},

methods: {

submitForm () {

// Ваша логика отправки формы обратной связи

// Вы можете отправить данные на сервер или выполнить другие действия

// Для примера, мы просто открываем модальное окно успеха отправки

this.showSuccessModal = true

},

closeModal () {

this.showSuccessModal = false

}

}

}

</script>

<style scoped>

form div {

margin-bottom: 10px;

}

label {

display: block;

font-weight: bold;

}

input,

textarea {

width: 30%;

}

/\* Стили модального окна \*/

.modal {

position: center;

top: 25%;

left: 30%;

width: 40%;

height: 50%;

background-color: rgba(0, 0, 0, 0);

display: flex;

justify-content: center;

align-items: center;

}

.modal-content {

background-color: #bbbbbb;

padding: 20px;

border-radius: 4px;

text-align: center;

}

.modal button {

margin-top: 10px;

}

</style>

**TheModal.vue**

<template>

<div class="modal" :class="{ 'show': showModal }">

<div class="modal-dialog">

<div class="modal-content">

<div class="modal-header">

<h5 class="modal-title">{{ title }}</h5>

<button type="button" class="btn-close" @click="closeModal"></button>

</div>

<div class="modal-body">

{{ message }}

</div>

<div class="modal-footer">

<button type="button" class="btn btn-primary" @click="closeModal">Закрыть</button>

</div>

</div>

</div>

</div>

</template>

<script>

export default {

props: {

title: String,

message: String

},

computed: {

showModal () {

return !!this.title || !!this.message

}

},

methods: {

closeModal () {

this.$emit('closeModal')

}

}

}

</script>

**HomePage.vue**

<template>

<div class="tab-content">

<h1 class="tab-title">Аренда серверов</h1>

<div class="slider-container">

<Slider :images="sliderImages" />

</div>

<p class="tab-text">

Добро пожаловать на нашу платформу аренды серверов! Мы предоставляем широкий спектр высококачественных серверов для удовлетворения ваших потребностей в хостинге и облачных вычислениях.

</p>

<p class="tab-text">

Независимо от того, нужен вам сервер для хранения и обработки данных, развертывания веб-приложений или запуска игрового сервера, у нас есть подходящий вариант для вас. Мы предлагаем гибкую систему тарифов, которая позволяет вам выбрать ресурсы и мощность сервера в соответствии с вашими требованиями.

</p>

<p class="tab-text">

Наши серверы обеспечивают высокую производительность, надежность и безопасность данных. Мы предлагаем возможности масштабирования, резервного копирования и мониторинга, чтобы у вас была полная контроль и гибкость в управлении вашими серверами.

</p>

<p class="tab-text">

Мы готовы помочь вам с выбором и настройкой сервера, чтобы удовлетворить ваши уникальные потребности. Свяжитесь с нами сегодня и начните использовать мощные серверные ресурсы для вашего бизнеса или проекта.

</p>

<div>

</div>

</div>

</template>

<script>

import Slider from '@/components/TheSlider.vue'

export default {

name: 'HomePage',

components: {

Slider

},

data () {

return {

sliderImages: [

'images/1.jpg',

'images/2.jpg',

'images/3.jpg',

'images/4.jpg',

'images/5.jpg',

'images/6.jpg',

'images/7.jpg',

'images/8.jpg'

// Добавьте пути к изображениям, которые вы хотите отобразить в слайдере

]

}

}

}

</script>

**AboutUs.vue**

<template>

<div class="tab-content">

<h1 class="tab-title">Преимущества аренды серверов</h1>

<p class="tab-text">

Аренда серверов предлагает множество преимуществ для вашего бизнеса или проекта. Вот некоторые из основных преимуществ, которые вы получаете при выборе нашей услуги аренды серверов:

</p>

<ul class="tab-list">

<li>Гибкость и масштабируемость: Вы можете легко настроить серверные ресурсы в соответствии с вашими потребностями. Масштабируйте ресурсы при необходимости, чтобы обеспечить оптимальную производительность.</li>

<li>Надежность и стабильность: Наши серверы работают на надежном оборудовании и обеспечивают высокую доступность. Ваше приложение или веб-сайт будут стабильно функционировать без простоев.</li>

<li>Безопасность данных: Мы обеспечиваем высокий уровень защиты данных. Все серверы находятся в безопасных центрах обработки данных с современными мерами безопасности.</li>

<li>Техническая поддержка: Мы предоставляем профессиональную техническую поддержку 24/7. Наша команда экспертов всегда готова помочь вам с любыми вопросами или проблемами, связанными с серверами.</li>

<li>Экономическая эффективность: Аренда серверов позволяет сэкономить деньги по сравнению с покупкой и обслуживанием собственного серверного оборудования. Вы платите только за использованные ресурсы.</li>

</ul>

<p class="tab-text">

Аренда серверов предоставляет вам возможность сконцентрироваться на развитии вашего бизнеса, а не на управлении серверами. Свяжитесь с нами сегодня и начните получать все преимущества аренды серверов для вашего успеха.

</p>

</div>

</template>

<script>

export default {

}

</script>

**OurServices.vue**

<template>

<div class="our-services">

<div v-if="showSuccessMessage" class="success-message">

<h3>Информация об условиях аренды отправлена на вашу электронную почту!</h3>

</div>

<h2>Выберите конфигурацию сервера:</h2>

<div class="services">

<div

class="service-item"

v-for="service in services"

:key="service.id"

@mouseover="handleMouseOver(service.id)"

@mouseout="handleMouseOut(service.id)"

>

<img :src="service.image" :alt="service.title" />

<div class="overlay" v-show="hoveredService === service.id">

<h4>{{ service.title }}</h4>

<p>{{ service.description }}</p>

<button @click="openOrderModal(service)">Заказать</button>

</div>

</div>

</div>

<!-- Модальное окно для заказа -->

<div class="modal " v-if="showModal">

<div class="modal-content">

<span class="close" @click="closeOrderModal">&times;</span>

<h3>Форма заказа</h3>

<form @submit.prevent="submitOrderForm">

<label for="name">Имя:</label>

<input type="text" id="name" v-model="orderForm.name" required>

<label for="email">Email:</label>

<input type="email" id="email" v-model="orderForm.email" required>

<label for="message">Сообщение:</label>

<textarea id="message" v-model="orderForm.message"></textarea>

<button type="submit">Отправить</button>

</form>

</div>

</div>

</div>

</template>

<script>

export default {

data () {

return {

services: [

{

id: 1,

title: 'Эконом',

description: 'Мог бы захостить на телефоне',

image: require('@/assets/images/econom.jpg')

},

{

id: 2,

title: 'Золотая середина',

description: 'Вся мощь домашнего ПК',

image: require('@/assets/images/medium.jpg')

},

{

id: 3,

title: 'Суперкомпьютер',

description: 'Арендуй сервер Яндекса',

image: require('@/assets/images/high-end.jpg')

}

],

hoveredService: null,

showModal: false,

showSuccessMessage: false, // Состояние сообщения об успешной отправке заказа

orderForm: {

name: '',

email: '',

message: ''

}

}

},

methods: {

handleMouseOver (serviceId) {

this.hoveredService = serviceId

},

handleMouseOut (serviceId) {

if (this.hoveredService === serviceId) {

this.hoveredService = null

}

},

openOrderModal (service) {

this.showModal = true

},

closeOrderModal () {

this.showModal = false

},

submitOrderForm () {

this.orderForm.name = ''

this.orderForm.email = ''

this.orderForm.message = ''

this.closeOrderModal()

this.showSuccessMessage = true

setTimeout(() => {

this.showSuccessMessage = false

}, 8000)

}

}

}

</script>

**ContactUs.vue**

<template>

<div>

<h1 class="tab-title">Обратная связь</h1>

<p class="tab-text">

Мы ценим ваше мнение и готовы ответить на ваши вопросы. Пожалуйста, заполните форму ниже, чтобы связаться с нами.

</p>

<Form></Form>

</div>

</template>

<script>

import Form from '@/components/TheForm.vue'

export default {

name: 'TheTab3',

components: {

Form

}

}

</script>

**styles.css**

body {

text-align: justify;

}

.slider-container {

text-align: center;

}

.btnleft {

margin-right: 3%;

width: 80px;

height: 40px;

background-color: rgba(0, 0, 0, 0.5);

color: #fff;

font-size: 20px;

text-align: center;

}

.btnright {

width: 80px;

height: 40px;

background-color: rgba(0, 0, 0, 0.5);

color: #fff;

font-size: 20px;

text-align: center;

}

.slider-controls {

padding: 10px 20px;

border: none;

border-radius: 4px;

color: #1e1111;

font-size: 16px;

font-weight: bold;

}

.slider-arrow {

width: 20%;

height: 50px;

display: inline-block;

background-color: #ffffff;

cursor: pointer;

}

.slider-arrow.left:before {

margin-top: 15px;

content: '';

width: 15px;

height: 15px;

display: inline-block;

border-top: 10px solid transparent;

border-bottom: 10px solid transparent;

border-right: 10px solid #000000;

}

.slider-arrow.right:before {

content: '';

margin-top: 15px;

width: 15px;

height: 15px;

display: inline-block;

border-top: 10px solid transparent;

border-bottom: 10px solid transparent;

border-left: 10px solid #020202;

}

.margin-center div{

margin: 30px auto;

}

@media (max-width: 550px){

img{

width: 300px;

height: 300px;

}

#carouselExampleDark{

display: none;

}

}

@media (max-width: 800px){

img{

width: 300px;

height: 300px;

}

}

.our-services {

text-align: center;

padding: 20px;

}

.server-config {

margin-bottom: 20px;

}

.services {

display: flex;

justify-content: center;

align-items: center;

flex-wrap: wrap;

}

.service-item {

position: relative;

margin: 10px;

width: 400px;

}

.service-item img {

width: 100%;

}

.success-message {

margin-left: 27%;

width: 46%;

border-radius: 0.5rem;

flex-direction: column;

justify-content: center;

align-items: center;

background-color: rgba(0, 0, 0, 0.7);

color: #fff;

}

.overlay {

position: absolute;

top: 0;

left: 0;

width: 100%;

height: 100%;

display: flex;

flex-direction: column;

justify-content: center;

align-items: center;

background-color: rgba(0, 0, 0, 0.7);

color: #fff;

opacity: 0;

transition: opacity 0.3s ease;

}

.overlay h4 {

margin-bottom: 10px;

}

.service-item:hover .overlay {

opacity: 1;

}

.modal {

display: flex;

justify-content: center;

align-items: center;

position: fixed;

top: 0;

left: 0;

width: 100%;

height: 100%;

background-color: rgba(0, 0, 0, 0.7);

z-index: 9999;

transition-property: transform;

transition-duration: 500ms;

transition-timing-function: linear;

animation: modalFadeIn 0.5s ease-in-out;

}

.modal-content {

position: relative;

background-color: #fff;

padding: 20px;

border-radius: 4px;

text-align: center;

width: 400px;

}

.close {

position: absolute;

top: 10px;

right: 10px;

cursor: pointer;

}

form {

margin-top: 20px;

}

form label {

display: block;

margin-bottom: 10px;

}

form input,

form textarea {

width: 100%;

padding: 8px;

margin-bottom: 10px;

}

form button {

padding: 10px 20px;

background-color: #007bff;

color: #fff;

border: none;

cursor: pointer;

}

form button:hover {

background-color: #0056b3;

}

@keyframes modalFadeIn {

from {

opacity: 0;

transform: translateY(-500px);

}

to {

opacity: 1;

transform: translateY(0);

}

}

**index.html**

<!DOCTYPE html>

<html lang="">

<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">

<link rel="icon" href="<%= BASE\_URL %>favicon.ico">

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-GLhlTQ8iRABdZLl6O3oVMWSktQOp6b7In1Zl3/Jr59b6EGGoI1aFkw7cmDA6j6gD" crossorigin="anonymous">

<title><%= htmlWebpackPlugin.options.title %></title>

</head>

<body>

<noscript>

<strong>We're sorry but <%= htmlWebpackPlugin.options.title %> doesn't work properly without JavaScript enabled. Please enable it to continue.</strong>

</noscript>

<div id="app"></div>

<!-- built files will be auto injected -->

</body>

</html>

**router.js**

import Vue from 'vue'

import VueRouter from 'vue-router'

import HomePage from './components/HomePage.vue'

import OurServices from './components/OurServices.vue'

import AboutUs from './components/AboutUs.vue'

import ContactUs from './components/ContactUs.vue'

Vue.use(VueRouter)

const routes = [

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

{ path: '/services', component: OurServices },

{ path: '/about', component: AboutUs },

{ path: '/contact', component: ContactUs }

]

const router = new VueRouter({

mode: 'history',

routes

})

export default router