Министерство науки и высшего образования РФ

Федеральное государственное автономное

образовательное учреждение

высшего образования

«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

ИНСТИТУТ УПРАВЛЕНИЯ БИЗНЕС-ПРОЦЕССАМИ

КАФЕДРА БИЗНЕС-ИНФОРМАТИКИ И МОДЕЛИРОВАНИЯ БИЗНЕС-ПРОЦЕССОВ

**КУРСОВАЯ РАБОТА**

Создание web-приложения с применением современных web-технологий

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.И. Никитин

подпись, дата

Студент УБ22-09Б, 432216993 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.А. Никифоров

подпись, дата

Красноярск 2024

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 2](#_Toc156398189)

[1 Теоретическая часть 3](#_Toc156398190)

[1.1 HTML 3](#_Toc156398191)

[1.2 CSS 4](#_Toc156398192)

[1.3 JavaScript 6](#_Toc156398193)

[1.4 PHP 7](#_Toc156398194)

[1.5 MySQL 8](#_Toc156398195)

[2 Практическая часть 10](#_Toc156398196)

[2.1 Средства разработки 10](#_Toc156398197)

[2.2 Проектирование модели данных 10](#_Toc156398198)

[2.3 Проектирование базы данных 11](#_Toc156398199)

[2.4 Описание основных программных модулей 12](#_Toc156398200)

[2.5 Основные страницы web-приложения 16](#_Toc156398201)

[ЗАКЛЮЧЕНИЕ 18](#_Toc156398202)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 19](#_Toc156398203)

# **ВВЕДЕНИЕ**

В наше время открывается все больше и больше предприятий различного масштаба, но взаимодействие между клиентами не всегда бывает простой. Не все клиенты предпочитают звонки или общение с предприятием на прямую. В связи с этим, создание сайта с удобным интерфейсом является необходимостью для предприятия.

Цель данной работы – разработка удобного web-приложения, которое будет удобное и простое в использовании со стороны предприятия и со стороны клиента.

В ходе данной работы будут использованы несколько представителейй web-технологий, такие как: HTML, CSS, PHP, JavaScript и MySQL. Представленные решения современные и перспективные, которые могут обеспечить комплексное решение практически любых задач при разработке динамичного веб-сайта любого масштаба и уровня сложности.

В конце данной работы будет представлено полноценное приложение, которое будет удовлетворять большинство потребностей со стороны предприятия и клиента. Также приложение будет готово к потенциальному увеличению функционала, смотря как предпочтёт предприятие.

Данный отчет будет состоять из подробных описаний и этапов разработке web-приложения, а также конечный продукт и возможные варианты развития данного приложения.

# **1 Теоретическая часть**

# **HTML**

HTML (Hypertext Markup Language) это код который используется для структурирования и разметки содержимого веб-страниц, и без него не было бы возможно создать удобные для пользователя интерфейсы, адаптированные под различные устройства.

В настоящее время доминирующей версией является HTML5, которая предоставляет разработчикам множество новых возможностей для создания современных и интерактивных веб-приложений. HTML5 добавляет новые элементы, атрибуты и API, которые значительно расширяют функциональность веб-страниц. Таким образом, HTML постоянно развивается и адаптируется к новым требованиям и возможностям.

Основными элементами HTML являются теги, которые используются для определения различных частей веб-страницы, таких как заголовки, параграфы, изображения, ссылки и другие. Теги представляют собой открывающие и закрывающие метки, между которыми находится содержимое элемента. Например, тег <p> используется для обозначения параграфа, и он открывается <p> и закрывается </p>. Теги в HTML обладают различными атрибутами, которые определяют дополнительные свойства элементов. Например, атрибут href определяет ссылку в теге <a> (гиперссылке), а атрибут src определяет путь к изображению в теге <img>. Атрибуты позволяют дополнительно настраивать элементы и делать их более интерактивными и информативными для пользователя.

Плюсами HTML являются его простота и доступность. HTML достаточно легок для изучения и использования даже для начинающих разработчиков. Кроме того, HTML является основным языком веб-разработки, и без него невозможно создать полноценную веб-страницу. Его стандартизированная структура и поддержка со стороны всех основных веб-браузеров обеспечивают надежность и совместимость веб-страниц. Также с развитием веб-технологий и повышением требований к веб-приложениям, HTML постоянно совершенствуется, и в настоящее время активно обсуждается его будущее. Новые спецификации и стандарты, такие как Web Components и WebAssembly, представляют собой попытки расширить возможности HTML и усовершенствовать веб-разработку. Web Components позволяет создавать собственные пользовательские элементы с поддержкой инкапсуляции и повторного использования, а WebAssembly добавляет возможность запуска высокопроизводительных приложений, написанных на других языках программирования, в веб-браузере.

Однако у HTML также есть недостатки. Например, HTML является статичным языком разметки, который не предоставляет мощных возможностей для динамического взаимодействия с пользователем. Для реализации сложной интерактивности и анимаций часто требуется использовать JavaScript, CSS и другие технологии, что может усложнять разработку веб-приложений.

## **CSS**

CSS (Cascading Style Sheets) — это язык разметки, используемый для описания визуального представления веб-страниц. Основная функция CSS состоит в том, чтобы определить стиль, внешний вид и расположение элементов веб-страницы.

Основные функции CSS:

1. Оформление элементов: CSS позволяет задавать стиль для элементов HTML, таких как текст, ссылки, таблицы, формы и изображения. Путем задания свойств, таких как цвет, шрифт, размер текста, отступы и рамки, разработчики могут создавать разнообразные дизайны веб-страниц.

2. Медиа запросы: CSS позволяет разработчикам задавать различные стили в зависимости от различных параметров устройства, таких как ширина экрана, разрешение, ориентация и т.д. Это позволяет создавать адаптивные и отзывчивые дизайны, которые хорошо смотрятся на различных устройствах.

3. Каскадирование и наследование: Одна из основных концепций CSS - каскадирование стилей, которое позволяет задавать несколько правил для одного элемента и определять приоритет этих правил. Также, CSS определяет наследование стилей, что позволяет применять стили для родительских элементов к их дочерним элементам.

4. Анимации и трансформации: CSS позволяет создавать анимации и трансформации для элементов веб-страницы. Это включает в себя изменение свойств, таких как положение, размер, цвет, прозрачность и другие атрибуты, что добавляет интерактивность и живость на странице.

5. Модульность и много разовость: CSS позволяет создавать модульные стили, которые могут быть легко использованы еще раз на различных страницах. Это значительно упрощает обслуживание и управление стилями на веб-сайте.

И вот представлены плюсы каскадных таблиц:

Плюсы CSS:

1. Раздельное оформление содержания (HTML) и стиля (CSS) сайта, что упрощает его разработку, поддержку и обновление.

2. Возможность создания адаптивного и отзывчивого дизайна, который корректно отображается на различных устройствах и разрешениях экранов.

3. Эффективное управление макетами и оформлением, что позволяет однократно изменить стиль элемента на всем сайте путем редактирования соответствующего CSS-правила.

4. Возможность создания анимации и интерактивных эффектов на веб-страницах без необходимости использовать дополнительные технологии.

Но кроме плюсов, есть и множество недостатков CSS:

1. Необходимость учиться изысканной структуре и синтаксису языка, включая разные аспекты каскадирования стилей.

2. Необходимость обеспечения кроссбраузерности в стилях, чтобы обеспечить одинаковое отображение сайта в различных браузерах.

3. Не всегда возможно абсолютно точное позиционирование элементов, особенно в сложных макетах, что может требовать дополнительных усилий.

CSS является мощным и эффективным инструментом для создания и управления внешним видом веб-сайтов, но требует от разработчиков определенных усилий и внимания к деталям.

## **JavaScript**

JavaScript - один из самых популярных и широко используемых языков программирования в мире. В отличие от многих других языков, JavaScript интерпретируемый, а не компилируемый, что означает, что код выполняется на стороне клиента, в браузере пользователя. Это позволяет создавать динамические и интерактивные веб-сайты, а также современные веб-приложения.

Вот основные возможности JavaScript:

1. Динамическая типизация: в JavaScript переменные не имеют строго определенного типа данных, что делает его гибким и удобным для разработки.

2. Объектно-ориентированное программирование: JavaScript поддерживает объектно-ориентированную парадигму программирования, что позволяет создавать объекты, методы и классы.

3. Асинхронное программирование: с помощью коллбэков, промисов и асинхронных функций JavaScript позволяет эффективно работать с асинхронными операциями, такими как загрузка данных из сети.

4. Манипуляция DOM: JavaScript позволяет изменять содержимое и структуру веб-страницы, взаимодействовать с элементами DOM, реагировать на события пользователя.

5. Модульность: с развитием стандарта ECMAScript модульная система стала частью языка, что позволяет разрабатывать масштабируемые и модульные приложения.

Основные плюсы JavaScript:

1. Универсальность: JavaScript может использоваться как на стороне клиента (в браузере), так и на стороне сервера (с помощью платформы Node.js).

2. Простота изучения: Язык является относительно простым для изучения и начала работы.

3. Широкая поддержка: JavaScript поддерживается почти всеми современными браузерами.

4. Возможности для создания интерактивных пользовательских интерфейсов.

5. Большое количество библиотек и фреймворков, таких как React, Angular и Vue, которые упрощают создание веб-приложений.

И основные недостатки:

1. Интерпретируемость: JavaScript интерпретируется браузером, что может замедлить выполнение некоторых операций.

2. Один поток выполнения: JavaScript является однопоточным, что может привести к блокировке интерфейса при выполнении тяжелых вычислений.

3. Необходимость обеспечения безопасности: JavaScript подвержен уязвимостям, таким как XSS (межсайтовый скриптинг).

4. Различия между браузерами: Разные браузеры могут по-разному интерпретировать и поддерживать определенные функции JavaScript.

JavaScript играет важную роль в современной веб-разработке и продолжает привлекать новых разработчиков благодаря своей гибкости, мощности и богатой экосистеме инструментов и библиотек.

## **PHP**

PHP (Hypertext Preprocessor) является одним из самых распространенных языков программирования, используемых для создания динамических веб-сайтов и веб-приложений. PHP предоставляет широкие возможности для обработки данных, взаимодействия с базами данных и создания интерфейсов.

Важной особенностью PHP является его простота в использовании и распространение. PHP код выполняется на сервере, что делает его особенно удобным для создания динамических веб-сайтов, где страницы создаются на лету в ответ на запросы пользователя. PHP также поддерживает множество баз данных, включая MySQL, PostgreSQL и многие другие, что делает его отличным выбором для создания веб-приложений, требующих хранения и обработки больших объемов данных.

Причина популярности PHP является его расширяемость. С помощью различных библиотек и расширений разработчики могут расширить функциональность PHP в соответствии с требованиями своих проектов. Существует множество расширений для работы с различными API, взаимодействия с веб-службами, обработки изображений, генерации PDF и множества других задач. Также PHP активно используется в современном развитии веб-приложений благодаря широкой поддержке фреймворков. Фреймворки, такие как Laravel, Symfony, CodeIgniter и другие, которые предоставляют разработчикам не только готовые инструменты и архитектуру для построения веб-приложений, но и стандартизацию в рамках командной разработки.

Кроме всех плюсов PHP также имеет свои недостатки. Некоторые критики указывают на сложность и неуклюжесть языка, особенно в сравнении с более современными языками программирования. Критика также прикладывается к производительности PHP, хотя последние версии языка демонстрируют улучшение в этом направлении. Кроме того, PHP имеет свой набор уязвимостей безопасности, и важно тщательно обеспечивать безопасность веб-приложений, созданных на PHP.

## **MySQL**

MySQL является одной из самых популярных и важных реляционных систем управления базами данных (СУБД) в современной разработке программного обеспечения. Она предоставляет широкий спектр функций для работы с данными.

MySQL имеет модульную архитектуру, которая позволяет ей поддерживать различные типы движков хранения, такие как InnoDB, MyISAM, MEMORY, CSV, и другие. Это позволяет разработчикам выбирать наиболее подходящий движок для их конкретных задач, что способствует гибкости и производительности MySQL.

Одной из ключевых особенностей MySQL является его масштабируемость. MySQL может быть развернута на отдельном сервере, на кластере серверов или в облаке, что позволяет ей эффективно обработать большие объемы данных и обеспечить высокую производительность при необходимости. Существует также множество решений и инструментов, которые способствуют горизонтальному и вертикальному масштабированию MySQL, что делает ее подходящей для широкого спектра задач, включая крупномасштабные приложения и высоконагруженные веб-сайты.

MySQL также активно используется в современной разработке веб-приложений и обладает широкой поддержкой со стороны разработчиков и сообщества. Существует множество инструментов, библиотек и фреймворков, которые предоставляют разработчикам удобные средства для взаимодействия с MySQL.

Но MySQL также имеет свои недостатки, в виде ограничения производительности при обработке больших объемов данных и высоких нагрузках. Кроме того, некоторые разработчики критикуют отсутствие некоторых современных функциональностей, которые могут быть доступны в других базах данных.

MySQL остается актуальным выбором для различных проектов, от небольших веб-сайтов до крупных предприятий, благодаря своей открытости, производительности и поддержке сообщества

# **2 Практическая часть**

## **2.1 Средства разработки**

Перед разработкой приложения были выбраны удобные версии утилит, необходимых для разработке:

1. Open Server Panel 5.4.3
2. Apache 2.4
3. PHP 7.2
4. PhpStorm

## **2.2 Проектирование модели данных**

Диаграмма вариантов использования играет важную роль в процессе проектирования программных систем, позволяя формально описать функциональные требования и задачи, которые должна решать разрабатываемая система с точки зрения её конечных пользователей.

На сайте должна быть предусмотрена авторизация пользователей, разделения на разные полномочия и обязанности, пользователь может просматривать, записываться и отмять свои услуги также способен редактировать свои личные данные. Администратору доступны все данные сайта с возможностью их редактировать, как и внешний интерфейс, ему доступен также отчет использования услуг(Рисунок 1).



Рисунок 1 – Диаграмма вариантов использования

Диаграмма классов(Рисунок 2) играет ключевую роль в объектно-ориентированном проектировании и моделировании предметной области, для которой разрабатывается программная система. Представлена в виде:

* Пользователя, который должен быть авторизирован, для распределения полномочий (пользователь или администратор);
* Услуг, представляют собой классы;
* История, в которой собраны все взаимодействия пользователей с услугами.

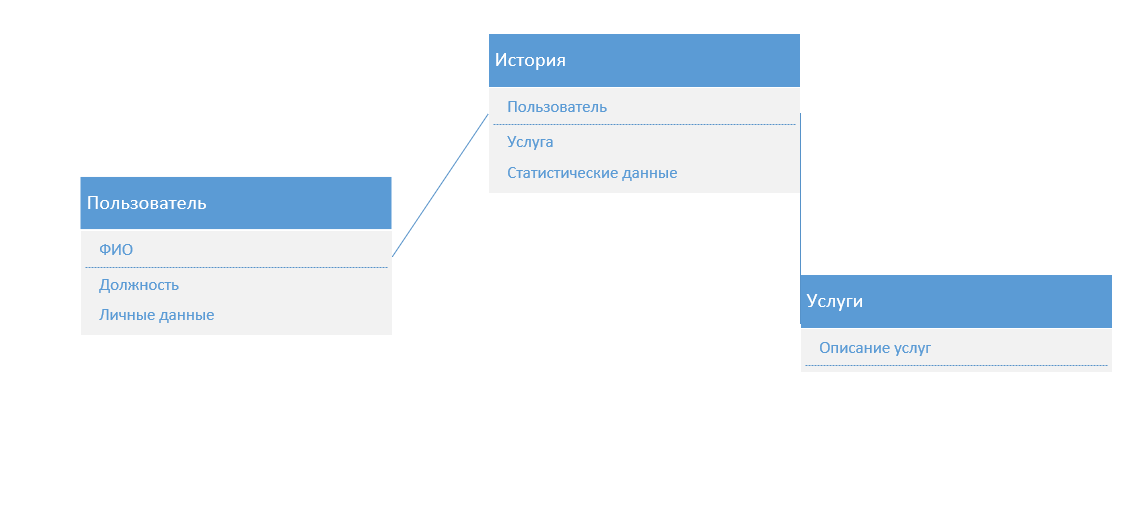


Рисунок 2 – Диаграмма классов

## **2.3 Проектирование базы данных**

С помощью диаграммы классов разработана база данных в третьей нормальной форме.

Третья нормальная форма является важной концепцией реляционных баз данных и служит для повышения их качества с точки зрения:

1. Минимизации избыточности данных. Когда таблица находится в 3NF, в ней не дублируются одни и те же данные. Это экономит место в БД и уменьшает риск возникновения аномалий при модификации данных.
2. Устранения модификационных аномалий 3NF гарантирует, что при обновлении атрибута этой сущности не придётся также обновлять другие не связанные атрибуты.
3. Повышения целостности Нормализованная до 3NF БД лучше защищена от потенциальных ошибок и несоответствий при внесении изменений в данные.
4. Упрощения структуры Комплексные логические взаимосвязи разбиваются на более простые связи в отдельных таблицах, что упрощает понимание и запросы к БД.
5. Более гибкого проектирования запросов к данным за счёт разнесения на отдельные связанные таблицы.

Это говорит о том, что вся таблица должна зависеть только от первичного ключа и ни от каких других столбцов.

## **2.4 Описание основных программных модулей**

Взаимодействие с базами данных происходит с помощью специального языка запросов, называемого SQL (Structured Query Language).

Основные типы SQL запросов, используемые для работы с данными:

SELECT - запрос на выборку данных из базы данных, самый распространенный тип запроса. Позволяет задавать различные условия выборки и сортировки.

INSERT - запрос на добавление новых данных (записей) в таблицу базы данных.

UPDATE - запрос на обновление существующих записей, позволяет изменить значения определенных полей в строках.

DELETE - запрос на удаление данных из таблицы базы, в соответствии с заданным условием.

CREATE - запрос на создание объектов базы данных - таблиц, индексов, представлений.

ALTER - запрос на изменение структуры существующих объектов БД.

DROP - запрос на удаление объектов базы данных.

Комбинируя и вложив эти основные типы SQL запросов - можно реализовать любую необходимую логику для получения и манипулирования данными из базы данных.

Например: $sql = "INSERT INTO history (s\_Ser, s\_dt, s\_name, s\_cost, s\_User, s\_n, s\_n1, s\_mail) VALUES ('$s\_Ser', NOW(), '$s\_name', '$s\_cost', '$s\_User', '$s\_n', '$s\_n1', '$s\_mail')";

Данный запрос используется для запыления отчетности использования услуги.

Для реализации взаимодействия запросов используется PHP, к примеру:

1. В начале идёт вывод данных администратора из сессии и форма для их редактирования:

$User = $\_SESSION['User'];

$login = $\_SESSION['login'];

...

Эти переменные берут нужные значения по ключам из сессии.

<form method='post' action=''>

<input value='<?php echo $login; ?>'>

...

</form>

Здесь значения выводятся в поля формы, чтобы можно было отредактировать и сохранить обратно в БД.

1. Рассмотрим вывод услуг из БД:

$sql = "SELECT \* FROM service";

Этот запрос извлекает все данные по услугам из таблицы service.

while ($row = $result->fetch\_assoc()) {

echo "<div>";

echo "<img src='{$row['img']}'>";

echo "{$row['name']}";

}

Здесь проходим циклом по строкам результата запроса и выводим данные из конкретного поля каждой строки в нужный блок на странице.

1. Пример удаления услуги из БД:

<form method='post' action='delete.php'>

<input type='hidden' value='<?php echo $row['id'];?>' name='id'>

<button type='submit'>Удалить</button>

</form>

*// Выводим данные пользователя из сессии*

$user\_name = $\_SESSION['user\_name'];

$user\_email = $\_SESSION['user\_email'];

*// Форма редактирования данных*

echo '<form action="edit\_user.php" method="post">';

echo '<input name="user\_name" value="'.$user\_name.'">';

echo '<input name="user\_email" value="'.$user\_email.'">';

echo '<button type="submit">Сохранить</button>';

echo '</form>';

*// Обработчик изменения данных*

*// edit\_user.php*

$new\_name = $\_POST['user\_name'];

$new\_email = $\_POST['user\_email'];

*// Запрос на обновление*

$sql = "UPDATE users

SET name='$new\_name', email='$new\_email'

WHERE id = {$\_SESSION['user\_id']}";

Таким образом происходит выгрузка, редактирование и сохранение обратно в БД данных о пользователе.

Далее рассмотрим функционал удаления услуги:

*// Условие удаления в массиве $services*

if(isset($\_POST['del\_service'])) {

*// Запрос на удаление*

$sql = "DELETE FROM services

WHERE id = {$\_POST['service\_id']}";

*// Перезагрузка $services*

$services = loadServices();

}

*// Кнопка удаления*

echo "<button name='del\_service'

value='{$service['id']}'>

Удалить

</button>";

Вот таким образом происходит полный цикл удаления услуги после нажатия на кнопку.

Далее результат этих действий на экране можно увидеть с помощью HTML и JS. Форма и ее атрибуты:

<form action="adm\_p\_o.php" method="post" class="ysp">

action="adm\_p\_o.php" - это обработчик формы, то есть PHP скрипт, который будет получать и обрабатывать данные формы после ее отправки.

method="post" определяет метод отправки данных формы - POST. Это более предпочтительный и безопасный метод отправки форм.

class="ysp" присваивает форме CSS класс ysp. Это нужно для применения определённых стилей к форме.

Поля ввода данных:

<input name="n" value="<?php echo $n; ?>"> - атрибут name нужен, чтобы получить это значение в скрипте-обработчике

value="<?php echo $n; ?>" - выводит PHP переменную $n как значение по умолчанию в этом поле ввода

Аналогично для других полей ввода - фамилии, почты, логина и пароля.

Кнопки управления:

При нажатии на кнопки "Выход", "Настройки сайта" и т.д. срабатывают JavaScript функции, которые делают редирект на другие страницы.

Например:

function logout() {

window.location.href = 'index.php';

}

Перенаправляет на страницу авторизации.

То есть с помощью этих функций организовано удобное клиент-сайд навигационное меню по админке.

Также данное сочетания языков позволяет делать «страницы» динамическими.

## **2.5 Основные страницы web-приложения**

Вот представлены основные страницы готового приложения.(Рисунки 1, 2, 3). Пароль и логин от админа – 1:1

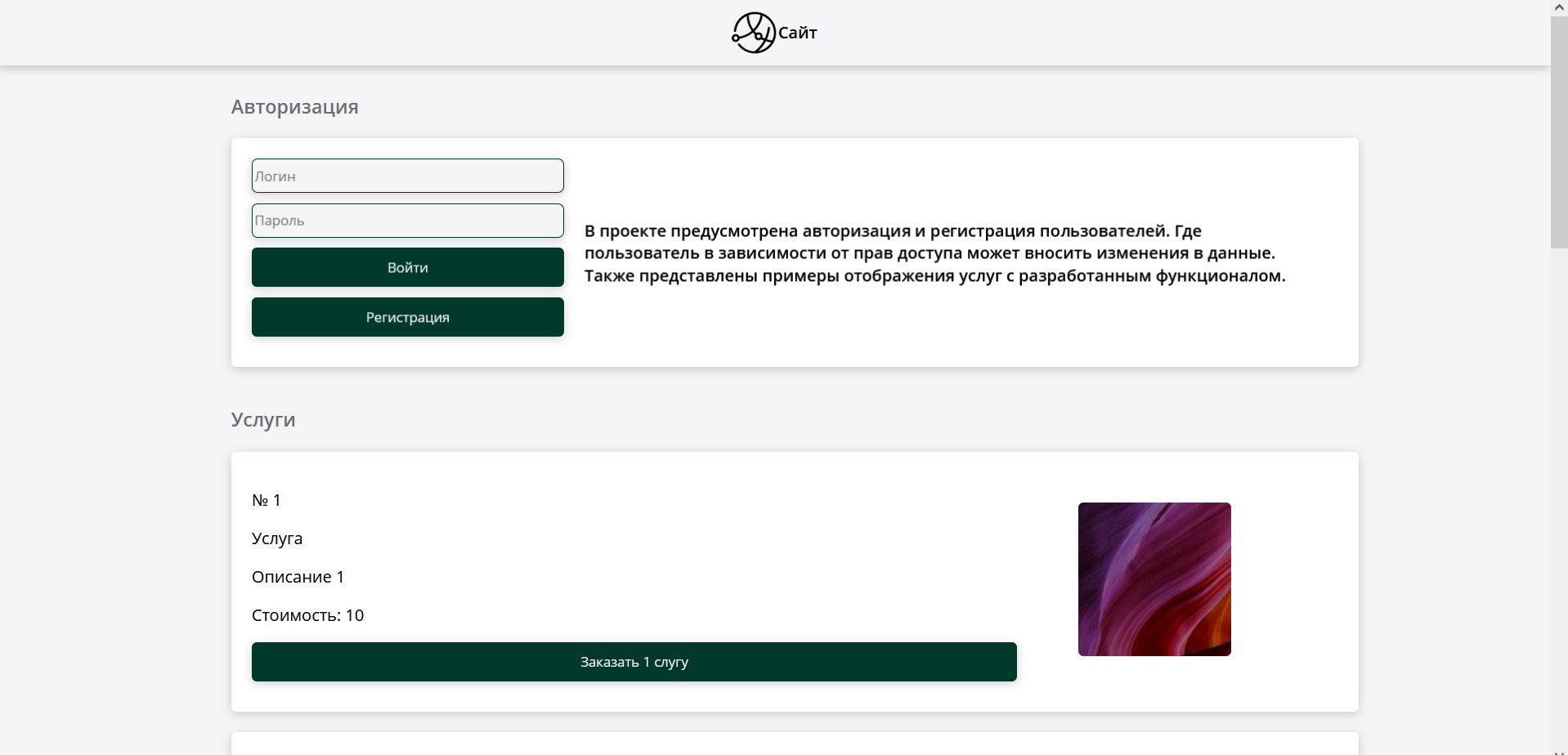


Рисунок 1 – Начальная страница

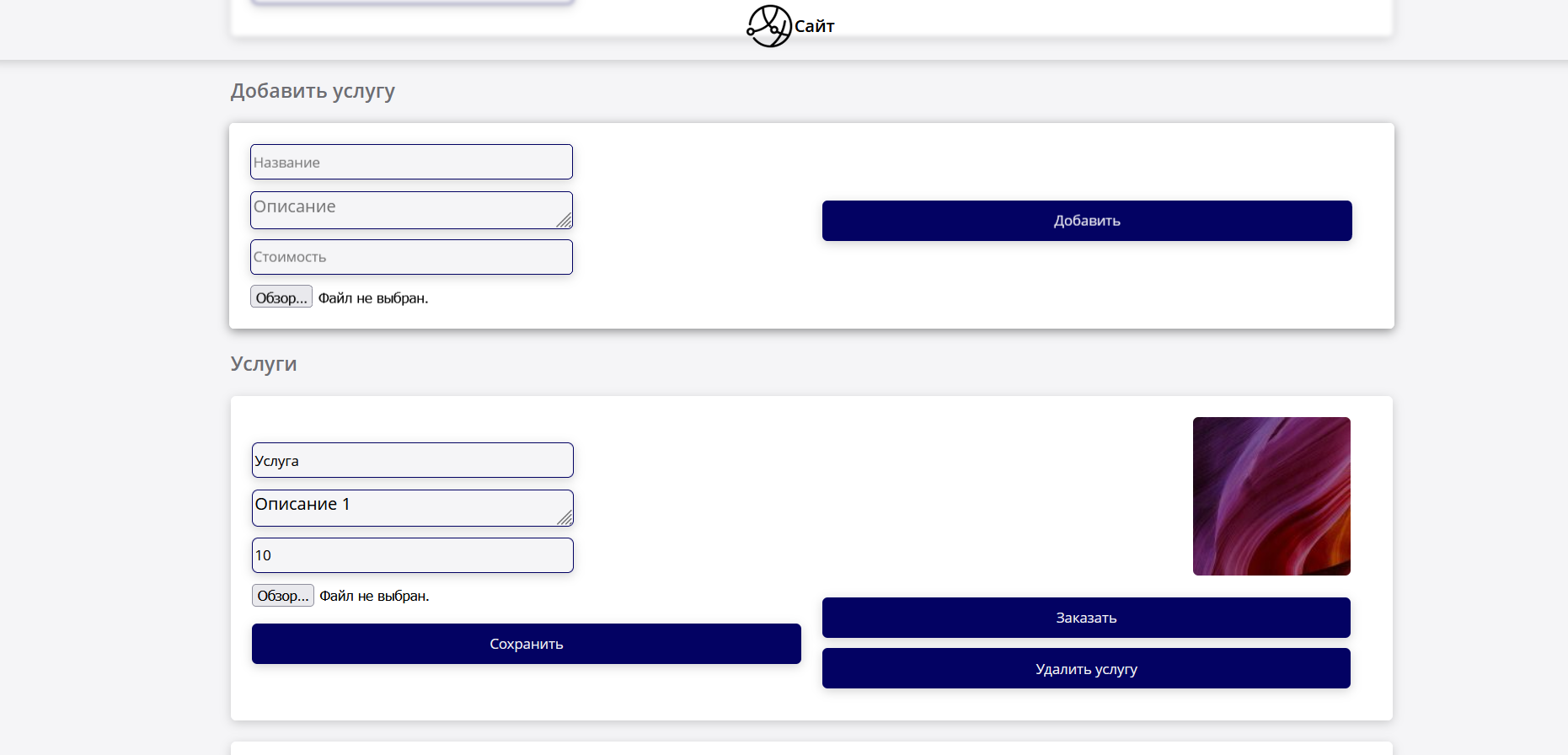


Рисунок 2 – Страница настройки услуг

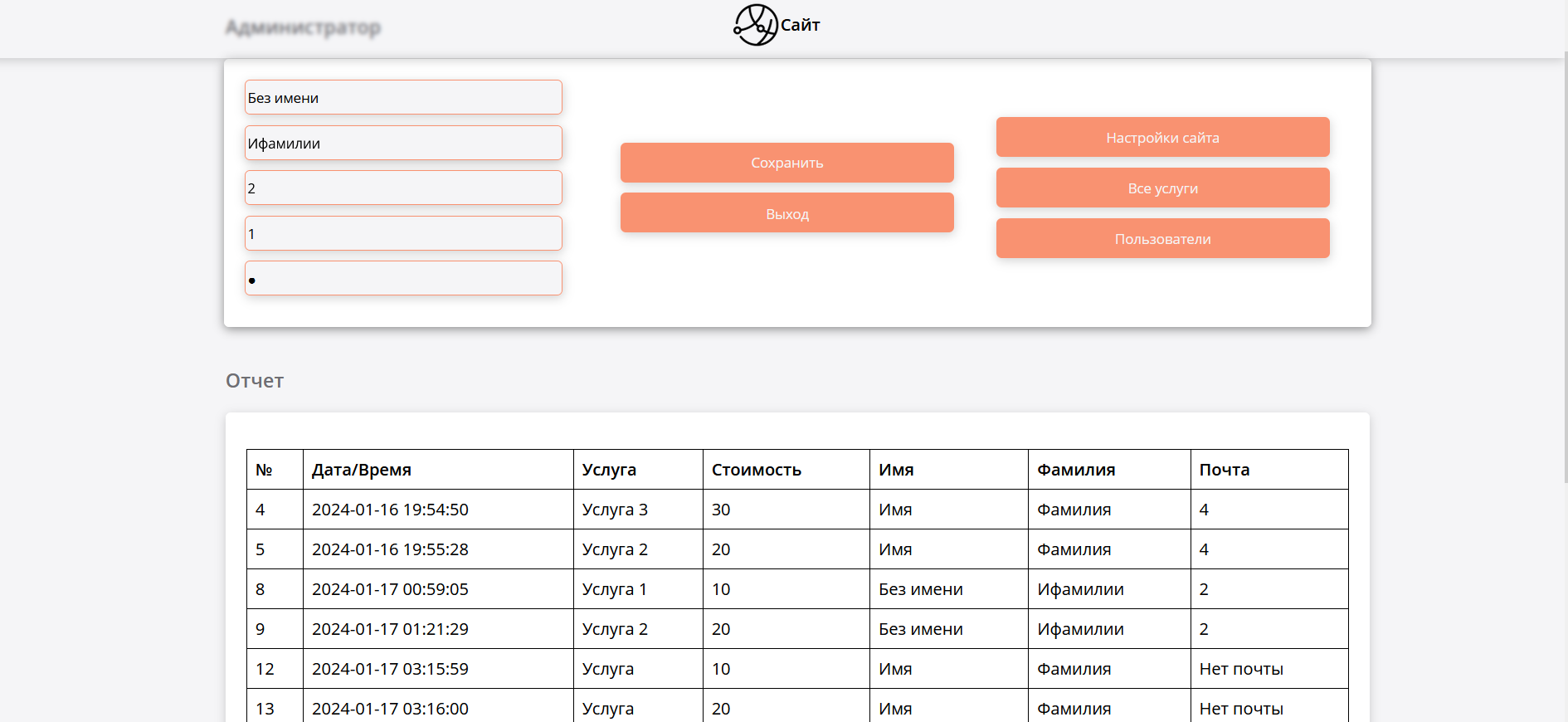


Рисунок 3 – Панель администратора «Админка»

# **ЗАКЛЮЧЕНИЕ**

Таким образом, в ходе данной курсовой работы были описаны все шаги разработки web-приложения с применением таких технологий как: HTML, CSS, JavaScript, PHP, MySQL. Данный сайт полностью удовлетворяет требования и имеет простой и функциональный интерфейс. Есть разделение пользователей и выбор различных «Услуг». Полностью реализована система регистрации пользователей и добавление доп. услуг. Дизайн сайта можно изменять под свой вкус.

# **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. MDN Web Docs - Mozilla / HTML: язык разметки гипертекста // URL: <https://developer.mozilla.org/en-US/docs/Web/HTML>. (Дата обращения 14.01.24)
2. MDN Web Docs – Mozilla / CSS: каскадные таблицы стилей // URL: <https://developer.mozilla.org/en-US/docs/Web/CSS>. (Дата обращения 14.01.24)
3. MDN Web Docs – Mozilla / JavaScript //URL: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>. (Дата обращения 14.01.24)
4. PHP: Hypertext Preprocessor / PHP: Что такое PHP? – Руководство // URL: <https://www.php.net/manual/en/intro-whatis.php>. (Дата обращения 14.01.24)
5. MySQL / MySQL Documentation // URL: <https://dev.mysql.com/doc/> (Дата обращения 14.01.24)