

титульник

СОДЕРЖАНИЕ

Перечень сокращений и обозначений	3
Введение.....	4
1 Анализ и разработка требований	6
1.1 Назначение и область применения	6
1.2 Постановка задачи.....	6
1.3 Выбор состава технических средств	7
1.4 Выбор технических и программных средств.....	8
2 Проектирование и разработка сайта.....	12
2.1 Проектирование базы данных	12
2.2 Проектирование и разработка базы данных	18
2.3 Разграничение прав доступа.....	19
3 Тестирование и отладка программного обеспечения	29
3.1 Функциональное тестирование	29
4 Инструкция по эксплуатации программного обеспечения.....	31
4.1 Конфигурация программного обеспечения	31
4.2 Установка программного обеспечения	31
4.3 Руководство администратора.....	33
5 Определение затрат на разработку веб-сайта.....	39
5.1 Технико-экономическое обоснование.....	39
5.2 Анализ экономического обоснования и рисков проекта.....	52
6 Охрана труда и техника безопасности при работе на ПК	54
6.1 Общие требования охраны труда.....	54
6.2 Требования охраны труда перед началом работы	54
6.3 Требования охраны труда во время работы	55
Заключение.....	56
Список использованных источников.....	57

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ

В настоящем дипломном проекте применяются следующие сокращения и обозначения:

БД – база данных

ОС – операционная система

ПК – персональный компьютер

ПО – программное обеспечение

СУБД – система управления базами данных

CSS – каскадные таблицы стилей

HTML – язык гипертекстовой разметки

ORM – объектно-реляционное отображение

UML – унифицированный язык моделирования

ВВЕДЕНИЕ

В современном бизнесе многое зависит от рекламы организации. Интернет – это современная рекламная площадка, позволяющая обеспечить приток новых клиентов. Поэтому создание веб-сайта компании является актуальной и значимой задачей, так как он является важнейшим источником информации для клиентов [20].

Целью дипломного проекта является разработка веб-сайта для салона красоты «Lovely Arkhangelsk», предназначенного для ознакомления клиентов с салоном красоты и спектром предоставляемых услуг, а также для ускорения и упорядочения работы администратора.

Создание веб-сайта представляет собой сложную задачу, которая требует объединения разных систем в единую базу в виде веб-сайта. Для этого надо понимать все процессы и задачи, которые требуется реализовать, а также выбрать наиболее подходящие технологии для их исполнения. Для достижения поставленной цели по разработке ПО требуется решить следующие задачи:

- собрать и проанализировать функциональные и эксплуатационные требования к ПО,
- описать работу ПО с помощью UML,
- выбрать состав программных и технических средств для разрабатываемого ПО,
- выделить объекты и атрибуты БД в соответствии с заданием,
- спроектировать БД,
- разработать БД,
- спроектировать интерфейс пользователя,
- реализовать интерфейс пользователя,
- разработать программные модули веб-сайта «Lovely Arkhangelsk»,
- выполнить тестирование и отладку разработанного ПО,
- разработать инструкцию по эксплуатации ПО,

- описать процесс установки разработанного ПО.

Создание веб-сайта салона красоты «Lovely Arkhangelsk» позволит:

- увеличить узнаваемость бренда,
- привлечь новых клиентов,
- повысить уровень лояльности клиентов,
- увеличить продажи услуг,
- снизить расходы на рекламу.

Разработка проекта веб-сайта для салона красоты «Lovely Arkhangelsk» является актуальным и перспективным проектом, который позволит повысить конкурентоспособность салона и увеличить его прибыль.

1 Анализ и разработка требований

1.1 Назначение и область применения

Салон красоты «Lovely Arkhangelsk» предоставляет широкий спектр услуг в области красоты и ухода за внешностью. Основной задачей салона является удовлетворение потребностей клиентов в качественном и профессиональном уходе за волосами, кожей, ногтями и телом. Салон ориентирован на оказание услуг, направленных на улучшение внешнего вида, поддержание здоровья кожи и волос, а также повышение самооценки и уверенности клиентов.

Основными видами деятельности салона красоты являются:

- оказание парикмахерских услуг,
- уход за лицом,
- маникюр и педикюр,
- удаление тату,
- продажа профессиональной косметики и средств ухода за волосами и лицом.

Сайт необходим салону для информирования клиентов об услугах.

1.2 Постановка задачи

Необходимо разработать веб-сайт для салона красоты «Lovely Arkhangelsk», который должен выполнять следующие функции:

- представлять информацию о салоне, его услугах, ценах и команде специалистов,
- обеспечивать возможность онлайн-записи на услуги.

Сайт должен способствовать привлечению новых клиентов, и улучшению взаимодействия с уже имеющимися клиентами салона.

Основными требованиями к сайту являются:

- простота использования,
- кроссплатформенность,
- привлекательный дизайн в нежных цветах,
- панель администратора для управления сайтом,
- возможность получения пользователями информации об услугах салона красоты,
- возможность просмотра портфолио мастера, а также ознакомления с изображениями выполненных мастерами работ,
- возможность регистрации и авторизации пользователя,
- возможность записи на выбранную услугу,
- возможность редактирования и добавления информации о мастерах и описания услуг,
- возможность просмотра цен на выбранную услугу.

1.3 Описание алгоритма функционирования системы

Для перехода на сайт требуется после открытия браузера в адресной строке ввести адрес сайта и нажать на кнопку «Перейти». Будет загружена главная страница сайта.

Клиент сайта должен иметь возможность посмотреть основную информацию о мастерах, работах, услугах и ценах. Если у клиента есть аккаунт, то он должен иметь возможность записаться на выбранную услугу к выбранному мастеру и на доступное для записи время, а также посмотреть свою историю услуг.

Мастер салона красоты должен иметь возможность посмотреть какая работа предстоит ему в ближайшее время.

Администратор сайта должен иметь возможность просмотреть и редактировать информацию о пользователях, создавать, редактировать и удалять данные об услугах и мастерах.

На основе определенных целей, задач и требований при помощи онлайн-инструмента draw.io спроектирована диаграмма вариантов использования, представленная на рисунке 1.

Диаграмма вариантов использования (Use Case Diagram) — это инструмент UML, используемый для моделирования функциональных требований системы. Она отображает взаимодействие между пользователями (актерами) и системой, показывая, какие задачи (варианты использования) могут быть выполнены системой.

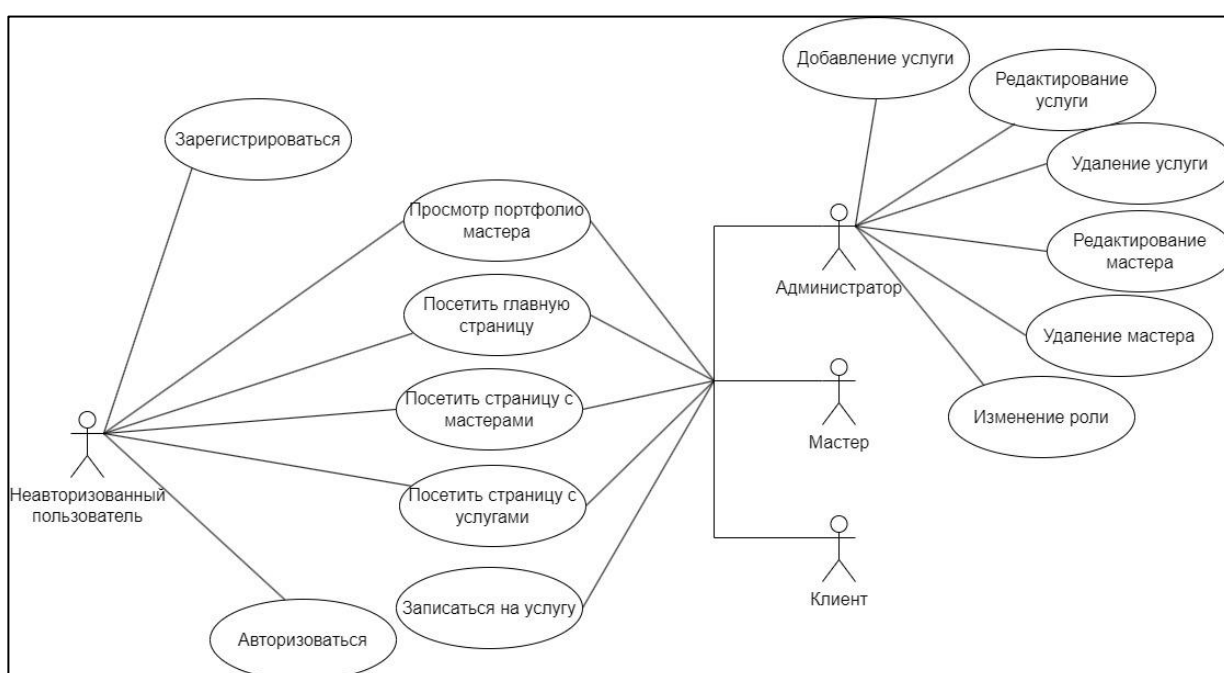


Рисунок 1 – Диаграмма вариантов использования сайта «Lovely Arkhangelsk»

1.4 Выбор технических и программных средств

На основе цели проекта, направленного на создание функционального и эстетически привлекательного веб-сайта для салона красоты «Lovely Arkhangelsk», был выбран следующий стек технологий. Этот стек обеспечит надежность, производительность и удобство разработки, а также поддержание и масштабирование сайта. Используемые технологии и инструменты:

- HTML, язык разметки гипертекста используется для создания структуры веб-страниц [12]. HTML определяет элементы, такие как заголовки, абзацы, ссылки, изображения и другие компоненты страницы,
- CSS, таблицы стилей каскадные используются для описания внешнего вида и форматирования HTML-элементов [6]. CSS позволяет изменять шрифты, цвета, макет и другие визуальные аспекты страницы,
- SASS, расширение синтаксиса CSS, предоставляющее дополнительные возможности, такие как вложенные правила, переменные, миксины и функции [19]. SASS упрощает написание и поддержание стилей,
- TypeScript, дополнение для JavaScript [14], добавляющее статическую типизацию и дополнительные возможности для улучшения разработки и отладки кода [21],
- React, библиотека JavaScript для создания пользовательских интерфейсов [9]. React позволяет создавать компоненты, которые могут управлять своим состоянием и жизненным циклом,
- Redux, предсказуемый контейнер состояния для JavaScript-приложений, предоставляющий централизованное хранилище для управления состоянием приложения [11],
- RTK Query, расширение для Redux Toolkit, обеспечивающее мощный и эффективный инструмент для управления состоянием данных и их кэширования, а также для работы с API [18],
- MySQL, система управления реляционными базами данных (СУБД), использующая язык SQL для создания, изменения и управления данными в базе данных.
- Node.js, среда выполнения JavaScript на стороне сервера, позволяющая создавать быстрые и масштабируемые сетевые приложения [16],
- Express, минималистичный и гибкий веб-фреймворк для Node.js, обеспечивающий набор функций для создания веб- и мобильных приложений [13],
- Sequelize, ORM для Node.js, который поддерживает различные базы

данных, включая MySQL [10]. Sequelize предоставляет удобный API для взаимодействия с базой данных, позволяя определять модели, выполнять запросы и управлять данными с помощью JavaScript.

Для разработки проекта с технологиями, приведенными выше, потребуется следующее ПО:

- PhpStorm, интегрированная среда разработки (IDE) для PHP и веб-разработки [8]. Поддерживает HTML, CSS, JavaScript, TypeScript, а также интеграцию с различными системами контроля версий и инструментами разработки,
- MySQL Workbench, визуальный инструмент для администрирования MySQL-баз данных, предоставляющий возможности для проектирования, моделирования, создания и управления базами данных [15],
- Postman, средство для тестирования запросов к серверной части проекта [17],
- Draw.io – бесплатный онлайн-инструмент для создания диаграмм, блок-схем, макетов, дизайнов [7].

Этот стек и ПО обеспечат эффективную разработку, управление и поддержание веб-сайта для салона красоты «Lovely Arkhangelsk».

Для обеспечения стабильной и эффективной работы веб-сайта необходимо установить определенные минимальные требования как на стороне клиента, так и на стороне сервера. Эти требования гарантируют, что система будет работать без сбоев, обеспечивая пользователям оптимальный опыт взаимодействия с сайтом, а также надежность и безопасность данных. Учитывая особенности проекта и используемые технологии, минимальные требования для функционирования системы следующие:

а) клиентская сторона:

- ОС Windows 7 и выше / macOS 10.12 и выше;
- процессор 2 ядра (рекомендовано: 4 ядра);
- оперативная память 4 ГБ (рекомендовано: 8 ГБ);
- СУБД MySQL 8.0;

- дисковое пространство 20 ГБ и больше;

б) серверная сторона:

- ОС Ubuntu Linux 22 64bit;
- веб-сервер Nginx или Apache;
- процессор 1 ГГц и выше;
- оперативная память 2 ГБ и выше;
- широкополосное подключение к сети Интернет со скоростью не менее 1 Мбит/с.

2 Проектирование и разработка сайта

2.1 Разработка дизайна и структуры сайта

У салона красоты уже есть разработанный логотип, представленный на рисунке 2, и на основе его цветов разработана палитра цветов сайта, представленная на рисунке 3.



Рисунок 2 – Логотип салона красоты «Lovely Arkhangelsk»

FE313AFF	E7E7E7FF	E8E8E8FF	000000FF	F7F7F7FF
----------	----------	----------	----------	----------

Рисунок 3 – Цветовая палитра сайта

На основе требований заказчика к проекту определены следующие элементы сайта:

а) страницы:

- главная;
- мастера;
- услуги;
- работы;
- портфолио мастера;
- запись на услугу;
- авторизация;
- регистрация;
- панель администратора;
- история услуг;

б) модальные окна:

- добавление услуги;
- редактирование услуги;
- редактирование мастера;
- прайс-лист выбранной услуги.

На каждой странице сайта должна присутствовать «шапка», включающая следующие элементы:

- логотип сайта по середине,
- горизонтальное меню навигации по сайту.

В «подвале» сайта должны присутствовать следующие элементы:

- логотип сайта по середине,
- контактный номер телефона салона красоты с левой стороны,
- время работы салона,
- адрес.

На главной странице сайта должны быть расположены слайдеры с карточками мастеров и с фотографиями выполненных работ. Вид прототипа главной страницы сайта «Lovely Arkhangelsk» представлен на рисунке 4.

На странице с мастерами салона должны быть расположены слайдеры с карточками мастеров, при нажатии на которые происходит перенаправление в

портфолио выбранного мастера. Вид прототипа страницы «Мастера» сайта «Lovely Arkhangelsk» представлен на рисунке 5.

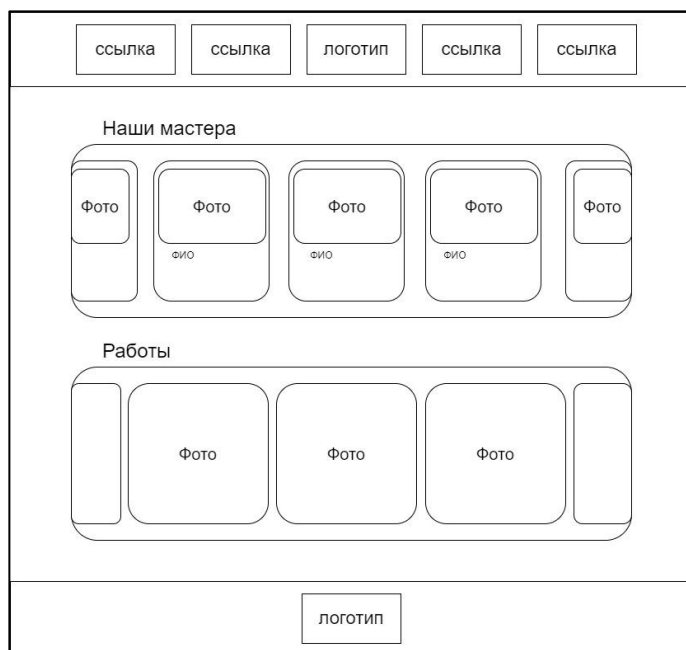


Рисунок 4 – Вид прототипа страницы «Главная» сайта «Lovely Arkhangelsk»

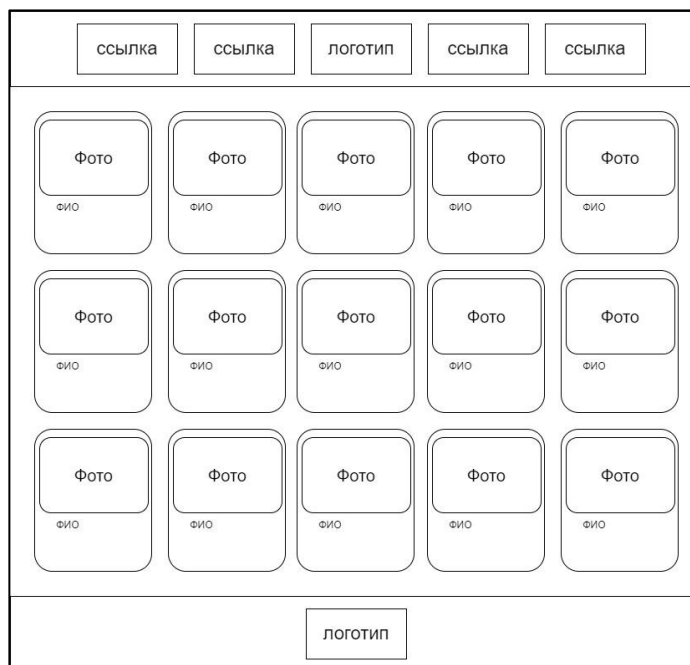


Рисунок 5 – Вид прототипа страницы «Мастера» сайта «Lovely Arkhangelsk»

На странице «Работы» должны быть расположены плитки из фотографий всех завершенных услуг. Вид прототипа страницы с работами мастеров сайта «Lovely Arkhangelsk» представлен на рисунке 6.



Рисунок 6 – Вид прототипа страницы «Работы» сайта «Lovely Arkhangelsk»

На странице «Услуги» должны быть расположены плитки из услуг и их наименований. Нажатие на каждую из них Вид прототипа страницы «Услуги» сайта «Lovely Arkhangelsk» представлен на рисунке 6.

На странице «Вход» должны быть расположены текстовые поля номера телефона клиента и пароль, а также кнопки входа и создания учетной записи. Вид прототипа страницы «Вход» сайта «Lovely Arkhangelsk» представлен на рисунке 7.

ссылка ссылка логотип ссылка ссылка

Вход

Телефон

Пароль

Вход

Создать аккаунт

логотип

Рисунок 7 – Вид прототипа страницы «Вход» сайта «Lovely Arkhangelsk»

Для создания сайта «Lovely Arkhangelsk» каждая страница реализована как отдельный компонент в React. Такой подход позволяет улучшить структуру и управление кодом, делая его более модульным и поддерживаемым. Использование компонентов в React обеспечивает повторное использование кода и упрощает разработку, тестирование и обновление отдельных частей веб-сайта.

Фрагмент кода страница «Главная» представлен листингом 1.

Листинг 1 – Фрагмент кода страницы «Главная»

```
// Импорт необходимых библиотек и комопнентов
import {FC} from "react";
import YandexMaps from "../YandexMaps/YandexMaps.tsx";
import {motion} from "framer-motion";
import { SliderWorks } from '../SliderWorks/SliderWorks.tsx';
import { Link } from 'react-router-dom';
import { SliderEmployees } from
'../SliderEmployees/SliderEmployees.tsx';
// Инициализация компонента PageHome
const PageHome: FC = () => {
  return(
    <>
      <motion.div
        className="page home-container"
        initial={{ y: -100, opacity: 0 }}
        animate={{ y: 0, opacity: 1 }}
        exit={{ y: 100, opacity: 0 }}
      >
        <h1>Главная</h1>
        <div className="home-elem">
          <Link className={ 'link' }
to={"/employees"}>Наши мастера</Link>
          <SliderEmployees/> { /* Слайдер с
карточками работников */ }
        </div>
        <div className="home-elem">
          <Link className={ 'link' }
to={"/works"}>Работы мастеров</Link>
          <SliderWorks employerId={0} /> { /*
Слайдер с плитками работ */ }
        </div>
        <YandexMaps /> { /* Использование API сервиса
"Яндекс.Карты" */ }
      </motion.div>
    </>
  )
}

// Экспорт созданного компонента для возможности
использования в будущем
export default PageHome
```

После завершения разработки интерфейса веб-сайта, можно утверждать, что создана пользовательская среда, которая сочетает в себе интуитивную понятность и привлекательность. Интерфейс обеспечивает комфортную

навигацию пользователя благодаря логической структуре и удобному расположению элементов управления.

2.2 Проектирование и разработка базы данных

Проектирование базы данных – ключевой этап разработки веб-сайта, обеспечивающий структурированное хранение и доступ к данным.

В данном разделе рассматриваются основные этапы проектирования базы данных для веб-сайта салона красоты «Lovely Arkhangelsk». Основная цель проектирования — создание эффективной и надежной структуры данных, удовлетворяющей всем требованиям пользователей и обеспечивающей высокую производительность системы. Правильно спроектированная база данных позволит организовать хранение информации о мастерах, услугах, клиентах и их бронированиях, а также обеспечит возможно

Для разработки веб-сайта салона красоты «Lovely Arkhangelsk» определены следующие функциональные и нефункциональные требования к базе данных:

а) функциональные требования:

- учет мастеров салона;
- учет услуг салона;
- учет клиентов;
- бронирование услуг;

б) нефункциональные требования:

- производительность;
- надежность;
- безопасность;
- масштабируемость.

На основе выявленных требований составлена физическая модель БД для веб-сайта салона красоты «Lovely Arkhangelsk», представленная на рисунке 8.

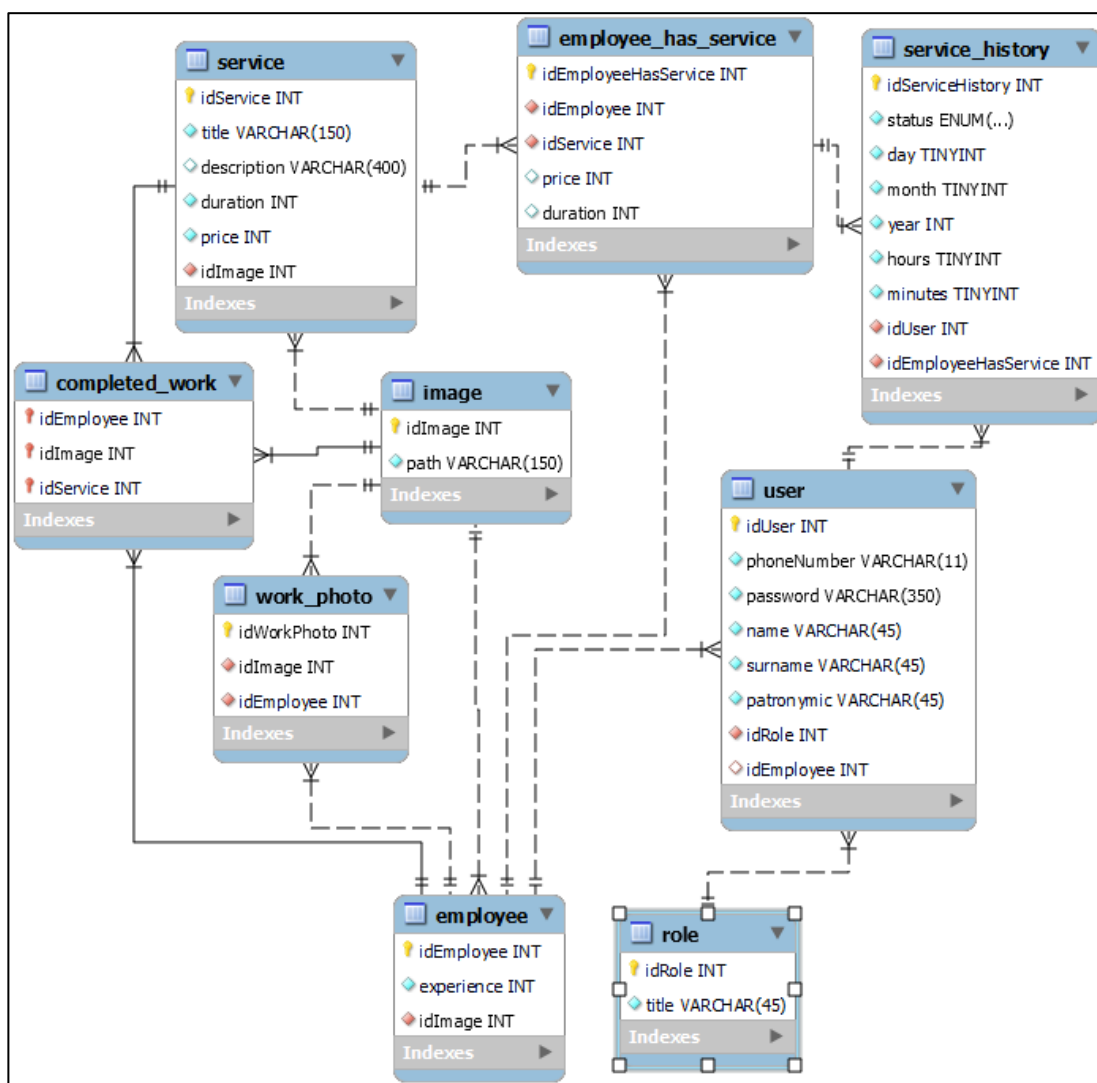


Рисунок 8 – MySQL Workbench. Концептуальная модель БД

2.3 Разграничение прав доступа пользователей

В качестве роли с наивысшим уровнем доступа предусмотрена роль «Администратор». Эта роль имеет полный доступ к административной панели сайта, обеспечивая возможность изменения информации на сайте. Фрагмент кода страницы «Панель администратора» представлен листингом 2. Панель администратора защищена от несанкционированного доступа, что означает, что неавторизованные пользователи и пользователи без роли «Администратор» не имеют возможности получить к ней доступ. Фрагмент кода метода API авторизации представлен листингом 3.

Листинг 2 – Фрагмент кода страницы «Панель администратора»

```
// Импорт необходимых библиотек и компонентов
import {FC} from "react";
import AdminLeftMenu from "../AdminLeftMenu.tsx";
import AdminBottomMenu from "../AdminBottomMenu.tsx";
import {useTypedSelector} from "../../store/hooks/redux.ts";
import {SectionStatistic} from
"./SectionStatistic/SectionStatistic.tsx";
import {SectionEmployers} from
"./SectionEmployers/SectionEmployers.tsx";
import {SectionClients} from
"./SectionClients/SectionClients.tsx";
import {SectionServices} from
"./SectionServices/SectionServices.tsx";
import { Navigate } from 'react-router-dom';

// Инициализация компонента
const PageAdmin: FC = () => {
  const {sectionName} = useTypedSelector(state =>
state.adminPanelReducer);
  const {user} = useTypedSelector(state => state.auth)

  return(
    <>
      {
        // Если пользователь не авторизован или у него
нет прав администратора, перенаправить на страницу авторизации
(!user || (user.role.title !== 'Администратор'))
        && <Navigate to='/login' />
      }
      <div
        className="page-admin__wrapper"
      >
        <AdminLeftMenu />
        <AdminBottomMenu />
        <div className="page admin-container">
          {
            // Проверка переменной на выбранный
пункт бокового меню
            sectionName == "statistic" ?
            <SectionStatistic /> :
            sectionName == "employers" ?
            <SectionEmployers /> :
            sectionName == "services" ?
            <SectionServices /> :
            <SectionClients />
          }
        </div>
      </div>
    </>
  )
}
```

```

    )
}

// Экспорт созданного компонента для использования в будущем
export default PageAdmin

```

Листинг 3 – Фрагмент кода метода авторизации

```

// Импорт необходимых библиотек и компонентов
import {baseApi} from "../api.ts";
import {login} from "../slices/authSlice.ts";

// Инициализация объекта методов для работы с учетной записью
пользователя
export const authApi = baseApi.injectEndpoints({
  endpoints: builder => ({
    // Инициализация метода авторизации
    login: builder.mutation<IAuthorizationResult,
    ILoginCommand>({
      query: (data) => ({
        url: `/auth/login`,
        method: "POST",
        body: data,
      }),
      async onQueryStarted(_, {dispatch,
queryFulfilled}){
        try{
          const {data} = await queryFulfilled
          dispatch(login(data))
        } catch (err){
          console.log(err)
        }
      },
    }),
    // Инициализация метода регистрации
    registration: builder.mutation<{user: IUser, message:
string}, IRegistrationCommand>({
      query: (data) => ({
        url: `/auth/registration`,
        method: "POST",
        body: data
      }),
      invalidatesTags: ['Users']
    })
  }),
})
// Экспорт объекта методов
export const {
  useLoginMutation,
  useRegistrationMutation
} = authApi

```

Вид страницы «Вход» представлен на рисунке 9. Вид страницы «Создание аккаунта» представлен на рисунке 10.

The screenshot shows the login page of the 'Lovely Arkhangelsk' website. At the top, there is a navigation bar with links: Главная, Мастера, Работы, Услуги, Записаться, and Войти. The logo 'Lovely Arkhangelsk' is centered in the navigation bar. Below the navigation bar, the title 'Вход' is displayed. There are two input fields: 'Телефон' (Phone) and 'Пароль' (Password). Below these fields is a red-bordered button labeled 'Вход'. Underneath the button is a link 'Создать аккаунт' (Create account). The footer contains the phone number '+79118727440', the logo 'Lovely Arkhangelsk', and the address 'г. Архангельск, д. Часовенское 31, строение 2' along with the operating hours '10:00-19:00 Каждый день'.

Рисунок 9 – Lovely Arkhangelsk. Вид страницы «Вход»

The screenshot shows the 'Создание аккаунта' (Create account) page of the 'Lovely Arkhangelsk' website. The navigation bar is identical to the login page. The title 'Создание аккаунта' is displayed. There are six input fields: 'Телефон' (Phone), 'Пароль' (Password), 'Подтверждение пароля' (Confirm password), 'Фамилия' (Surname), 'Имя' (Name), and 'Отчество' (Patronymic). Below these fields is a red-bordered button labeled 'Создать аккаунт' (Create account). Underneath the button is a link 'Уже есть аккаунт' (Already have an account). The footer is identical to the login page.

Рисунок 10 – Lovely Arkhangelsk. Вид страницы «Создание аккаунта»

Вид страницы «Панель администратора» с выбранным разделом «Статистика» представлен на рисунке 11.

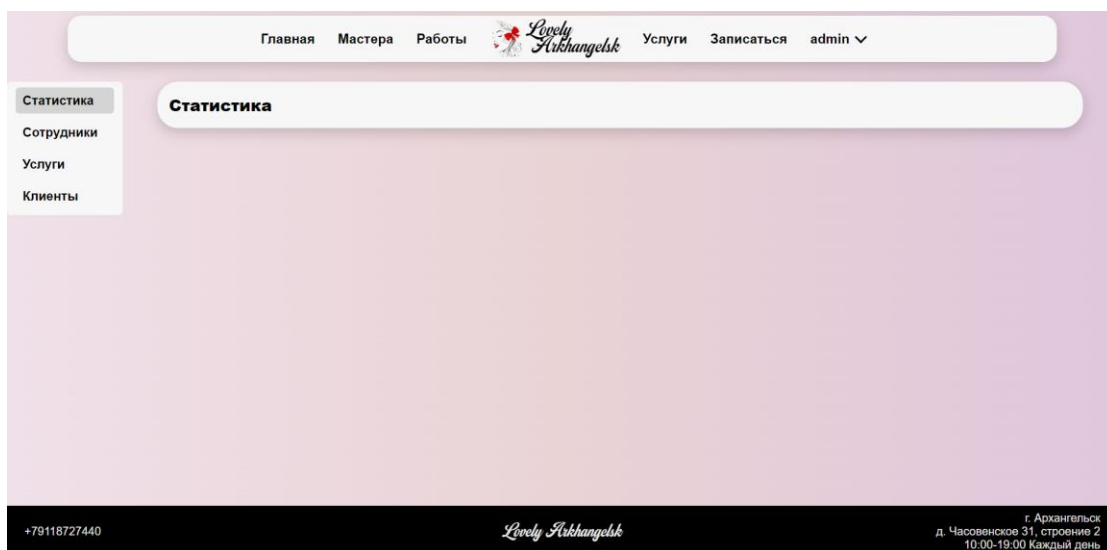


Рисунок 11 – Lovely Arkhangelsk. Вид страницы «Панель администратора» с выбранным разделом «Статистика»

Вид страницы «Панель администратора» с выбранным разделом «Сотрудники» представлен на рисунке 12.

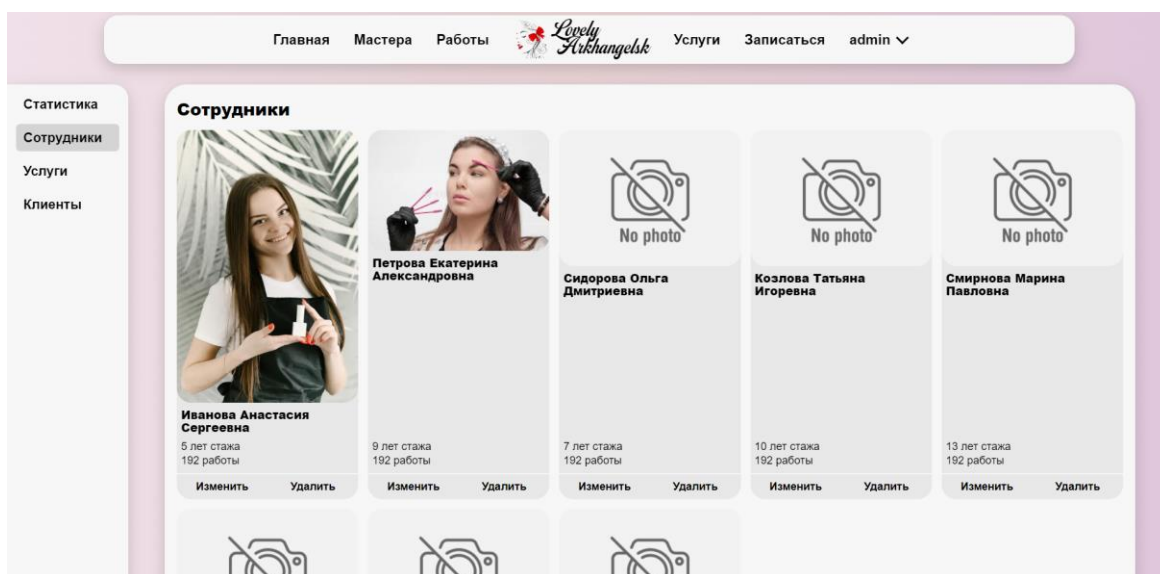


Рисунок 12 – Lovely Arkhangelsk. Вид страницы «Панель администратора» с выбранным разделом «Сотрудники»

Вид страницы «Панель администратора» с выбранным разделом «Услуги» представлен на рисунке 13.

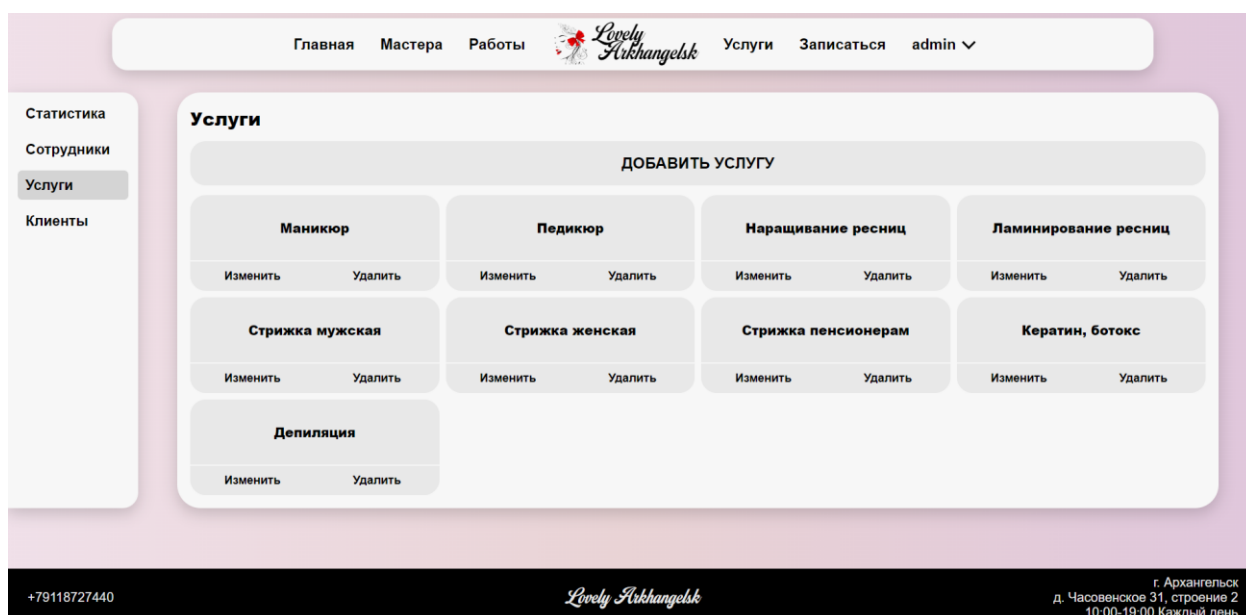


Рисунок 13 – Lovely Arkhangelsk. Вид страницы «Панель администратора» с выбранным разделом «Услуги»

Для записи клиентов на выбранную услугу реализован отдельный модуль, расположенный на странице «Записаться», представленной на рисунке 14. Фрагмент кода страницы «Записаться» представлен листингом 4.

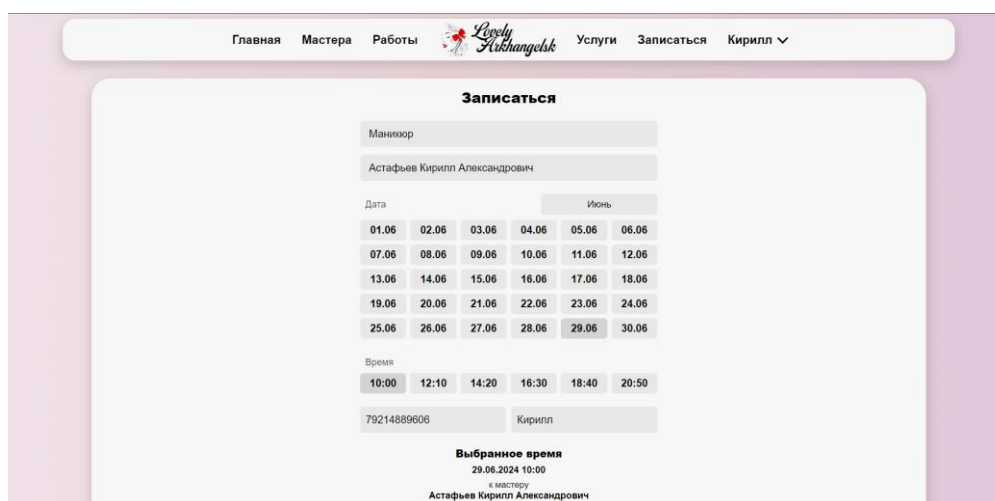


Рисунок 14 – Lovely Arkhangelsk. Вид страницы «Записаться»

Листинг 4 – Фрагмент кода страницы «Записаться»

```
// Подключение необходим библиотек и компонентов
import React, { FC, useState } from 'react';
import { motion } from "framer-motion";
import { SelectEmployee } from "../SelectEmployee.tsx";
import { SelectService } from "../SelectService.tsx";
import { useTypedSelector } from "../../store/hooks/redux.ts";
import { RadiosDate } from "../RadiosDate.tsx";
import { RadiosTime } from "../RadiosTime.tsx";
import { getMessageFromError } from '../../utils/errors.ts';
import { message } from 'antd';
import { Navigate } from 'react-router-dom';
import { useGetEmployeeHasServiceByIdQuery } from
'../../api/methods/employeeHasServiceApi.ts';
import { formatNumber } from '../../utils/dateFormatter.ts';
import { isNumeric } from '../../utils/isNumeric.ts';
import { useCreatePostMutation } from
'../../api/methods/timeApi.ts';

// Инициализация компонента
const PageTime: FC = () => {
  const { user } = useTypedSelector(state => state.auth)
  const [submitting, setSubmitting] = useState(false);
  const [phoneNumber, setPhoneNumber] = useState<string>(user
? user.phoneNumber: '')
  const [clientName, setClientName] = useState<string>(user ?
user.name : '')
  const [createPost] = useCreatePostMutation()
  const timeSlice = useTypedSelector(state =>
state.timeFormReducer)
  const queryResult = useGetEmployeeHasServiceByIdQuery({
    idService: timeSlice.idService,
    idEmployee: timeSlice.idEmployee
  }, { skip: !timeSlice.idService || !timeSlice.idEmployee
});

  const serviceInfo = queryResult.data;

  // Асинхронное событие при отправке формы
  const handleOnSubmit = async (e:
React.FormEvent<HTMLFormElement>, setSubmitting: (submitting:
boolean) => void) => {
    e.preventDefault();
    setSubmitting(true);

    if(!isNumeric(phoneNumber)){
      message.info('Некорректный номер телефона', 3)
      setSubmitting(false);
      return;
    }
    if(user?.idEmployee === timeSlice.idEmployee){
```

```

        message.info('Нельзя записаться на услугу к самому себе', 3)
        setSubmitting(false);
        return;
    }
    if(user){
        const response = await createPost({
            idUser: user.idUser,
            idEmployee: timeSlice.idEmployee,
            idService: timeSlice.idService,
            dateYear: timeSlice.dateYear,
            dateMonth: timeSlice.dateMonth,
            dateDay: timeSlice.dateDay,
            dateHours: timeSlice.dateHours,
            dateMinutes: timeSlice.dateMinutes
        })

        if ('error' in response) {
            const errorString =
getMessageFromError(response)
            if (errorString == 'unknown') {
                message.error(`Возникла серверная
ошибка.\nПовторите действие позже`, 3)
            } else {
                message.error(`${errorString}`, 3)
            }
            setSubmitting(false);
        } else {
            message.info(response.data.message, 3)
            setTimeout(() => {
                window.location.reload();
            }, 2000);
        }
    }
}
return (
    <>
    {
        !user && <Navigate to='/login' />
    }

    <motion.div
        className="page time-container"
        initial={{ opacity: 0.3 }}
        animate={{ opacity: 1 }}
        exit={{ opacity: 0.3 }}
    >
        <h1>Записаться</h1>
        <form className="time" onSubmit={(e) =>
handleOnSubmit(e, setSubmitting)}>
            <SelectService />
            {
timeSlice.idService != null && <SelectEmployee />

```

```

    }
    {
        timeSlice.idEmployee != null &&
<RadiosDate />
    }
    {
        timeSlice.dateDay &&
timeSlice.dateMonth && timeSlice.dateYear && <RadiosTime />
    }
    {
        timeSlice.dateHours &&
timeSlice.dateMinutes &&
        <motion.div className={"client-info"}
            initial={{ opacity: 0 }}
            animate={{ opacity: 1 }}
            exit={{ opacity: 0 }}>
            <input className={"number form-
elem"} type={"tel"} value={phoneNumber}
                placeholder={"71234567890"}
                maxLength={11}
                onChange={ (e) =>
setPhoneNumber (e.currentTarget.value)} required={true} disabled
/>
            <input className={"name form-elem"}
type={"text"} value={clientName} placeholder={"Имя"}
                onChange={ (e) =>
setClientName (e.currentTarget.value)} required={true} disabled
/>
            </motion.div>
        }
        {
            timeSlice.dateHours &&
timeSlice.dateMinutes && (phoneNumber?.length === 11) &&
clientName &&
            <>
                <div className="info">
                    <span className={"info-
title"}>Выбранное время</span>
                    <span className={"info-
time"}>{formatNumber (timeSlice.dateDay)} . {formatNumber (timeSlic
e.dateMonth)} . {timeSlice.dateYear}
{timeSlice.dateHours} : {formatNumber (timeSlice.dateMinutes)} </sp
an>
                    <span className={"employee-
name__wrapper"}>к мастеру<span className={"employee-
name"}>{timeSlice.nameEmployee}</span></span>
                    <span className={"service-
name__wrapper"}>на услугу<span className={"service-
name"}>{timeSlice.nameService}</span></span>
                    <span className={"info-
price"}>{serviceInfo?.price} руб. {serviceInfo?.duration}
минут</span>
                </div>
            </>
        }
    }

```

```

<motion.button className="confirm"
                                type={"submit"}
                                id="timeConfirm"
                                disabled={submitting}
                                initial={{ opacity: 0 }}
                                animate={{ opacity: 1 }}
                                exit={{ opacity: 0 }}
                                >
                                Подтвердить запись
                                </motion.button>
                                </>
                                }
                                </form>
                                </motion.div>
                                </>
                                )
                                }

// Экспорт созданного компонента
export default PageTime

```

После успешно заполненной формы записи пользователю предлагается подтвердить действие о записи при помощи информации «Выбранное время» и кнопки «Подтвердить запись». Вид информации о записи представлен на рисунке 15.

79214889606 Кирилл

Выбранное время
 29.06.2024 10:00
 к мастеру
Астафьев Кирилл Александрович
 на услугу
Маникюр
 1450 руб. 130 минут

Подтвердить запись

Рисунок 15 – Lovely Arkhangelsk. Вид информации о записи на странице «Записаться»

3 Тестирование и отладка программного обеспечения

3.1 Функциональное тестирование

В процессе разработки программного обеспечения одним из ключевых этапов является тестирование, цель которого — выявление и устранение ошибок, обеспечивающих стабильную и корректную работу продукта.

Одним из наиболее эффективных методов тестирования является функциональное тестирование, выполняемое методом «черного ящика». Этот подход позволяет проверять работу системы на соответствие требованиям без необходимости анализа ее внутренней структуры и кода. В таблице 1 представлен набор тестов для страницы «Главная» при ситуации, когда пользователь не авторизован.

Таблица 1 – Набор тестов страницы «Главная»

Действие	Ожидаемый результат	Полученный результат
Нажатие кнопки «Главная» в шапке сайта	Нет действия	Совпал с ожидаемым
Нажатие кнопки «Мастера» в шапке сайта	Перенаправление пользователя на страницу с мастерами	Совпал с ожидаемым
Нажатие кнопки «Работы» в шапке сайта	Перенаправление пользователя на страницу с фотографиями завершённых услуг	Совпал с ожидаемым
Нажатие кнопки с логотипом салона красоты в шапке сайта	Нет действия	Совпал с ожидаемым
Нажатие кнопки «Услуги» в шапке сайта	Перенаправление пользователя на страницу с услугами	Совпал с ожидаемым

Продолжение таблицы 1

Действие	Ожидаемый результат	Полученный результат
Нажатие кнопки «Записаться» в шапке сайта	Перенаправление пользователя на страницу авторизации	Совпал с ожидаемым
Нажатие кнопки «Войти» в шапке сайта	Перенаправление пользователя на страницу авторизации	Совпал с ожидаемым
Нажатие кнопки «Наши мастера»	Перенаправление пользователя на страницу с мастерами	Совпал с ожидаемым
Нажатие на карточку мастера с подписью «Астафьев Кирилл Александрович»	Перенаправление пользователя на страницу мастера с именем «Астафьев Кирилл Александрович»	Совпал с ожидаемым
Наведение курсора на виджет карты, зажатие левой кнопки мыши и движение мыши по горизонтальным осям	Передвижение виджета карты	Совпал с ожидаемым
Наведение курсора на виджет карты, прокручивание колёсика мыши	Изменение масштаба виджета карты	Совпал с ожидаемым

4 Инструкция по эксплуатации программного обеспечения

4.1 Конфигурация программного обеспечения

Перед началом установки необходимо произвести конфигурацию модулей при помощи следующих компонентов:

- `package.json`, определяет зависимости и скрипты проекта,
- `Dockerfile`, описывает инструкции для создания Docker-образа,
- `docker-compose.yml`, конфигурирует многоконтейнерное Docker-приложение,
- `.env`, содержит переменные окружения,
- `nginx.conf`, настраивает веб-сервер Nginx,
- `tsconfig.json`, задает конфигурацию TypeScript.

Настройка файлов конфигурации перед развертыванием и эксплуатацией программного обеспечения имеет критическое значение для обеспечения его корректной работы и безопасности.

4.2 Установка программного обеспечения

Для развертывания серверной части понадобится установленный Docker. Для установки на ОС Linux Ubuntu требуется запустить терминал и выполнить команды «`sudo apt update`», а затем «`sudo apt install docker.io`».

После установки необходимого ПО можно приступить к контейнеризации. Для этого необходимо перейти в папку проекта «LovelyArkhangelsk», где расположен файл `docker-compose.yml`, фрагмент кода которого представлен листингом 5, и в терминале выполнить команду `docker-compose up`, которая в автоматическом режиме развернет все необходимое. После выполнения всех действий по развёртыванию ПО, веб-сайт будет доступен по заранее занятому адресу в сети Интернет.

Листинг 5 – Фрагмент кода файла «docker-compose.yml»

```
version: '3.9'

services:
  database:
    container_name: diplom_database
    image: mysql:8.0
    hostname: databasediplomapi
    ports:
      - "3306:3306"
    build:
      network: host
    environment:
      MYSQL_ROOT_PASSWORD: "root"
      MYSQL_DATABASE: "lovely_arkhangelsk"
    volumes:
      - ./SQL:/docker-entrypoint-initdb.d
    logging:
      driver: none
    restart: always

  react:
    container_name: diplom_react
    image: diplom_react
    hostname: reactdiplomapi
    build:
      network: host
      context: ./App
      dockerfile: Dockerfile
    ports:
      - "6000:6000"
    stdin_open: true
    logging:
      driver: none
    restart: always

  api:
    container_name: diplom_api
    image: diplom_api
    hostname: diplomapi
    build:
      network: host
      context: ./Api
      dockerfile: Dockerfile
    ports:
      - "5000:5000"
    depends_on:
      - database
    restart: always
```



```
volumes:
  db:

networks:
  backend:
    external: false
    driver: bridge
```

4.3 Руководство администратора

Для того чтобы начать пользоваться веб-сайтом, требуется ввести его адрес в адресной строке браузера. Перемещаться между страницами можно с помощью кнопок навигации, которые находятся либо в верхней части веб-сайта, либо в боковом меню при использовании мобильного устройства. Вид бокового меню навигации представлен рисунком 16.

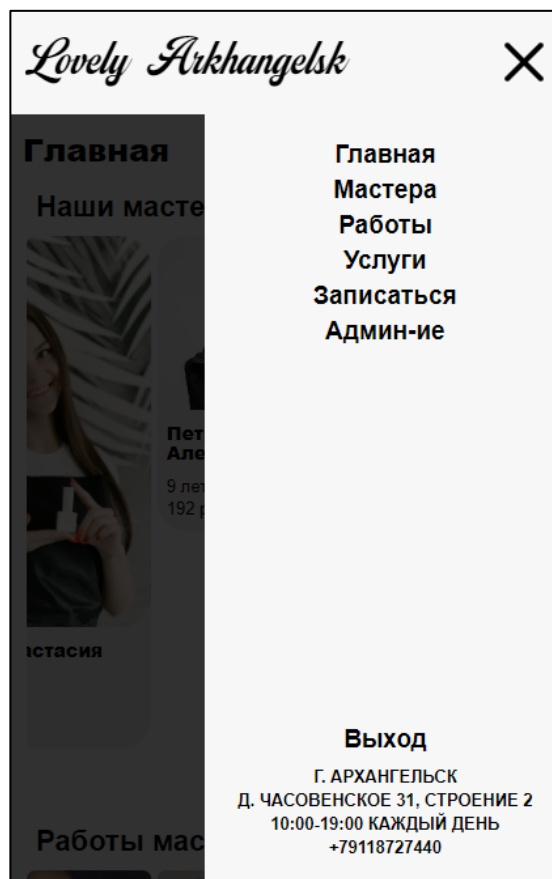


Рисунок 16 – Lovely Arkhangelsk. Вид бокового меню навигации

Переход на страницу «Панель администратора» доступен только авторизованным пользователям с ролью «Администратор». В список возможностей администратора входит следующее:

- добавление и редактирование информации об услугах,
- редактирование информации о мастерах,
- редактирование информации о клиентах и их ролях,
- просмотр статистики салона красоты,
- добавление и удаление фотографий работ мастеров.

При переходе на страницу «Панель администратора» пользователю предлагается боковое меню навигации по разделам страниц с левой стороны. Вид страницы «Панель администратора» представлен рисунком 17.

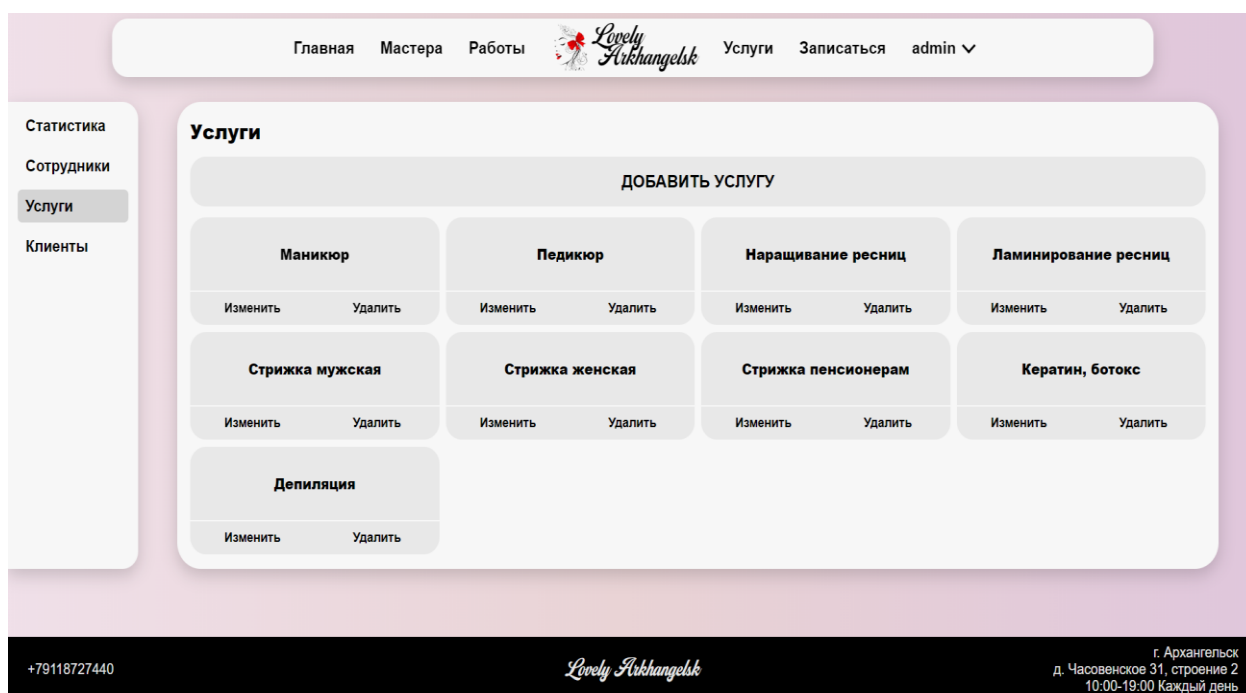


Рисунок 17 – Lovely Arkhangelsk. Вид страницы «Панель администратора»

В разделе «Сотрудники» администратору доступны следующие действие:

- удаление информации о мастере по нажатию кнопки «Удалить» у соответствующей карточки,

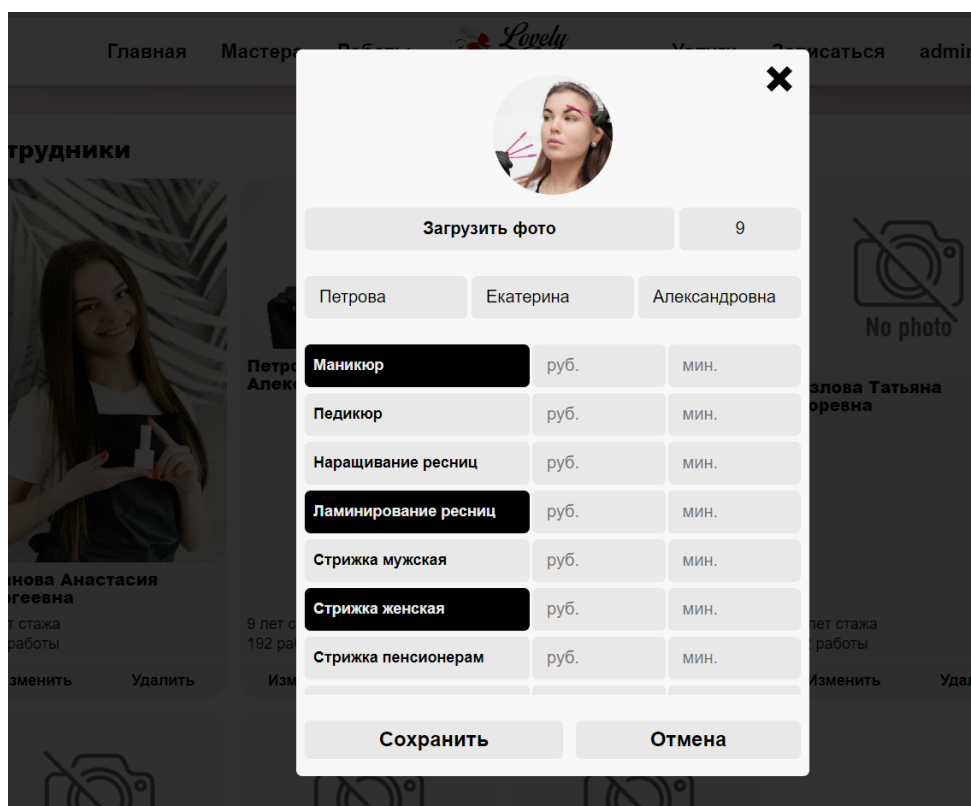
- редактирование информации о мастере по нажатию кнопки «Измени» у соответствующей карточки,

- переход в портфолио мастера по нажанию на соответствующую карточку.

При нажатии на кнопку «Изменить» происходит открытие модального окна изменения информации о мастере, где администратору предоставляется возможность изменить следующие данные:

- ФИО,
- фото мастера,
- стаж в годах,
- услуги, которые оказывает мастер,
- стоимость и длительность выбранных услуг специально у мастера.

Вид модального окна изменения информации о мастере представлен рисунком 18.



The modal window displays the following information:

- Profile picture: A woman with long brown hair.
- Buttons: 'Загрузить фото' (Upload photo) and '9' (likely indicating the number of photos).
- Fields for FIO: 'Петрова', 'Екатерина', 'Александровна'.
- Table of services:

Услуга	руб.	мин.
Маникюр		
Педикюр		
Наращивание ресниц		
Ламинирование ресниц		
Стрижка мужская		
Стрижка женская		
Стрижка пенсионерам		

At the bottom of the modal window are two buttons: 'Сохранить' (Save) and 'Отмена' (Cancel).

Рисунок 18 – Lovely Arkhangelsk. Вид модального окна изменения информации о мастере

Для сохранения измененных данных требуется нажать на кнопку «Сохранить». Для отмены внесенных данных и закрытия модального окна требуется нажать на кнопку «Отмена».

В разделе «Услуги» администратору доступны следующие действия:

- создание информации о новой услуге по нажатию соответствующей кнопки с подписью «Добавить услугу»,
- удаление информации об услуге по нажатию кнопки «Удалить» у соответствующей карточки,
- редактирование информации об услуге по нажатию кнопки «Изменить» у соответствующей карточки,
- переход в портфолио мастера при нажатии на соответствующую карточку.

При нажатии на кнопки «Добавить услугу» и «Изменить» появляются соответствующие модальные окна, где администратор может ввести следующие данные:

- название услуги,
- описание услуги (опционально),
- стоимость услуги,
- длительность услуги.

Вид модального окна добавление информации об услуге представлен рисунком 19.

Вид модального окна «Изменение услуги» представлен рисунком 20.

Для сохранения измененных данных требуется нажать на кнопку «Создать». Для отмены внесенных данных и закрытия модального окна требуется нажать на кнопку «Отмена».

В разделе «Клиенты» администратору доступны следующие действия:

- удаление информации о пользователе,
- изменение роли пользователя.

Главная Маникюр Д.С. Lovely Удалить admin

[Название услуги]

Название услуги

Описание (по наличию)

Длительность (минут) Стоимость

Создать Отмена

Рисунок 19 – Lovely Arkhangelsk. Вид модального окна добавления информации об услуге

Главная Услуги Маникюр Д.С. Lovely Удалить admin

Маникюр

Маникюр

Описание (по наличию)

60 1400

Создать Отмена

Рисунок 20 – Lovely Arkhangelsk. Вид модального окна «Изменение услуги»

Для удаления информации о пользователе требуется нажать на соответствующую кнопку с иконкой мусорного бака в конце строки с нужным пользователем.

Для изменения роли пользователя необходимо нажать на выпадающий список и выбрать соответствующую роль в строке с данными необходимого пользователя. Если роль пользователя была изменена на «Мастер», то у него создается собственное портфолио мастера. Если роль была изменена с «Мастер» на какую-то иную, портфолио мастера будет удалено без возможности сохранения данных.

5 Определение затрат на разработку веб-сайта

5.1 Техничко-экономическое обоснование

В текущем разделе дипломного проекта проводится экономическое обоснование разработки веб-сайта для салона красоты «Lovely Arkhangelsk».

Сумма затрат на создание сайта складывается из расходов по оплате машинного времени, расходов по оплате труда разработчика сайта и общих затрат $Z_{\text{спп}}$, руб.

Затраты на создание сайта $Z_{\text{спп}}$, руб., определяются по формуле

$$Z_{\text{спп}} = Z_{\text{спп}}^{\text{МВ}} + Z_{\text{общ}} , \quad (1)$$

где $Z_{\text{спп}}^{\text{МВ}}$ – затраты на оплату машинного времени, руб.;

$Z_{\text{общ}}$ – общие затраты, руб.

Трудоёмкость разработки сайта t , чел. ч, определяется по формуле

$$t = t_o + t_{\text{и}} + t_a + t_{\text{п}} + t_{\text{от}} + t_{\text{д}} , \quad (2)$$

где t_o – затраты труда на подготовку описания задачи, чел.ч;

$t_{\text{и}}$ – затраты труда на исследование алгоритма решения задачи, чел.ч;

t_a – затраты труда на разработку алгоритма, чел.ч;

$t_{\text{п}}$ – затраты труда на программирование по готовому макету, чел.ч;

$t_{\text{от}}$ – затраты труда на отладку сайта ЭВМ, чел.ч;

$t_{\text{д}}$ – затраты труда на подготовку документации, чел.ч.

Определение затрат вычисляется при помощи условного числа операторов. Условное число операторов Q , ед, на сайте определяется по формуле

$$Q = q \cdot c \cdot (1+p), \quad (3)$$

где q – число операторов (исходных команд), ед;

c – коэффициент, учитывающий новизну и сложность сайта;

p – коэффициент коррекции сайта в ходе разработки, зависит от точности и корректности поставленной задачи (0.05-0.1).

В разработанном сайте число операторов составляет около 486 ($q=486$).

Коэффициент, учитывающий новизну и сложность сайта, определяется исходя из таблицы 2 на пересечении групп сложности и степени новизны.

Таблица 2 – Значение коэффициента c и B

Язык программирования	Группа сложности	Степень новизны				Коэффициент B
		А	Б	В	Г	
Высокого уровня	1	1,38	1,26	1,15	1,20	1,20
	2	1,30	1,19	1,08	0,65	1,35
	3	1,20	1,10	1,00	0,60	1,50
Низкого уровня	1	1,58	1,45	1,32	0,79	1,20
	2	1,49	1,37	1,24	0,74	1,35
	3	1,38	1,26	1,15	0,69	1,50

Сайт по степени новизны относится к одной из четырёх групп:

- 1) группа А – разработка принципиально новых задач,
- 2) группа Б – разработка оригинальных программ,
- 3) группа В – разработка программ с использованием типовых решений,
- 4) группа Г – разовая типовая задача.

По степени сложности сайт относится к одной из трёх групп:

- 1) алгоритмы оптимизации и моделирования систем,
- 2) задачи учёта, отчётности и статистики,
- 3) стандартные алгоритмы.

Созданный сайт по степени новизны относится к разработке программ с использованием типовых решений (группа В), а по степени сложности алгоритма – к стандартным алгоритмам (группа 3).

По таблице 3 коэффициент $c = 1$ и коэффициент $B = 1,5$.

С учётом того, что задача поставлена достаточно чётко, коэффициент p принимается равным 0,05 ($p=0,05$).

Условное число операторов, согласно формуле (3), составляет

$$Q = 486 \cdot 1 \cdot (1,00 + 0,05) = 510,3 \text{ ед}$$

Затраты труда на подготовку описания задачи t_o , чел. ч, точно определить невозможно, т.к. это связано с творческим характером работы. С учетом этого можно принять данное значение равным 30 чел. ч ($t_o=30$).

Таблица 3 – Коэффициент квалификации разработчика (К)

Опыт работы	Коэффициент квалификации
До двух лет	0,80
2-3 года	1,00
3-5 лет	1,10 – 1,20
5-7 лет	1,30 – 1,40
Более 7 лет	1,50 – 1,60

Затраты труда на изучение описания задачи с учётом уточнения описания и квалификации программиста t_u , чел. ч., определяются по формуле

$$t_u = Q \cdot \frac{B}{(75 \dots 85 \cdot K)}, \quad (4)$$

где B – коэффициент увеличения затрат труда вследствие недостаточного описания задачи, уточнений и некоторой недоработки;

K – коэффициент квалификации работника.

По таблице 3 для работающих до двух лет $K = 0,8$.

$$T_{\text{и}} = 510,3 \cdot \frac{1,5}{80 \cdot 0,8} = 11,97 \text{ чел.ч}$$

Далее необходимо вычислить затраты труда на различных стадиях разработки сайта.

Затраты труда на разработку алгоритма решения задачи t_a , чел.ч, определяются по формуле

$$t_a = \frac{Q}{50 \dots 75 \cdot K}, \quad (5)$$

Исходя из формулы расчета затрат на разработку алгоритма решения задачи, получено следующие значение

$$t_a = \frac{510,3}{65 \cdot 0,8} = 9,81 \text{ чел.ч}$$

Затраты труда на составление сайта по готовому макету $t_{\text{п}}$, чел.ч, определяются по формуле

$$t_{\text{п}} = \frac{Q}{50 \dots 75 \cdot K}, \quad (6)$$

Получение размера затрат труда на создание сайта по готовому макету составило

$$t_{\pi} = \frac{510,3}{50 \cdot 0,8} = 12,75 \text{ чел.ч}$$

Затраты труда на отладку сайта на ЭВМ при комплексной отладке $t_{от}$, чел.ч., определяются по формуле

$$t_{от} = 1,5 \cdot t_{от}^A, \quad (7)$$

где $t_{от}^A$ – затраты труда на отладку сайта на ЭВМ при автономной отладке одной задачи, чел.ч.

Затраты труда на отладку сайта на ЭВМ при автономной отладке одной задачи $t_{от}^A$, чел.ч, определяются по формуле

$$t_{от}^A = \frac{Q}{50 \dots 75 \cdot K}, \quad (8)$$

По формуле (8), можно получить размер затрат на отладку сайта

$$t_{от}^A = \frac{510,3}{65 \cdot 0,8} = 9,81 \text{ чел.ч}$$

Далее требуется рассчитать затраты труда на отладку сайта на ЭВМ при комплексной отладке по формуле (7)

$$t_{от} = 1,5 \cdot 9,81 = 14,72 \text{ чел.ч}$$

Затраты труда на подготовку документации по задаче t_d , чел.ч, определяются по формуле

$$t_d = t_{др} + t_{до}, \quad (9)$$

где $t_{др}$ – затраты труда на подготовку материалов рукописи, чел.ч;

$t_{до}$ – затраты на редактирование, печать и оформление документации, чел.ч.

Затраты труда на подготовку материалов рукописи $t_{др}$, чел.ч, определяются по формуле

$$t_{др} = \frac{Q}{150...200 \cdot K} \quad (10)$$

Тогда по формуле (10) можно получить размер затрат труда на подготовку материалов рукописи

$$t_{др} = \frac{510,3}{175 \cdot 0,8} = 3,56 \text{ чел.ч}$$

Затраты на редактирование, печать и оформление документации $t_{до}$, чел.ч, определяются по формуле

$$t_{до} = 0,75 \cdot t_{др} \quad (11)$$

Тогда по формуле (11) можно получить размер затрат на редактирование, печать и оформление документации

$$t_{до} = 0,75 \cdot 3,56 = 2,67 \text{ чел.ч}$$

Далее требуется рассчитать затраты труда на подготовку документации по задаче по формуле (10)

$$t_{д} = 3,56 + 2,67 = 6,23 \text{ чел.ч}$$

Трудоёмкость разработки сайта, согласно формуле (2), составляет

$$t = 30 + 11,97 + 9,81 + 12,75 + 14,72 + 6,23 = 85,48 \text{ чел. ч.}$$

При восьмичасовом рабочем дне это составляет одиннадцать рабочих дней.

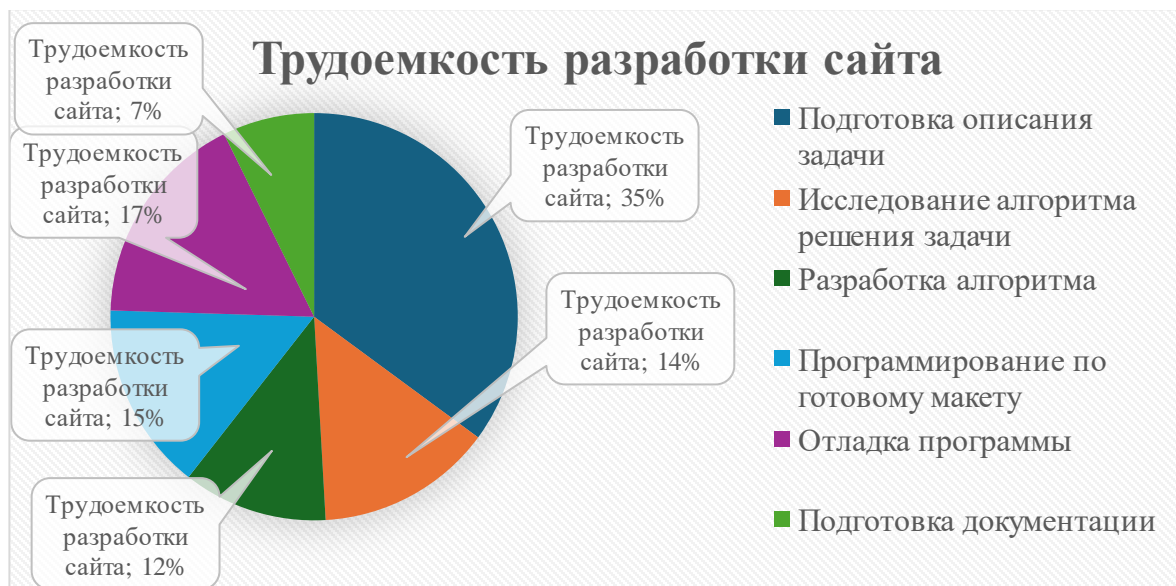


Рисунок 21 – Структура трудозатрат проекта. Диаграмма круговая

Среди затраченного на разработку проекта времени больше всего ушло на подготовку описания задачи в силу отсутствия четкого плана разработки и опыта.

Затраты на оплату машинного времени при отладке сайта $З_{спп}^{MB}$, руб., определяются по формуле

$$З_{спп}^{MB} = C_{час} \cdot t_{ЭВМ}, \quad (12)$$

где $C_{час}$ – цена машино-часа арендного времени, руб/ч;

$t_{ЭВМ}$ – фактическое время отладки сайта на ЭВМ, чел.ч.

Фактическое время отладки $t_{ЭВМ}$, чел.ч, определяется по формуле:

$$t_{ЭВМ} = t_{п} + t_{д} + t_{от} , \quad (13)$$

По формуле (13) фактическое время отладки равняется

$$t_{ЭВМ} = 12,75 + 6,23 + 14,72 = 33,70 \text{ чел.ч}$$

Цена машино-часа $C_{час}$, руб/ч, определяется по формуле

$$C_{час} = \frac{3_{ЭВМ}}{T_{ЭВМ}} , \quad (14)$$

где $T_{ЭВМ}$ – действительный месячный фонд времени ЭВМ, ч.

Действительный месячный фонд времени ЭВМ $T_{ЭВМ}$, ч, определяется по формуле

$$T_{ЭВМ} = 8 \cdot (K_{д} - K_{пв}) - t_{пр} \cdot 4 , \quad (15)$$

где $K_{д}$ – общее количество дней в месяце;

$K_{пв}$ – количество праздничных и выходных дней в месяце;

$t_{пр}$ – время простоя в профилактических работах, ч.

В период с 19.04.2024 по 16.05.2024 общее количество календарных дней $K_{д} = 28$, число праздничных и выходных дней $K_{пв} = 12$ в соответствии со статьей 112 трудового кодекса РФ [4] и Постановлением Правительства РФ от 10.08.2023 N 1314 «О переносе выходных дней в 2024 году».

Время простоя в профилактических работах определяется как еженедельная профилактика по 4 часа.

По формуле (15) действительный месячный фонд времени равняется

$$T_{ЭВМ} = 8 \cdot (28-12) - 4 \cdot 4 = 112 \text{ ч}$$

Затраты на эксплуатацию ЭВМ $Z_{общ}$, руб., определяются по формуле

$$Z_{ЭВМ} = Z_{ам} + Z_{эл}, \quad (16)$$

где $Z_{ам}$ – издержки на амортизацию, руб.;

$Z_{эл}$ – издержки на электроэнергию, потребляемую ЭВМ, руб.;

Компьютер, на котором выполнена разработка сайта, приобретён по рыночной цене 44500,00 руб. Согласно НК РФ Ч. 2 ст. 256 и ст. 257 амортизируемым имуществом признается имущество со сроком полезного использования более двенадцати месяцев и первоначальной стоимостью более ста тысяч рублей [1]. Так как имущество, используемое для разработки программного продукта, имеет первоначальную стоимость менее ста тысяч рублей, то затраты на амортизацию учитывать не требуется, то есть $Z_{ам} = 0$ руб.

Стоимость электроэнергии, потребляемой за месяц, $Z_{эл}$, руб., определяется по формуле

$$Z_{эл} = P_{ЭВМ} \cdot T_{ЭВМ} \cdot C_{эл}, \quad (17)$$

где $P_{ЭВМ}$ – суммарная мощность ЭВМ, кВт;

$C_{эл}$ – стоимость 1 кВт·ч электроэнергии, руб.

Согласно техническому паспорту ЭВМ, потребление электроэнергии составляет 0,4 кВт·ч.

Согласно Постановлению Агентства по тарифам и ценам Архангельской области от 20.12.2023 тариф на электроэнергию составляет 6,06 руб. за один кВт × ч [5].

С учетом этого стоимость электроэнергии, потребляемой за месяц, по формуле (17)

$$З_{эл} = 0,4 \cdot 112 \cdot 6,06 = 271,49 \text{ руб.}$$

Затраты на эксплуатацию ЭВМ, согласно формуле (16), составляют

$$З_{ЭВМ} = 0 + 271,49 = 271,49 \text{ руб.}$$

Цена машино-часа, согласно формуле (14), составляет

$$C_{\text{час}} = 271,49 / 112 = 2,42 \text{ руб/ч}$$

Затраты на оплату машинного времени при отладке сайта, согласно формуле (12), составляют

$$З_{\text{смп}}^{\text{МВ}} = 2,42 \cdot 33,70 = 81,55 \text{ руб.}$$

Общие затраты $З_{\text{общ}}$, руб., определяются по формуле

$$З_{\text{общ}} = \Phi_{\text{от}} + З_{\text{страх}} + З_{\text{ам}} + З_{\text{эл}} + З_{\text{пр}}, \quad (18)$$

где $З_{\text{зп}}$ – издержки на заработную плату, руб.;

$З_{\text{страх}}$ – издержки на страховые взносы с оплаты труда, руб.;

$З_{\text{пр}}$ – издержки на прочие и накладные расходы, руб.

Заработная плата работников $З_{\text{зп}}$, руб., определяется по формуле

$$З_{\text{зп}} = \text{оклад} \cdot K_{\text{сев}} \cdot K_{\text{р-н}}, \quad (19)$$

где $K_{\text{сев}}$ – коэффициент, учитывающий северную надбавку для работающих в местах, приравненных к Крайнему Северу;

K_{p-n} – коэффициент, учитывающий районную надбавку на территориях, приравненных к Крайнему Северу.

Для расчета заработной платы оклад программиста принимаем равным величине МРОТ, который согласно ст. 1 Федерального закона от 19.06.2000 № 82-ФЗ «О минимальном размере оплаты труда» в редакции Федерального закона от 27.11.2023 № 548-ФЗ [3], составляет 19 242,00 руб. в месяц без учета районных коэффициентов и надбавок.

Так как разработка проекта ведется на территории с особыми климатическими условиями – г. Архангельск, то на основании статьи 317 ТК РФ [4] и ст. 11 закона РФ от 19.02.1993 г. № 4520-1 [2] к окладу применяются районные коэффициенты и надбавки: районный коэффициент составляет 20% от основной заработной платы, а выплаты за выслугу лет, проработанных на территории, приравненной к территории Крайнего Севера – 50% от основной заработной платы.

Заработная плата работников по формуле (19) составляет

$$З_{зп} = 19242,00 \cdot (1 + 0,5 + 0,7) = 32711,40 \text{ руб.}$$

Страховые взносы с оплаты труда $З_{страх}$, руб., определяются по формуле

$$З_{страх} = \frac{\Phi_{от} \cdot H_{страх}}{100}, \quad (20)$$

где $\Phi_{от}$ – фонд оплаты труда, руб.;

$H_{страх}$ – размер страховых взносов, %.

Фонд оплаты труда $\Phi_{от}$ определяется по формуле

$$\Phi_{от} = З_{зп} / (D_{общ} - D_{вых}) \cdot D_{раб}, \quad (21)$$

где $З_{зп}$ – заработная плата работников, руб.;

$D_{\text{общ}}$ – общее количество рабочих дней, дн;

$D_{\text{вых}}$ – количество выходных и праздничных дней, дн;

$D_{\text{раб}}$ – количество фактических отработанных рабочих дней, дн.

Фонд оплаты труда $\Phi_{\text{от}}$ по формуле (21) составляет

$$\Phi_{\text{от}} = 32711,40 : (28 - 12) \cdot 11 = 22489,09 \text{ руб.}$$

Издержки на страховые взносы ($H_{\text{страх}}$) с оплаты труда определяются гл. 34 ст. 425 НК РФ как сумма взносов в ПФР – 22%, ФОМС – 5,1% и ФСС – 2,9%. С 01.01.2023 года страховые взносы перечисляются в налоговую единым платежным поручением общей суммой, поэтому размер страховых взносов с заработной платы можно принять равным 30% без учета взносов на травматизм, так как эта сумма зависит от опасности производства. Страховые взносы на страхование от несчастных случаев на производстве и профзаболеваний не учитываются.

Страховые взносы с оплаты труда по формуле (20) составляют

$$З_{\text{страх}} = \frac{22489,09 \cdot 30}{100} = 6746,73 \text{ руб.}$$

Прочие затраты $З_{\text{пр}}$, руб., принимаются в размере 10% в общей сумме затрат и определяются по формуле

$$З_{\text{пр}} = (\Phi_{\text{от}} + З_{\text{страх}} + З_{\text{ам}} + З_{\text{эл}}) \cdot 10/90 \quad (22)$$

По формуле (22) прочие затраты составляют

$$З_{\text{пр}} = \frac{(22489,09 + 6746,73 + 0 + 271,49) \cdot 10}{90} = 3278,59 \text{ руб.}$$

Общие затраты, согласно формуле (18), составляют

$$З_{\text{общ}} = 22489,09 + 6746,73 + 0 + 271,49 + 3278,59 = 32785,90 \text{ руб.}$$

Затраты на создание сайта, согласно формуле (1), составляют

$$З_{\text{смп}} = 81,55 + 32785,90 = 32867,45 \text{ руб.}$$

На основе затрат на создание сайта построена диаграмма затрат (рисунок 14).

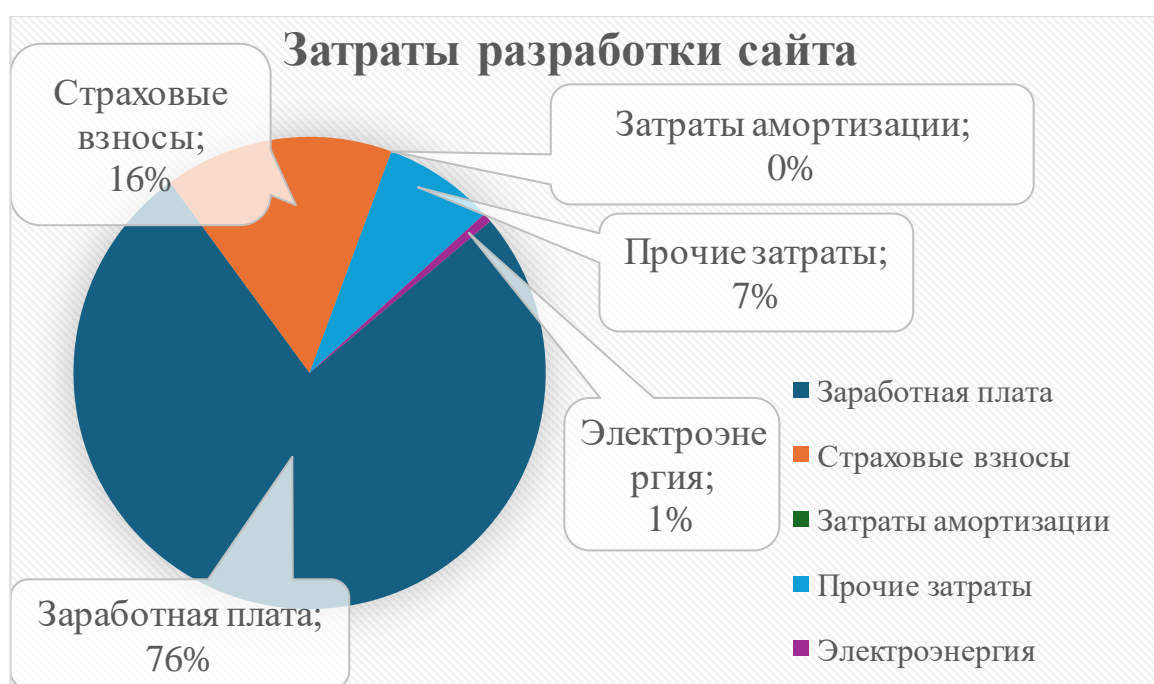


Рисунок 22 – Диаграмма затрат сайта. Диаграмма круговая

В итоге при разработке сайта основные затраты пришлись на заработную плату разработчика, так как большая часть процесса разработки составляет личный труд разработчика.

В конечном итоге трудоёмкость создания сайта составила 85,48 чел.ч. В результате выполненных расчётов затраты на создание сайта составляют 32867,45 руб. Средняя стоимость создания сайта варьируется от 20000,00 руб.,

до 50000,00 руб. в зависимости от сложности задачи, объёма работ и других факторов, влияющих на разработку сайта.

5.2 Анализ экономического обоснования и рисков проекта

Риски в проекте – это потенциальные события или условия, которые могут повлечь за собой неблагоприятные последствия для процесса разработки, его участников и заказчика. Для эффективного управления проектом необходимо определить риски и определить мероприятия по их снижению.

В таблице 4 представлены риски разработки сайта.

Таблица 4 – Риски проекта в процессе разработки

Риски	Мероприятия по снижению риска
Нечеткая постановка задачи	Разбивание задач на более мелкие, выполнимые этапы
Ошибочный выбор технологий	Тщательный анализ технологий и требований к разработке проекта
Низкая производительность веб-сайта	Оптимизация кода продукта
Неэффективное продвижение веб-сайта	Проработка плана продвижения веб-сайта

В ходе выполнения дипломного проекта по разработке веб-сайта для салона красоты «Lovely Arkhangelsk» было выявлено несколько ключевых экономических преимуществ, которые значительно улучшают эффективность работы салона и снижают его операционные издержки.

Одним из основных экономических преимуществ использования разработанного веб-сайта является сокращение необходимости в административном персонале. С введением автоматизированной системы бронирования на веб-сайте, эти задачи обработки запросов выполняются

программно, что позволяет сократить или полностью исключить расходы на оплату труда администратора.

Внедрение системы онлайн-бронирования позволяет мастерам больше не заниматься ручной записью клиентов, что освобождает значительное количество времени, которое они могут посвятить непосредственному оказанию услуг. Автоматизация процесса записи клиентов исключает ошибки, связанные с человеческим фактором, такие как двойное бронирование или потеря записей, что также повышает эффективность работы.

Современный и функциональный веб-сайт способствует улучшению имиджа салона и служит эффективным инструментом маркетинга. Интеграция с социальными сетями и возможность онлайн-обратной связи позволяют более эффективно привлекать новых клиентов, снижая затраты на традиционные методы рекламы.

6 Охрана труда и техника безопасности при работе на ПК

6.1 Общие требования охраны труда

К самостоятельной работе на ПК допускаются лица, прошедшие предварительные и периодические медицинские осмотры, вводный инструктаж по охране труда, обучение безопасным приемам работы на ПК и первичный инструктаж на конкретном рабочем месте.

При работе на ПК работнику необходимо учитывать возможное воздействие следующих вредных и опасных производственных факторов: повышенная температура поверхностей ПК, повышенное напряжение в электрической цепи, уровень статического электричества, уровень электромагнитных излучений, напряженность электрического поля, недостаточная освещенность рабочей зоны, повышенная яркость света, повышенная контрастность, зрительное напряжение, длительные статические нагрузки и монотонность трудового процесса.

При работе на ПК работник обязан соблюдать производственную и технологическую дисциплину, регулировать режим труда и отдыха, выполнять задачи по должностной инструкции, поддерживать порядок на рабочем месте, сообщать о неисправностях ПК и электропитания руководителю и соблюдать требования пожарной и электробезопасности.

За нарушение требований охраны труда работник привлекается к ответственности в порядке, установленном законодательством РФ.

6.2 Требования охраны труда перед началом работы

Перед началом работы на ПК пользователь должен проветрить помещение, отрегулировать освещенность на рабочем месте, проверить правильность расположения компьютера для исключения неудобных поз и

напряжений тела, убрать посторонние предметы с рабочего места и освободить подходы к нему.

Необходимо проверить исправность кабельных соединений, проверить отсутствие пыли на экране монитора и клавиатуре, протереть их при необходимости специальной салфеткой.

6.3 Требования охраны труда во время работы

Подключение ПК к сети электропитания выполнять только имеющимися штатными сетевыми кабелями при закрытых кожухах и наличии заземления.

При включении ПК необходимо соблюдать следующую последовательность действий: сначала включить сетевой фильтр, затем источник бесперебойного питания, системный блок и монитор. После включения ПК и запуска программы, убедиться в отсутствии дрожания и мерцания изображения на экране монитора, настроить яркость, контрастность, цвет и размер символов, и фон экрана для наиболее комфортного и четкого восприятия изображения.

Во время работы на ПК работник должен избегать натягивания, скручивания, перегиба и пережима шнуров электропитания, предотвращать попадание предметов на шнуры, избегать влаги на поверхности ПК, не прикасаться к задней панели системного блока при включенном питании, не оставлять включенный ПК без наблюдения, не загромождать рабочее место, не проводить самостоятельный ремонт ПК, не закрывать вентиляционные отверстия, не часто отключать и включать ПК, соблюдать расстояние от глаз до экрана (50–70 см), не превышать двух часов непрерывной работы без перерыва.

ЗАКЛЮЧЕНИЕ

В ходе выполнения дипломного проекта успешно разработан веб-сайт для салона красоты «Lovely Arkhangelsk». Основная цель проекта заключалась в создании функционального и эстетически привлекательного веб-сайта, который способствует привлечению новых клиентов, повышению уровня обслуживания существующих клиентов и улучшению общей репутации салона. В процессе разработки использовались современные веб-технологии и инструменты, такие как HTML, CSS, TypeScript, а также фреймворк React. Для создания серверной части сайта и реализации базы данных применены Node.js и MySQL. Server

В ходе дипломного проектирования решены следующие задачи:

- выделены объекты и атрибуты БД в соответствии с заданием,
- спроектирована БД,
- разработана БД,
- спроектирован интерфейс пользователя,
- реализован интерфейс пользователя,
- разработаны программные модули веб-сайта «Lovely Arkhangelsk»,
- выполнено тестирование и отладка разработанного ПО,
- разработана инструкция по эксплуатации ПО,
- описан процесс установки разработанного ПО.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Российская Федерация. Законы. Налоговый Кодекс Российской Федерации. Часть вторая : Федеральный закон № 117-ФЗ : [принят Государственной думой 19 июля 2000 года : одобрен Советом Федерации 26 июля 2000 года]. – Текст : электронный // КонсультантПлюс : [сайт]. – 2024. – URL: https://www.consultant.ru/document/cons_doc_LAW_28165/ (дата обращения: 19.05.2024).

2. Российская Федерация. Законы. О государственных гарантиях и компенсациях для лиц, работающих и проживающих в районах Крайнего Севера и приравненных к ним местностях : Федеральный закон № 4520-1 : [принят и одобрен Президентом Российской Федерации Б. Ельциным 19 февраля 1993 года]. – Текст : электронный // КонсультантПлюс : [сайт]. – 2024. – URL: https://www.consultant.ru/document/cons_doc_LAW_1786/ (дата обращения: 19.05.2024).

3. Российская Федерация. Законы. О минимальном размере оплаты труда : Федеральный закон № 82-ФЗ : [принят Государственной Думой 2 июня 2000 года : одобрен Советом Федерации 7 июня 2000 года]. – Текст : электронный // КонсультантПлюс : [сайт]. – 2024. – URL: https://www.consultant.ru/document/cons_doc_LAW_27572/ (дата обращения: 19.05.2024).

4. Российская Федерация. Законы. Трудовой кодекс Российской Федерации : Федеральный закон № 197-ФЗ : [принят Государственной Думой 21 декабря 2001 года : одобрен Советом Федерации 26 декабря 2001 года]. – Текст : электронный // КонсультантПлюс : [сайт]. – 2024. – URL: https://www.consultant.ru/document/cons_doc_LAW_34683/ (дата обращения: 19.05.2024).

5. Официальное опубликование правовых актов : региональный цифровой ресурс : Постановление № 81 – э / 5 Об установлении цен (тарифов) на электрическую энергию : сайт / Правительства Архангельской Области. –

Архангельск, 2023 – 2024. – URL:
<http://publication.pravo.gov.ru/document/2901202312250052> (дата обращения:
19.05.2024). – Текст: электронный.

6. CSS documentation — DevDocs. – Текст : электронный. // MDN contributors: [сайт]. – 2005-2024. – URL: <https://devdocs.io/css/> (дата обращения: 26.05.2024).

7. draw.io Documentation. – Текст : электронный // JGraph Ltd: [сайт]. – 2005-2023. – URL: <https://www.drawio.com/doc/> (дата обращения: 26.05.2024).

8. Explore PhpStorm features | PhpStorm Documentation. – Текст : электронный // JetBrains s.r.o: [сайт]. – 2000-2024. – URL: <https://www.jetbrains.com/help/phpstorm/quick-start-guide-phpstorm.html> (дата обращения: 26.05.2024).

9. Getting Started – React. – Текст: электронный // Meta Platforms: [сайт] – 2024. – URL: <https://legacy.reactjs.org/docs/getting-started.html> (дата обращения: 26.05.2024).

10. Getting Started | Sequelize. – Текст : электронный // Sequelize Contributors: [сайт]. – 2024. – URL: <https://sequelize.org/docs/v7/getting-started/> (дата обращения: 26.05.2024).

11. Getting Started with Redux | Redux. – Текст : электронный // Dan Abramov and the Redux documentation authors: [сайт]. – 2015-2024. – URL: <https://redux.js.org/introduction/getting-started> (дата обращения: 19.05.2024).

12. HTML documentation – DevDocs. – Текст : электронный. // MDN contributors: [сайт]. – 2005-2024. – URL: <https://devdocs.io/html/> (дата обращения: 20.05.2024).

13. Installing Express. – Текст : электронный // StrongLoop: [сайт] – 2017. – URL: <https://expressjs.com/en/starter/installing.html/> (дата обращения: 23.05.2024).

14. JavaScript documentation – DevDocs. – Текст : электронный. // MDN contributors: [сайт]. – 2005-2024. – URL: <https://devdocs.io/javascript/> (дата обращения: 30.05.2024).

15. MySQL :: MySQL Workbench Manual. – Текст : электронный // Oracle: [сайт]. – 2024. – URL: <https://dev.mysql.com/doc/workbench/en/> (дата обращения: 26.05.2024).

16. Node.js v20.3.1 Documentation. – Текст : электронный // OpenJS Foundation: [сайт]. – 2009-2024. – URL: <https://nodejs.org/docs/latest-v20.x/> (дата обращения: 21.05.2024).

17. Postman documentation overview | Postman Learning Center. – Текст : электронный // Postman, Inc: [сайт]. – 2024. – URL: <https://learning.postman.com/docs/introduction/overview/> (дата обращения: 22.05.2024).

18. RTK Query Overview | Redux Toolkit. – Текст : электронный // Dan Abramov and the Redux documentation authors: [сайт] – 2015-2024. – URL: <https://redux-toolkit.js.org/rtk-query/overview> (дата обращения: 24.05.2024).

19. Sass: Documentation. – Текст : электронный // Sass: [сайт] – 2006-2024. – URL: <https://sass-lang.com/documentation/> (дата обращения: 29.05.2024).

20. The Importance of a Website for Your Business Success. – Текст : электронный // LYFE MARKETING: [сайт] – 2024. – URL: <https://www.lyfemarketing.com/blog/importance-of-a-website/> (дата обращения: 22.05.2024).

21. TypeScript: The starting point for learning TypeScript. – Текст : электронный // Microsoft: [сайт] – 2012-2024. – URL: <https://www.typescriptlang.org/docs/> (дата обращения: 28.05.2024).