 ***Ministerul Educaţiei, Culturii și Cercetării al Republicii Moldova***

#### *Universitatea Tehnică a Moldovei*

#### *Facultatea Calculatoare, Informatică şi Microelectronică*

***Departamentul Informatică și Ingineria Sistemelor***

**LUCRARE DE AN**

**Tema: Aplicatie Web “Photo studio”**

**Disciplina: Baze de date**

**Elaborat: s*t. gr. IA-225 Albul Valeria***

**Verificat: asist. univ*.,***

**Chișinău 2020**

Содержание

[Глава I. Введение в BD 4](#_Toc156496037)

[1.1 Концепция баз данных: Общий обзор 4](#_Toc156496038)

[1.2 Системы Управления Базами Данных: Общий Обзор 4](#_Toc156496039)

[Глава II. Проект. BD-дизайн для данного проекта. Концептуальная модель, логическая модель. ER-диаграмма БД для данного проекта. 6](#_Toc156496040)

[2.1 Постановка задачи. 6](#_Toc156496041)

[2.2 Цель и задачи проекта 6](#_Toc156496042)

[2.3 Презентация Информационного Содержимого (Результатов) 7](#_Toc156496043)

[2.4 Представление концептуальной модели /по подразделам и/или по сущностям (объектам). 7](#_Toc156496044)

[2.5 Oписание схемы связей между между сущностями /объектами области исследования. 8](#_Toc156496045)

[2.6 Макет базы данных. 9](#_Toc156496046)

[2.7 Cхема БД в "DESIGNER" в Phpmyadmin. 10](#_Toc156496047)

[2.8 Таблица покрытия. 11](#_Toc156496048)

[Глава III (Введение в WEB-программирование). 12](#_Toc156496049)

[3.1 HTML Обзор понятий, используемых в индивидуальном годовом проекте. 12](#_Toc156496050)

[3.2 Языки скриптов. Обзор. 13](#_Toc156496051)

[3.3 PHP Обзор и концепции, используемые в индивидуальном годовом проекте ("Индивидуальная работа по обобщению курса") 14](#_Toc156496052)

[3.4 Веб-приложения. Клиент-серверная архитектура. Веб-сервер Apache, сервер данных Mysql. Реляционные базы данных. Обзор. 15](#_Toc156496053)

[3.5 Администрирование/управление базами данных с помощью MySQL. Обзор и концепции, используемые в годовом проекте. 15](#_Toc156496054)

[Глава IV. (Структура базы данных годового проекта) 17](#_Toc156496055)

[4.1 Макет Базы данных «studio» представленный по таблицам: 17](#_Toc156496056)

[4.2 Структура таблицы “clients” PhpMyadmin 17](#_Toc156496057)

[4.3 Структура таблицы “services” PhpMyadmin 17](#_Toc156496058)

[4.4 Структура таблицы “photographers” PhpMyadmin 17](#_Toc156496059)

[4.5 Структура таблицы “sessions” PhpMyadmin 18](#_Toc156496060)

[4.6 Структура таблицы “payments” PhpMyadmin 18](#_Toc156496061)

[4.7 Структура таблицы “photos” PhpMyadmin 18](#_Toc156496062)

[Глава V. (Структура веб-приложения) 19](#_Toc156496063)

[5.1 Назначение веб-приложения. 19](#_Toc156496064)

[5.2 Описание веб-приложения. Представление блок-схемы. 19](#_Toc156496065)

[5.3 Описание операций CRUD с таблицами БД. 22](#_Toc156496066)

[5.4 Компонент "Администратор". 23](#_Toc156496067)

[Глава VI (Листинг веб-приложений) 27](#_Toc156496068)

[PHP файлы: 27](#_Toc156496069)

[CSS файлы: 55](#_Toc156496070)

[Выводы 61](#_Toc156496071)

[Библиография 62](#_Toc156496072)

# **Глава II. Проект. BD-дизайн для данного проекта. Концептуальная модель, логическая модель. ER-диаграмма БД для данного проекта.**

## **2.1 Постановка задачи.**

Цель создания базы данных для записи на фотосессию в фотостудии - обеспечить удобное и эффективное управление информацией о клиентах и их заказах. Пользователи, которые будут пользоваться этим приложением, могут включать в себя как персонал фотостудии, так и сами клиенты. Рассмотрим две основные категории пользователей:

1. Администраторы фотостудии:
   * Запись новых клиентов и управление их данными.
   * Организация и отслеживание расписания фотосессий.
   * Мониторинг оплат и финансовых транзакций.
   * Поддержание актуальной информации о доступных услугах и тарифах.
   * Генерация отчетов и статистики о работе фотостудии.
2. Клиенты:
   * Регистрация в системе и создание личного профиля.
   * Запись на фотосессию, выбор доступных услуг и определение предпочтительного времени.
   * Просмотр истории заказов и оплат.
   * Взаимодействие с фотостудией через обратную связь.
   * Получение уведомлений о статусе заказа и подтверждениях.

Общие функциональности базы данных могут включать в себя:

* Управление данными о клиентах (личная информация, контакты).
* Расписание фотосессий с возможностью добавления, редактирования и удаления.
* Учет заказанных услуг, оплат и дополнительных опций.
* Система уведомлений для администраторов и клиентов.
* Интерфейс для генерации отчетов и статистики.

Такое приложение позволит фотостудии более эффективно управлять бизнес-процессами, предоставлять удобные возможности для клиентов, и повысит общую эффективность работы.

## **2.2 Цель и задачи проекта**

**Цель проекта для фотостудии:** Создание и внедрение информационной системы (включая базу данных) для управления фотосессиями и клиентскими данными с целью повышения эффективности работы фотостудии, улучшения обслуживания клиентов и оптимизации бизнес-процессов.

**Задачи проекта:**

1. **Разработка базы данных:**
   * Определение структуры базы данных для хранения информации о клиентах, заказах и расписании фотосессий.
   * Создание таблиц для учета услуг, тарифов, и других сопутствующих данных.
2. **Система записи и управления расписанием:**
   * Разработка механизма онлайн-записи на фотосессии.
   * Интеграция функционала управления расписанием для администраторов.
3. **Профили клиентов:**
   * Реализация системы учета и хранения личных данных клиентов.
   * Создание возможности для клиентов просматривать историю своих заказов и изменять свои предпочтения.
4. **Учет заказов и оплат:**
   * Внедрение функционала для оформления заказов и выбора услуг.
   * Организация системы учета оплат и выставления счетов.

**2.3 Презентация Информационного Содержимого (Результатов)**

Сформулированный проект породит ряд результатов (output'ов), которые станут информационным содержимым разработанной системы управления деятельностью фотостудии. Эти результаты будут служить для достижения поставленных целей в рамках проекта. Вот обзор основных элементов информационного содержимого:

**База Данных:**

* **Таблицы и Связи:** Структура базы данных будет включать таблицы для хранения данных о клиентах, заказах, услугах, расписаниях и др. Связи между этими таблицами обеспечат целостность и непротиворечивость данных.

**Система Управления Пользователями:**

* **Профили Пользователей:** Создание профилей для различных пользователей, таких как администраторы и клиенты, с различными уровнями доступа.

**Пользовательский Интерфейс (UI):**

* **Панель Управления:** Интуитивный интерфейс для администраторов с удобным доступом к функциональностям управления данными и мониторинга активности.
* **Портал для Клиентов**: Дружественный интерфейс для клиентов, предоставляющий возможность бронирования, просмотра и изменения личных данных, размещения заказов и взаимодействия с фотостудией.

**Система Бронирования :**

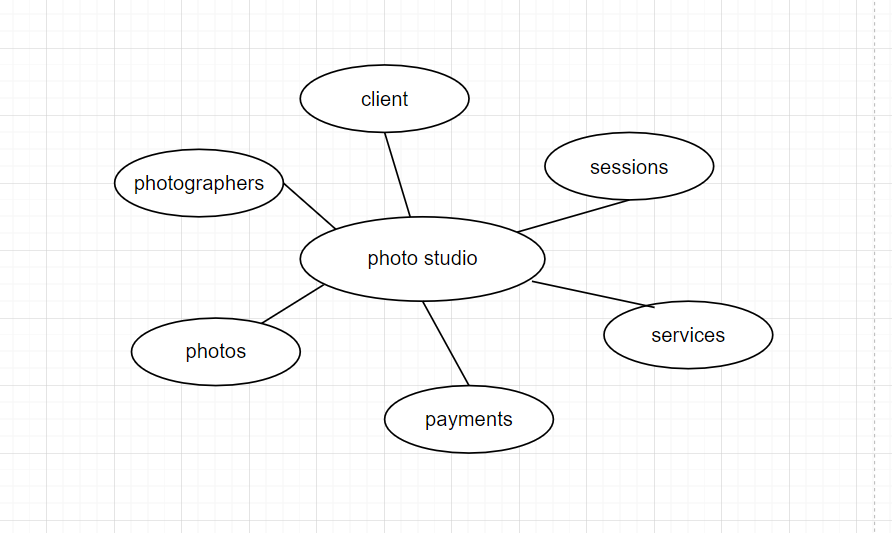
* **Интерактивный Календарь:** Система онлайн-бронирования для клиентов, позволяющая выбирать и бронировать дату для съемок.

**Управление Заказами и Платежами:**

* **Форма Заказа:** Упрощенная система для клиентов для размещения заказов на услуги фотостудии.

## **2.4 Представление концептуальной модели /по подразделам и/или по сущностям (объектам).**

Концептуальная модель базы данных представляет собой высокоуровневое описание структуры данных и их взаимосвязей. Ниже приведена концептуальная модель для базы данных, включающая таблицы "sessions", "services", "payments", "clients", "photos" и "photographers".



**Рис 2.4.1** Концептуальная модель

## **2.5 Oписание схемы связей между между сущностями /объектами области исследования.**

Схема связей между сущностями базы данных фотостудии представляет собой взаимосвязи и зависимости между различными таблицами. Ниже приведено описание основных связей между сущностями/объектами области исследования:

1. **Связь между таблицами "sessions" и "services":**
   * Каждая фотосессия ("sessions") связана с конкретной услугой ("services") через внешний ключ **service\_id**. Эта связь позволяет отслеживать, какая услуга была выбрана для каждой фотосессии.
2. **Связь между таблицами "sessions" и "clients":**
   * Каждая фотосессия ("sessions") связана с клиентом ("clients") через внешний ключ **client\_id**. Эта связь устанавливает, для какого клиента была проведена каждая сессия.
3. **Связь между таблицами "sessions" и "photographers":**
   * Каждая фотосессия ("sessions") связана с фотографом ("photographers") через внешний ключ **photographer\_id**. Эта связь определяет, какой фотограф проводил каждую сессию.
4. **Связь между таблицами "payments" и "sessions":**
   * Каждый платеж ("payments") связан с конкретной фотосессией ("sessions") через внешний ключ **session\_id**. Эта связь помогает отслеживать, какие сессии были оплачены, и связывает информацию о платеже с конкретной фотосессией.
5. **Связь между таблицами "photos" и "sessions":**
   * Каждая фотография ("photos") связана с конкретной фотосессией ("sessions") через внешний ключ **session\_id**. Эта связь позволяет связывать фотографии с определенной сессией.
6. **Связь между таблицами "clients" и "sessions":**
   * Каждая фотосессия ("sessions") связана с клиентом ("clients") через внешний ключ **client\_id**. Эта связь позволяет устанавливать, для какого клиента проводится каждая фотосессия.
7. **Связь между таблицами "photographers" и "sessions":**
   * Каждая фотосессия ("sessions") связана с фотографом ("photographers") через внешний ключ **photographer\_id**. Эта связь помогает отслеживать, какие фотографы проводят каждую сессию.

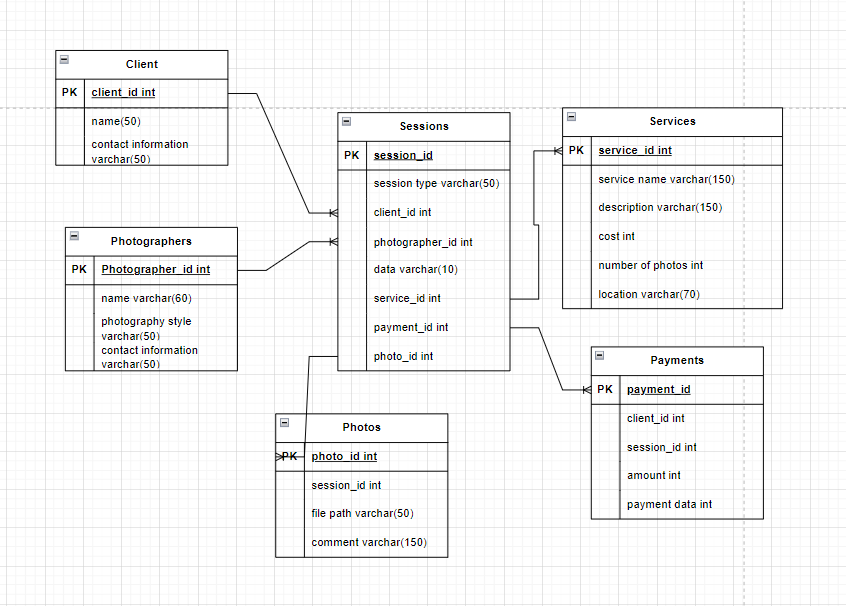
Эти связи обеспечивают целостность данных и позволяют эффективно анализировать и управлять информацией о фотосессиях, клиентах, услугах, платежах, фотографиях и фотографах в контексте фотостудии.

## **2.6 Макет базы данных.**

Основываясь на предыдущих обсуждениях концептуальной модели и связей между сущностями, вот макет базы данных для фотостудии. Этот макет представляет собой описание таблиц с полями, связями между ними и внешними ключами:

1. **Таблица "sessions" (Фотосессии):**
   * **session\_id** (PK) - Идентификатор сессии
   * **client\_id** (FK) - Идентификатор клиента (ссылка на "clients")
   * **photographer\_id** (FK) - Идентификатор фотографа (ссылка на "photographers")
   * **service\_id** (FK) - Идентификатор услуги (ссылка на "services")
   * **date**- Дата фотосессии
2. **Таблица "services" (Услуги):**
   * **service\_id** (PK) - Идентификатор услуги
   * **service\_name** - Наименование услуги
   * **price** - Стоимость услуги
3. **Таблица "payments" (Платежи):**
   * **payment\_id** (PK) - Идентификатор платежа
   * **session\_id** (FK) - Идентификатор сессии (ссылка на "sessions")
   * **amount** - Сумма платежа
   * **payment\_date** - Дата платежа
4. **Таблица "clients" (Клиенты):**
   * **client\_id** (PK) - Идентификатор клиента
   * **name** - Имя клиента
   * **email** - Адрес электронной почты клиента
   * **phone\_number** - Номер телефона клиента
5. **Таблица "photos" (Фотографии):**
   * **photo\_id** (PK) - Идентификатор фотографии
   * **session\_id** (FK) - Идентификатор сессии (ссылка на "sessions")
   * **photo\_url** - URL фотографии
6. **Таблица "photographers" (Фотографы):**
   * **photographer\_id** (PK) - Идентификатор фотографа
   * **name** - Имя фотографа
   * **email** - Адрес электронной почты фотографа
   * **phone\_number** - Номер телефона фотографа

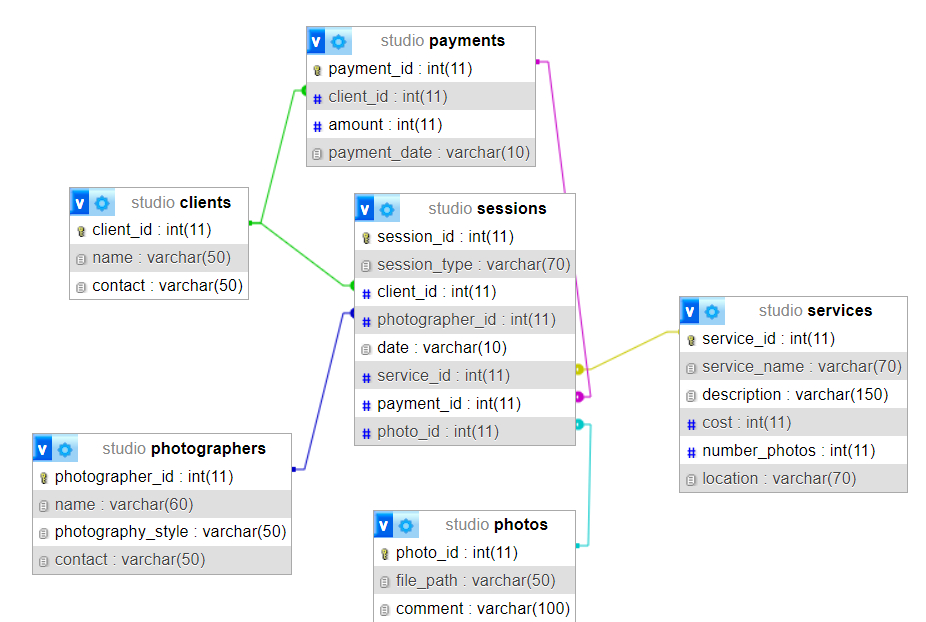
Это всего лишь общий макет, и при реализации базы данных могут быть учтены более детальные аспекты, такие как индексы, ограничения целостности, а также дополнительные поля для поддержки бизнес-логики и требований фотостудии.



**Рис 2.6.1** Макет БД

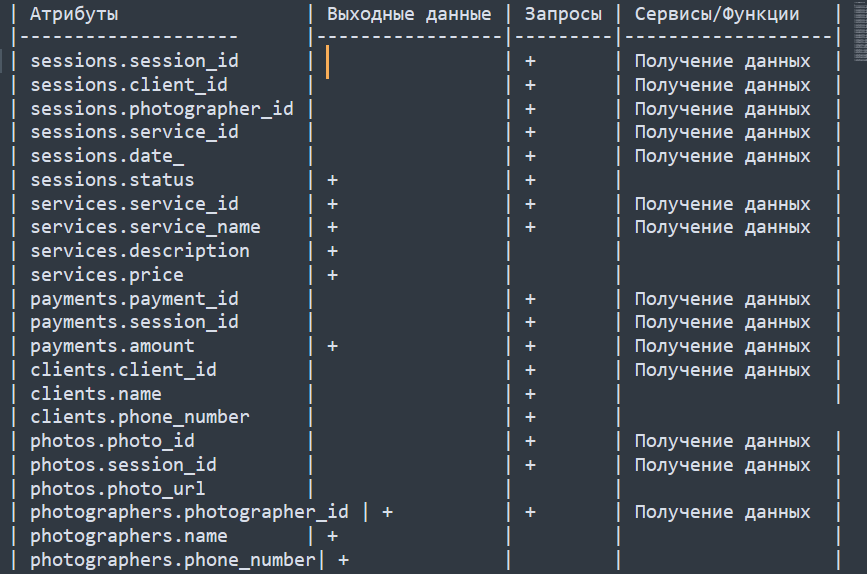
## **2.7 Cхема БД в "DESIGNER" в Phpmyadmin.**

Схема базы данных - это визуальное представление структуры вашей базы данных, которое обеспечивает ясное представление об отношениях между таблицами и их атрибутах. В PhpMyAdmin существует инструмент 'DESIGNER', который облегчает процесс проектирования базы данных. В этом визуальном редакторе вы можете создавать, изменять и визуально анализировать структуру своей базы данных. С помощью 'DESIGNER' вы можете легко определить таблицы, их поля и связи между ними, что упрощает процесс проектирования и обслуживания вашей базы данных.



**Рис. 2.7.1** Cхема БД в "DESIGNER" в Phpmyadmin.

## **2.8 Таблица покрытия.**



**Рис. 2.8.1** Таблица покрытия.

**Объяснение:**

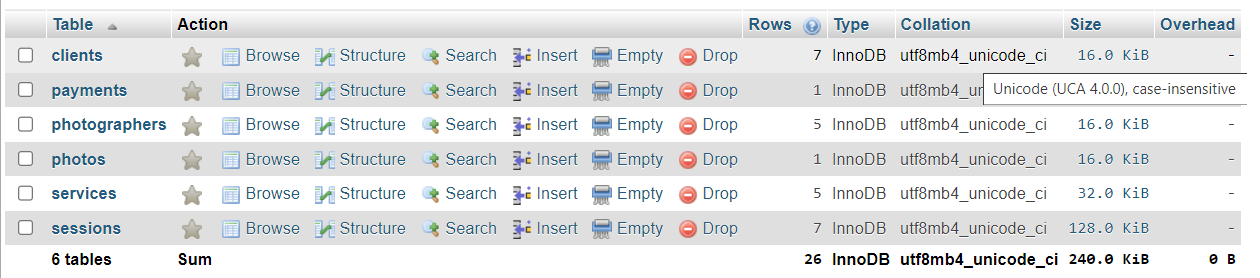
1. **Атрибуты:** Это поля в ваших таблицах.
2. **Выходные данные:** Какие атрибуты влияют на выходные данные, например, при отображении информации для пользователя.
3. **Запросы:** Какие атрибуты используются в запросах для извлечения или обновления данных.
4. **Сервисы/Функции:** Какие атрибуты участвуют в предоставлении определенных сервисов или функциональности.

Пример:

* Для атрибута **sessions.client\_id** мы видим, что он используется как в выходных данных (например, для отображения имени клиента), так и в запросах (например, для получения данных сессии клиента).

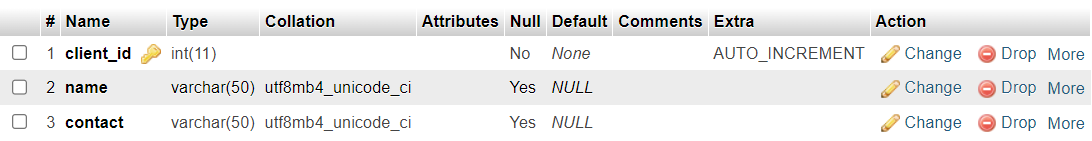
# **Глава IV. (Структура базы данных годового проекта)**

## **4.1 Макет Базы данных «studio» представленный по таблицам:**



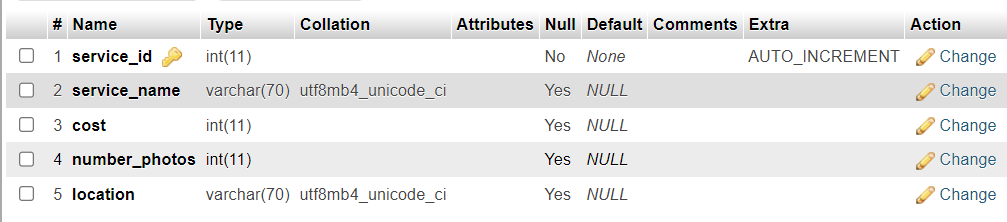
**Рис. 4.1** Список таблиц.

## **4.2 Структура таблицы “clients” PhpMyadmin**



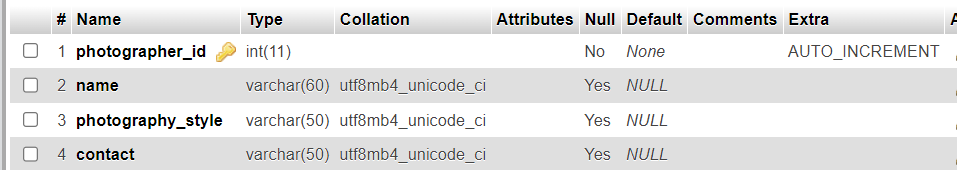
**Рис. 4.2** Структура таблицы для хранения данных о клиентах.

## **4.3 Структура таблицы “services” PhpMyadmin**



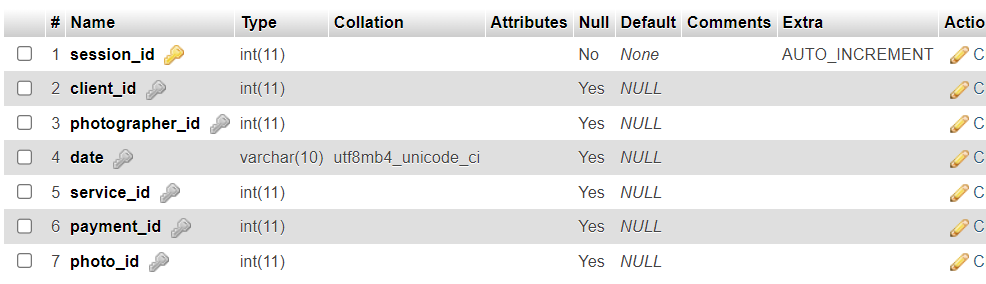
**Рис. 4.3** Структура таблицы для хранения данных о сервисах

## **4.4 Структура таблицы “photographers” PhpMyadmin**



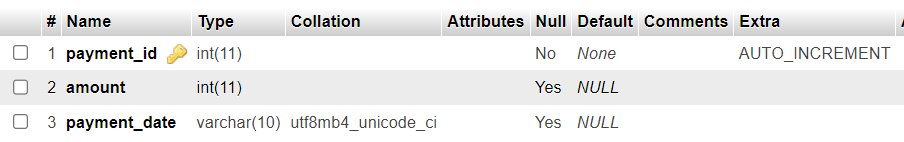
**Рис. 4.4**Структура таблицы для хранения данных о фотографах

## **4.5 Структура таблицы “sessions” PhpMyadmin**



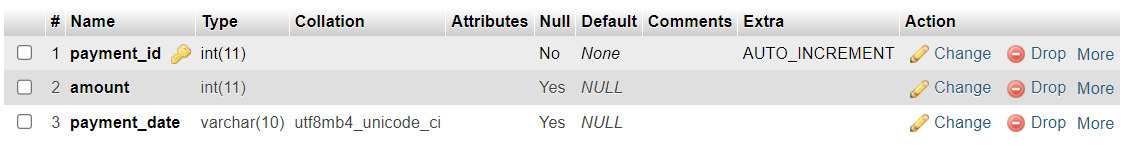
**Рис. 4.5** Структура таблицы для хранения данных о сессиях

## **4.6 Структура таблицы “payments” PhpMyadmin**



**Рис. 4.6** Структура таблицы для хранения данных о оплатах

## **4.7 Структура таблицы “photos” PhpMyadmin**



**Рис. 4.3** Структура таблицы для хранения данных о фотографиях

# **Глава V. (Структура веб-приложения)**

## **5.1 Назначение веб-приложения.**

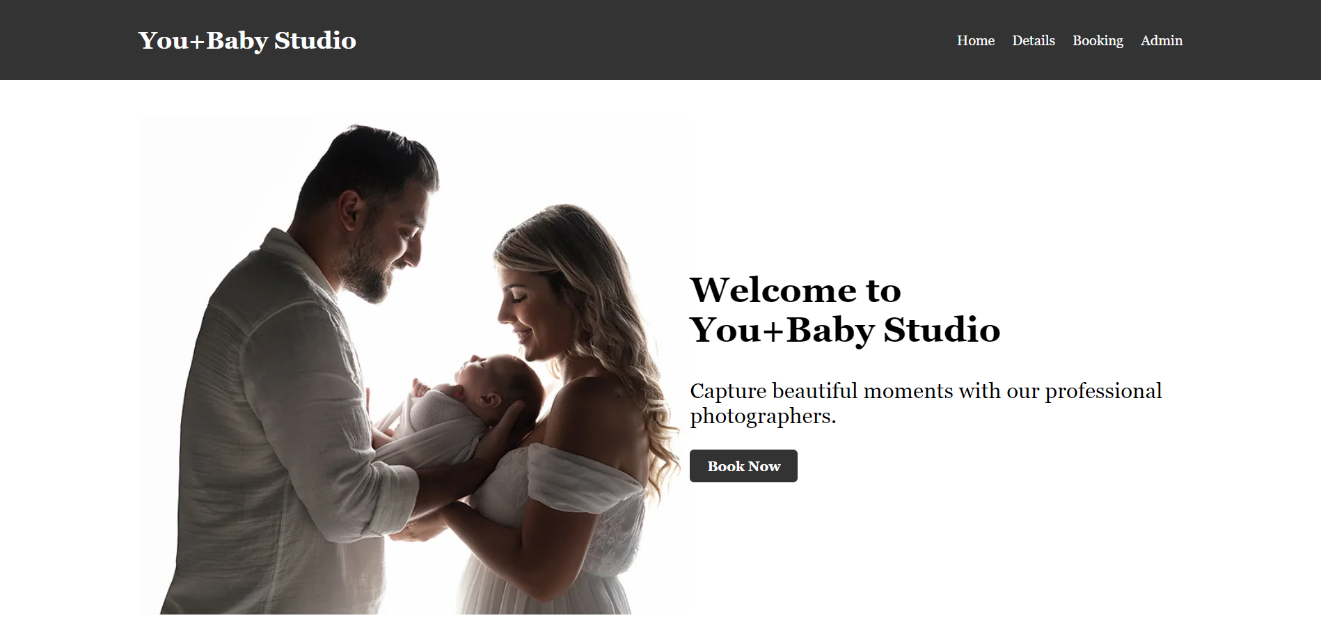
Целью создания веб-сайта для записи на фотосессии в вашей фотостудии является предоставление клиентам удобного и эффективного инструмента для онлайн-бронирования фотосессий. Этот веб-сайт ставит своей задачей упростить процесс записи, делая его быстрым и интуитивно понятным.

Основными назначениями веб-платформы являются:

1. **Удобство Записи:** Ваш сайт предоставляет посетителям простой и понятный интерфейс, сделав процесс онлайн-записи максимально удобным.
2. **Информационная Поддержка:** В разделах сайта подробно представлена информация о различных услугах фотосессий, стилях фотографии и стоимости услуг. Посетители могут ознакомиться с вашим портфолио и выбрать подходящий формат фотосессии.
3. **Административная Панель:** Административная панель предоставляет вам инструменты для эффективного управления заявками, клиентской базой данных и фотографов.

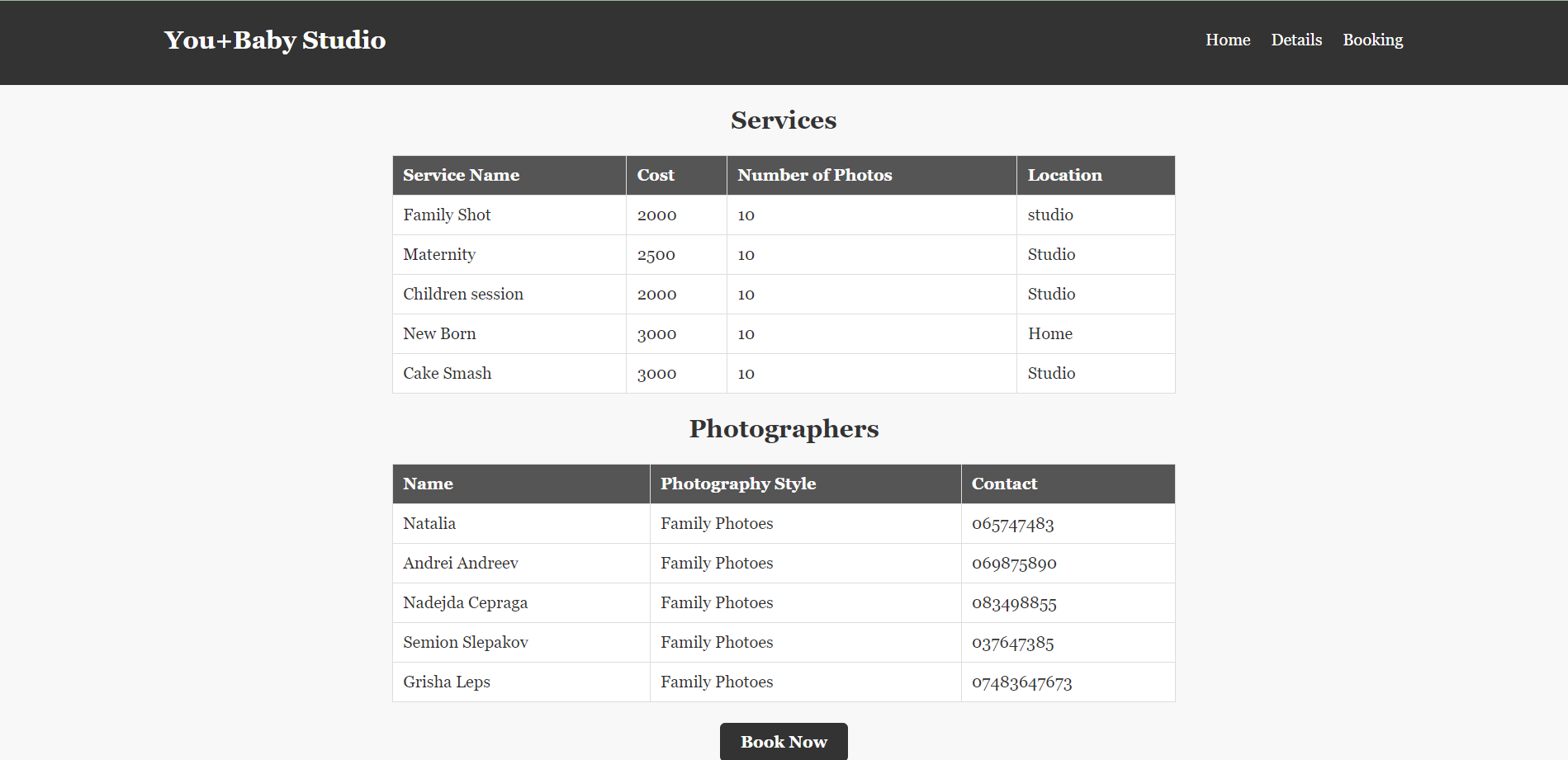
## **5.2 Описание веб-приложения. Представление блок-схемы.**

You+Baby Studio - это веб-приложение, созданное для удобного бронирования фотосессий в студии. Он предоставляет клиентам возможность легко и быстро записаться на услуги фотографа для создания красивых и запоминающихся фотографий.



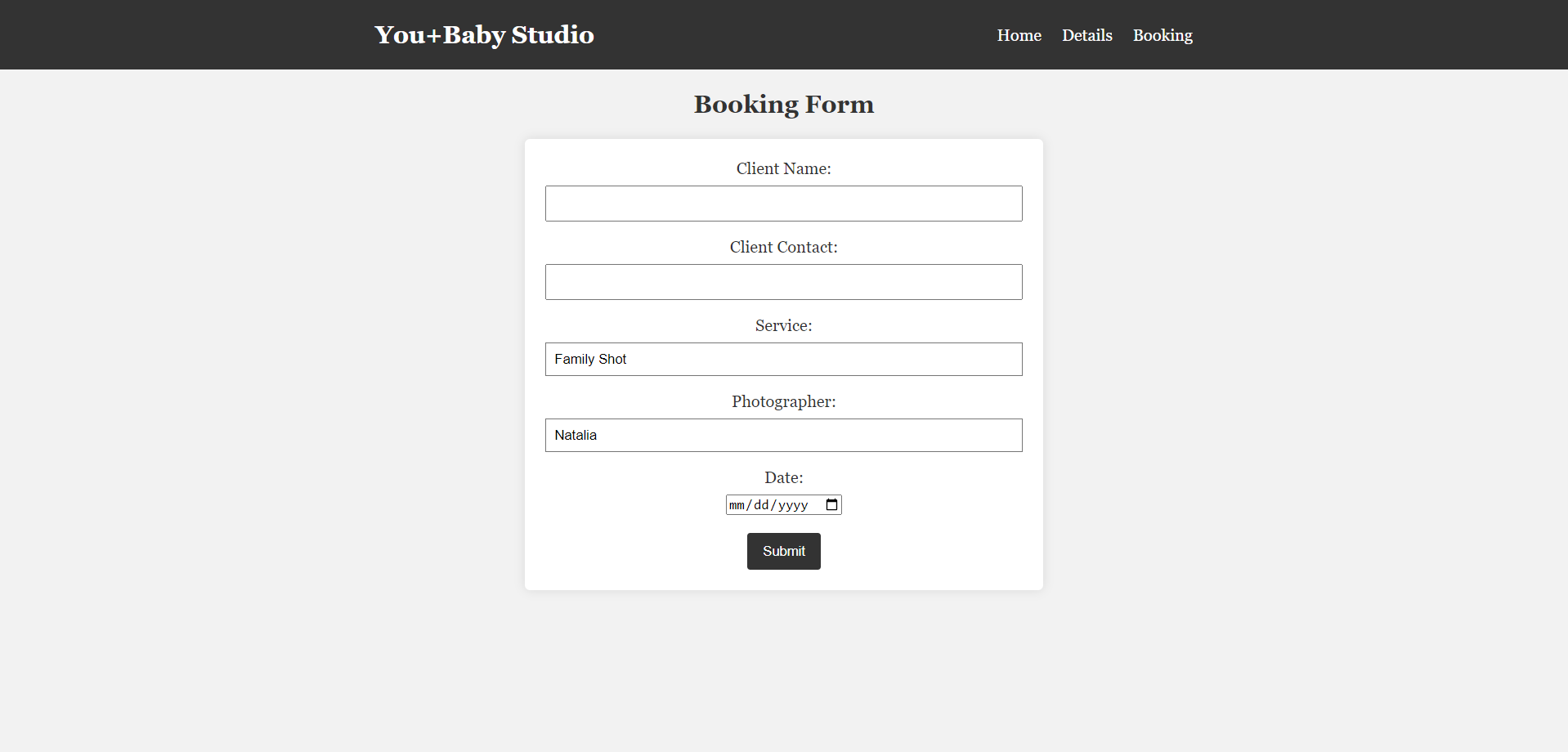
**Рис. 5.1** Главная страница

Назначение главной страницы – визуально представить потенциальным клиентам атмосферу и стиль студии. Здесь размещены привлекательные фотографии и информация о том, как мы делаем ваш опыт фотосессии неповторимым. Основная цель – вдохновить посетителей и направить их к действию. Кнопка "Book Now" становится ключом к заполнению формы бронирования, начиная процесс создания уникальной фотосессии.



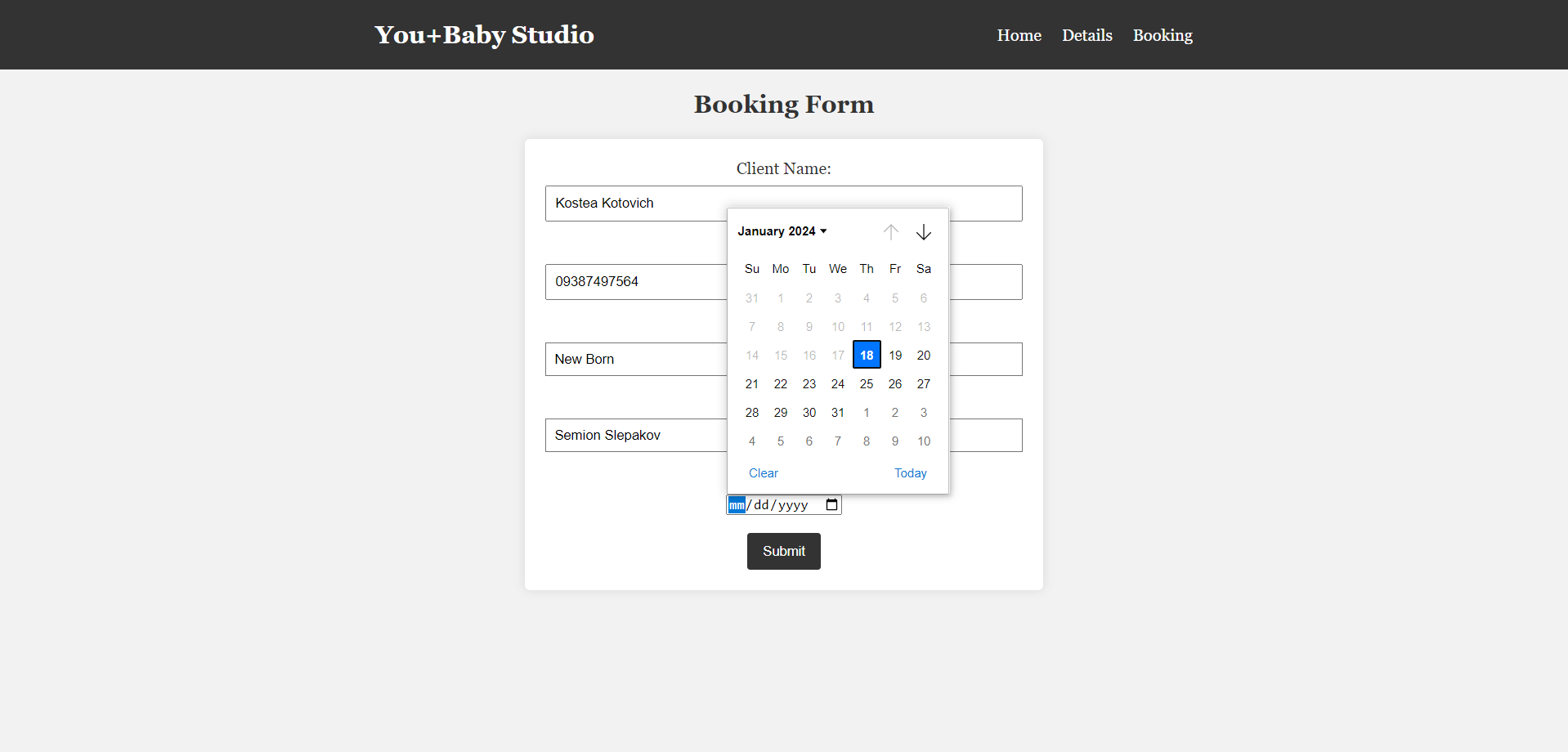
**Рис.5.2** Страница details

Страница "Details" призвана предоставить подробную информацию о наших услугах, фотографах и доступных пакетах. Здесь клиенты могут ознакомиться с различными вариантами фотосессий, а также выбрать наилучший вариант, соответствующий их пожеланиям. Эта страница помогает пользователям принимать осознанные решения перед бронированием.



**Рис.5.3** Страница “booking”

Страница "Booking" предоставляет пользователю удобную форму для бронирования фотосессии. Форма собирает основные данные, такие как имя клиента, контактная информация, выбранная услуга, предпочтительная дата и фотограф. Заполнив форму, клиенты инициируют процесс создания своей персональной фотосессии. Кнопка "Submit" отправляет данные для обработки и подтверждения бронирования.



**Рис.5.4** Страница “booking”. Заполнение формы.

Эта форма предоставляет удобный способ клиентам бронировать свою фотосессию. Все необходимые поля помечены звездочкой (\*), что обозначает их обязательность для заполнения.

**1. Имя клиента:**

* Текстовое поле "Client Name" предназначено для ввода имени клиента.
* Обязательное поле.

**2. Контактные данные клиента:**

* Текстовое поле "Client Contact" предназначено для ввода контактной информации клиента.
* Обязательное поле.

**3. Выбор услуги:**

* Выпадающий список "Service" содержит варианты услуг, доступных для бронирования.
* Варианты услуг загружаются из базы данных.
* Обязательное поле.

**4. Выбор фотографа:**

* Выпадающий список "Photographer" содержит имена доступных фотографов.
* Варианты фотографов загружаются из базы данных.
* Обязательное поле.

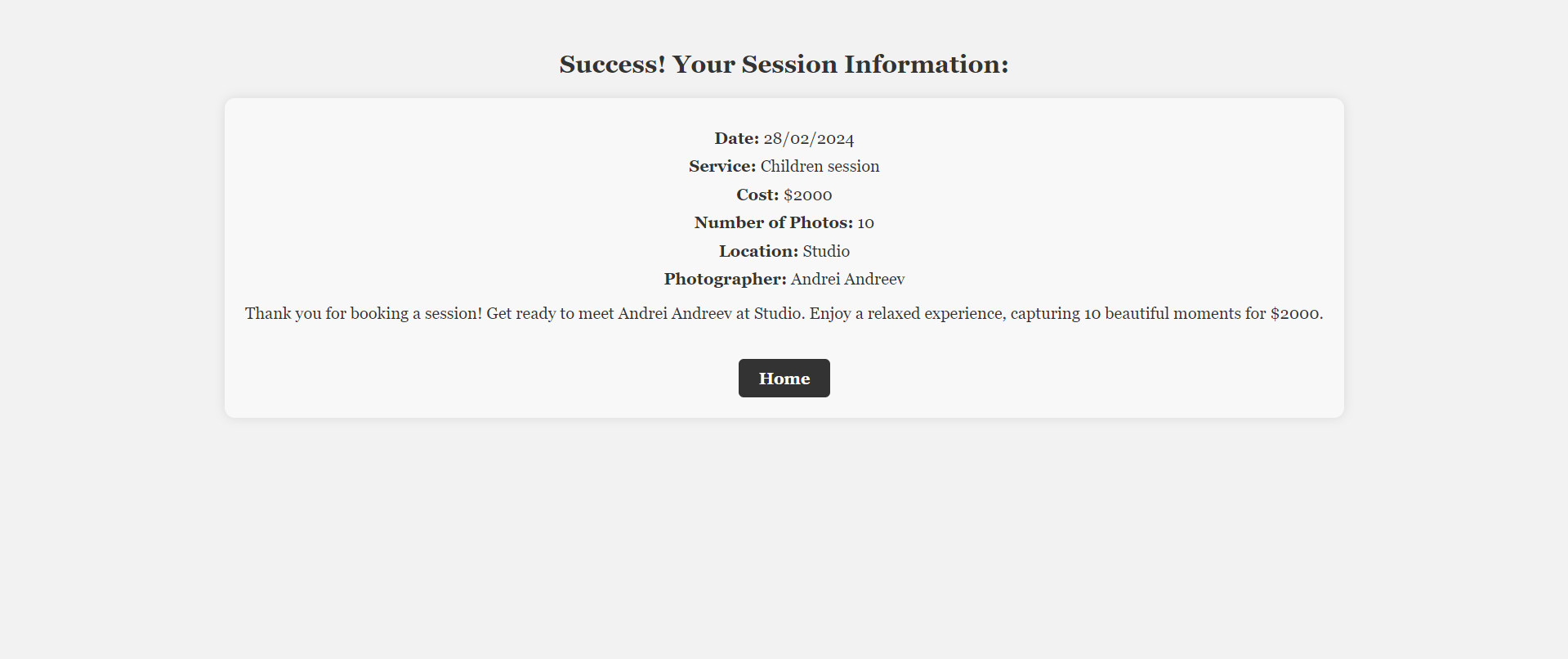
**5. Дата фотосессии:**

* Поле "Date" предоставляет возможность выбора предпочтительной даты для фотосессии.
* Минимальная дата установлена на текущую дату.
* Обязательное поле.

**6. Кнопка "Submit":**

* Кнопка "Submit" предназначена для отправки данных формы.
* После заполнения формы, клиент может нажать "Submit" для отправки запроса на бронирование.
* Отправленные данные обрабатываются на сервере с использованием PHP-скрипта (process\_booking.php).

Форма разработана с учетом простоты использования и предоставляет клиентам возможность удобного и быстрого бронирования фотосессии.



**Рис.5.5** Полезная информация после записи

## **5.3 Описание операций CRUD с таблицами БД.**

Операции CRUD (Create, Read, Update, Delete) представляют основные действия, выполняемые с данными в базе данных. Ниже приведено описание каждой из этих операций в контексте таблиц БД вашего веб-приложения.

**1. Create (Создание):**

* **Таблица "Clients":** Добавление нового клиента при бронировании фотосессии. Когда клиент заполняет форму на странице бронирования, его данные добавляются в таблицу "Clients".
* **Таблица "Sessions":** Создание новой записи о фотосессии. После успешного бронирования фотосессии данные о сессии добавляются в таблицу "Sessions".

**2. Read (Чтение):**

* **Таблица "Clients":** Отображение списка клиентов на странице администратора. Позволяет администратору просматривать информацию о всех клиентах.
* **Таблица "Sessions":** Просмотр информации о всех сессиях на странице администратора. Дает администратору возможность отслеживать все фотосессии.

**3. Update (Обновление):**

* **Таблица "Sessions":** Редактирование данных о фотосессии. Администратор может изменять детали существующих сессий, такие как клиент, фотограф, дата и тип услуги.

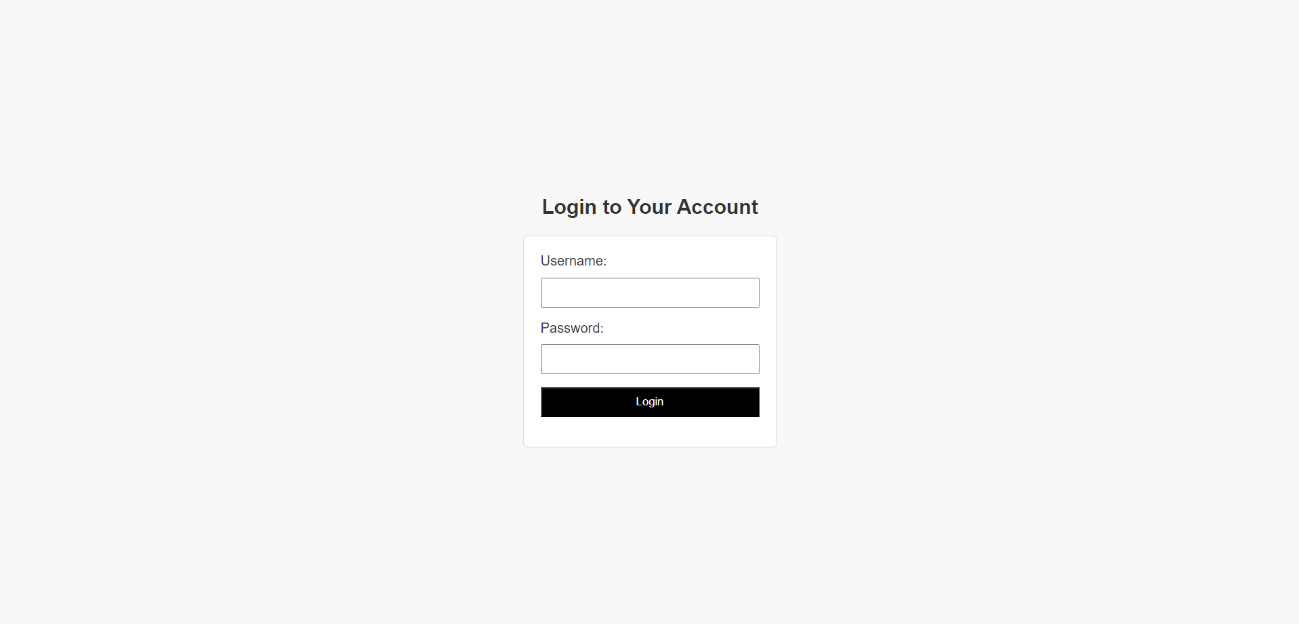
**4. Delete (Удаление):**

* **Таблица "Sessions":** Удаление записи о фотосессии. Администратор может удалять сессии, которые больше не актуальны или были отменены.
* **Таблица "Clients":** Удаление данных о клиентах. В случае необходимости администратор может удалять информацию о клиентах.
* **Таблица "Services":** Удаление услуг. Администратор может редактировать перечень предоставляемых услуг, включая добавление и удаление услуг.
* **Таблица "Photographers":** Удаление данных о фотографах. В случае изменения состава фотографов администратор может управлять списком фотографов.

Операции CRUD обеспечивают полный жизненный цикл данных в приложении, начиная от создания новых записей и заканчивая их удалением, обеспечивая тем самым управление всеми аспектами информации в базе данных.

## **5.4 Компонент "Администратор".**

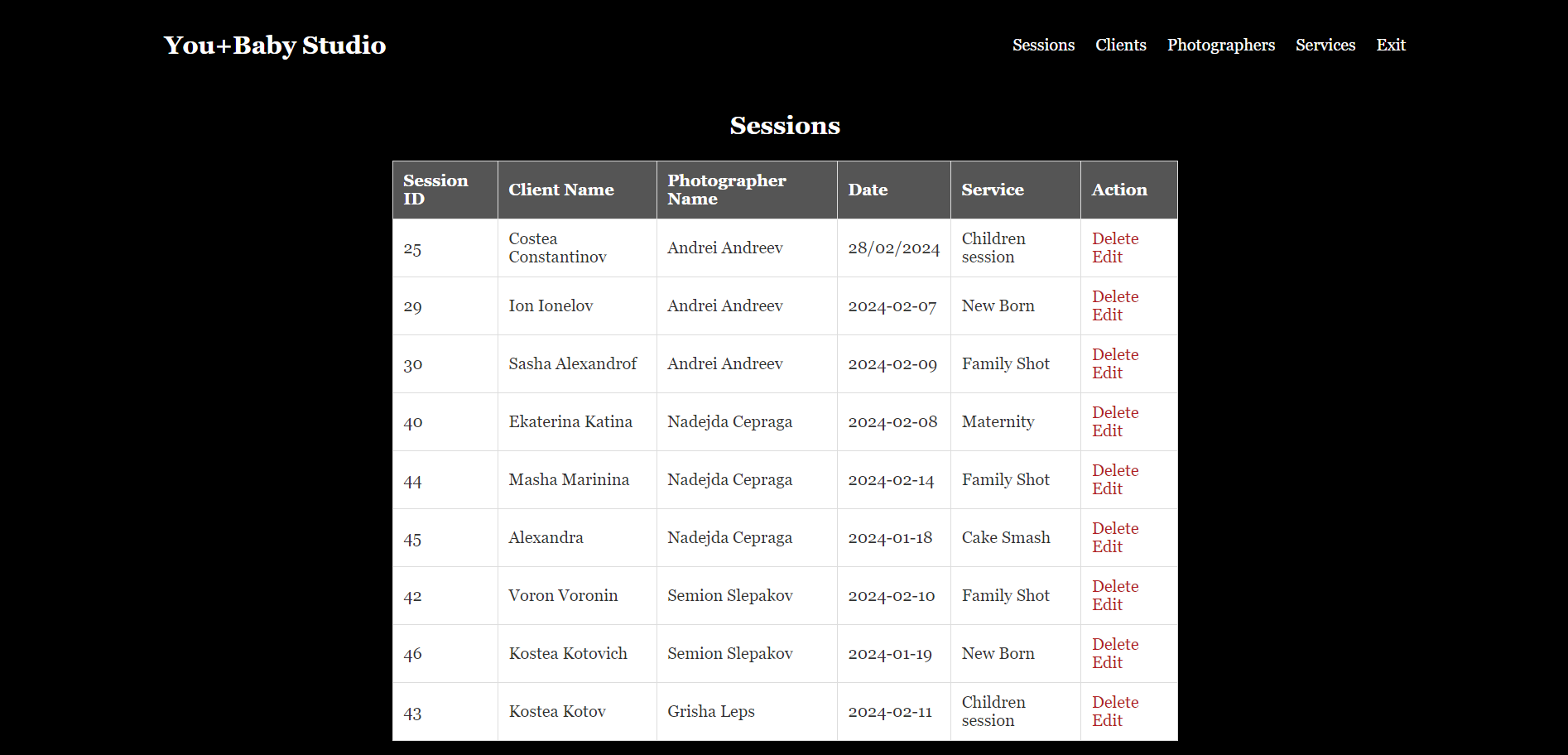
На странице администратора предоставляются различные функции для управления данными, связанными с фотосессиями и клиентами. Эта страница обеспечивает администратора всей необходимой информацией и инструментами для эффективного управления фотостудией.



**Рис.5.6** Вход в административную панель

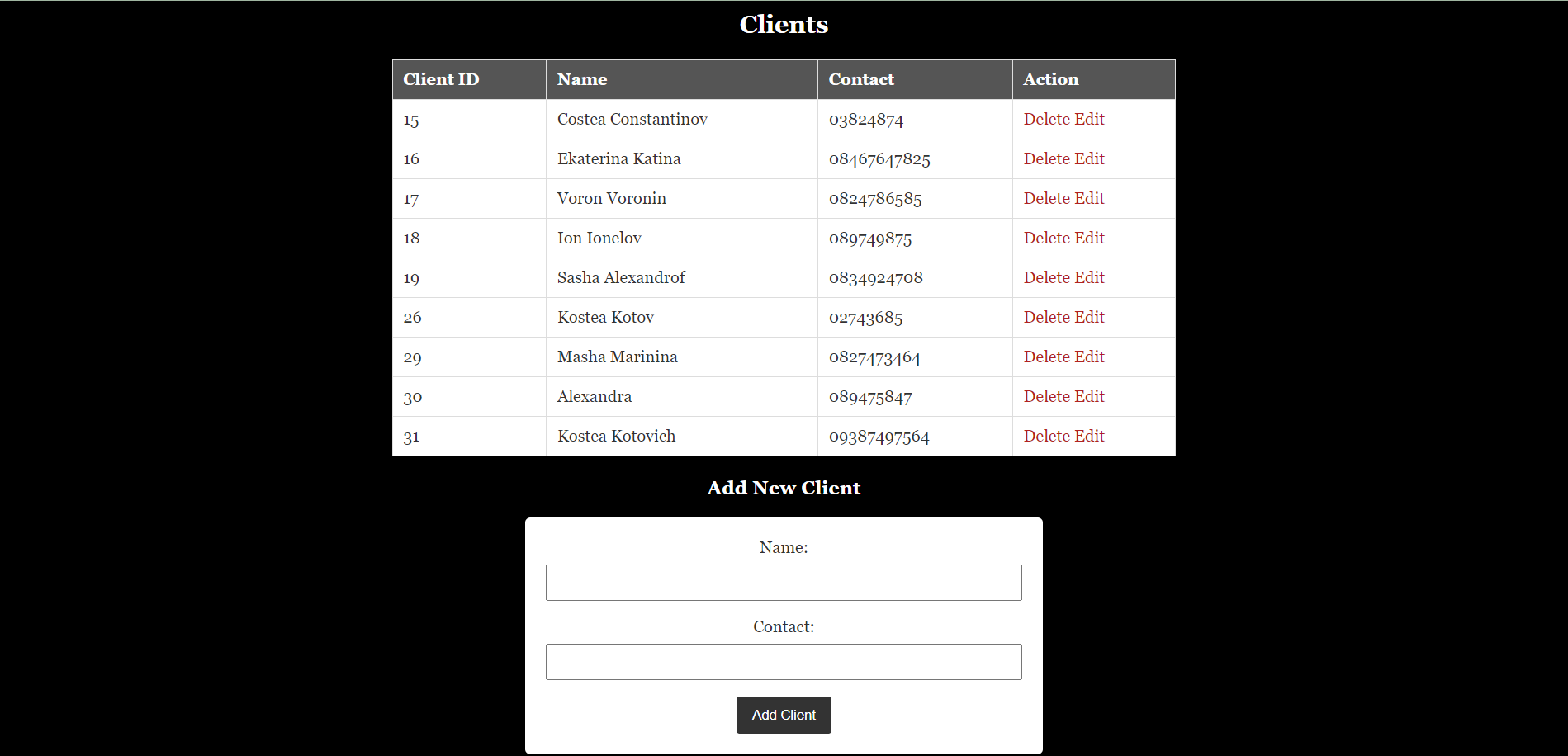
**Основные функции страницы администратора:**

1. **Просмотр и управление сессиями:**
   * Полный список фотосессий с отображением основных деталей (клиент, фотограф, дата, услуга).
   * Возможность редактирования деталей существующих сессий (обновление).
   * Удаление фотосессий, которые более не актуальны.



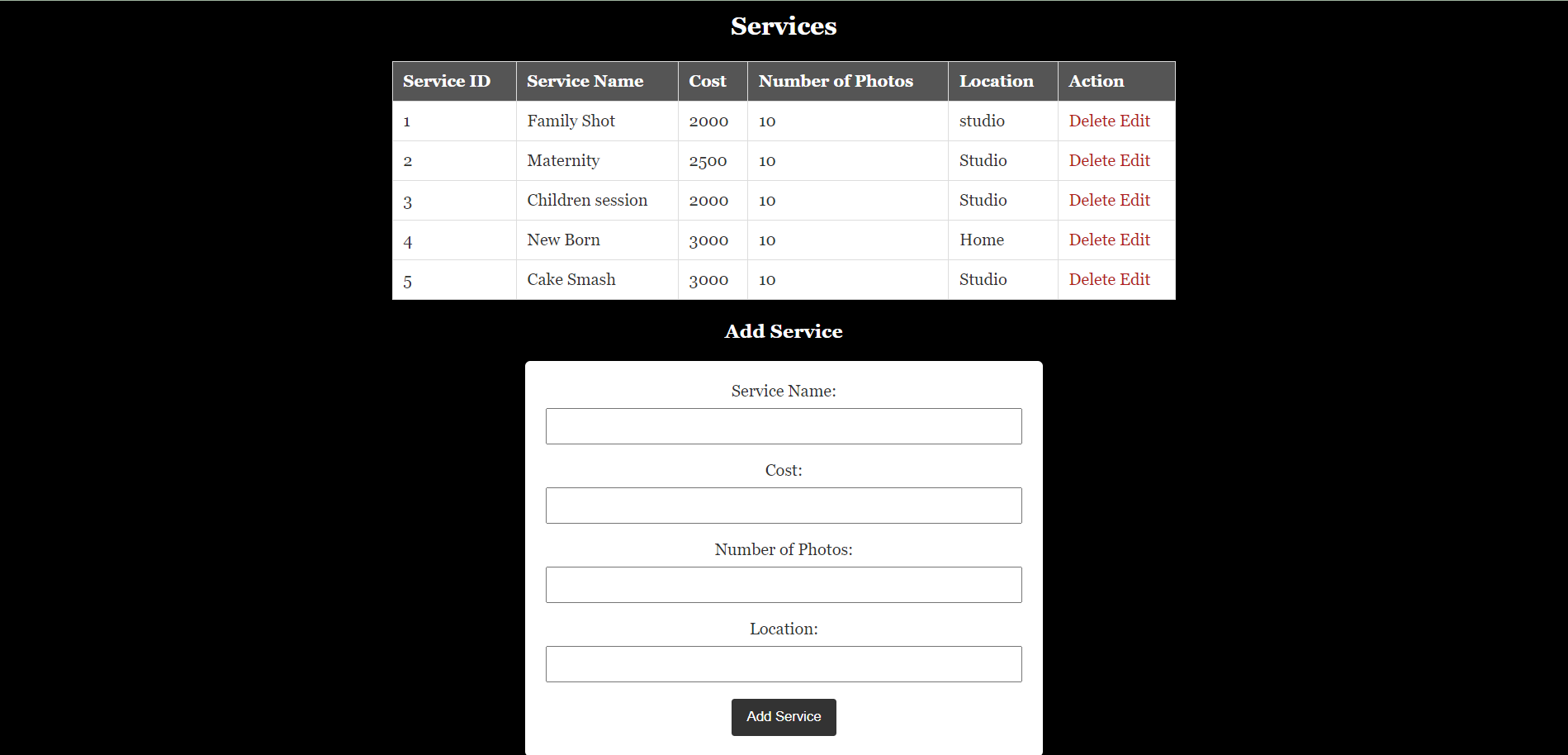
**Рис.5.7** Редактируемая таблица для хранения информации о сессиях

1. **Управление клиентами:**
   * Просмотр списка клиентов, включая основные данные (имя, контакт).
   * Возможность удаления данных о клиентах.



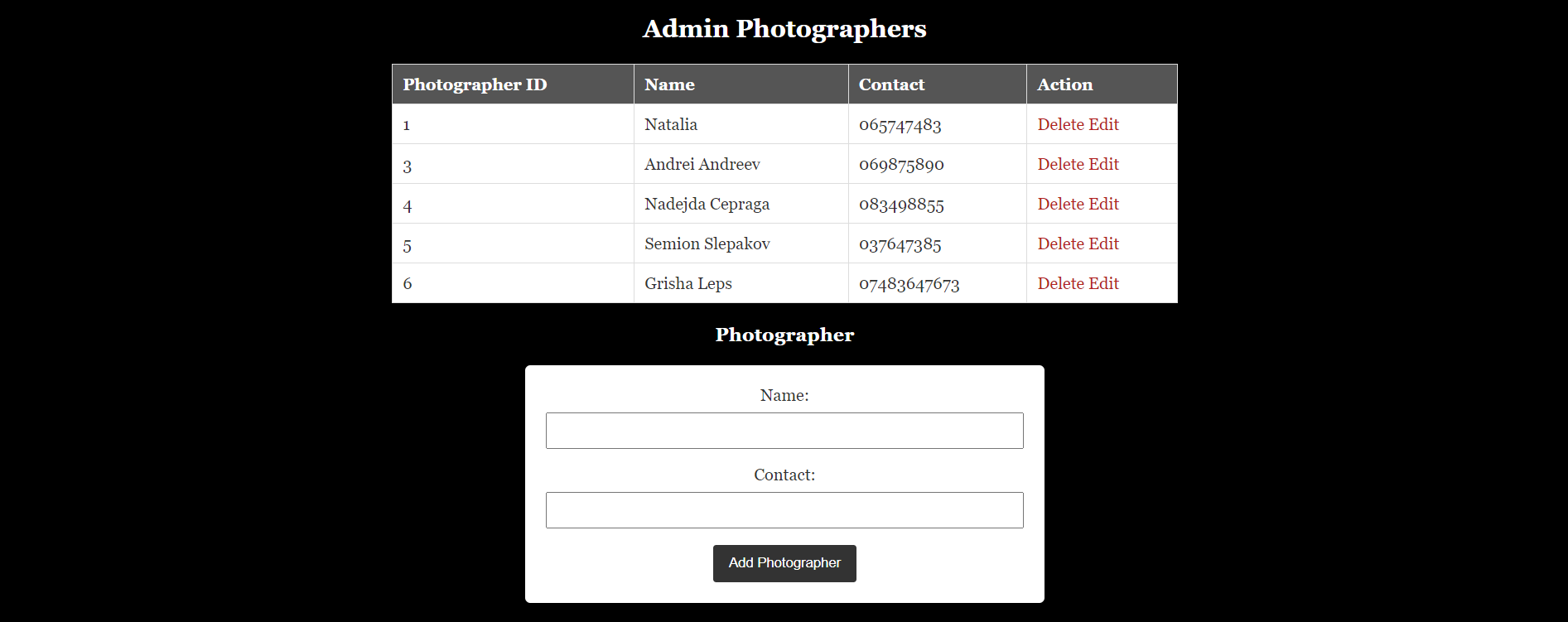
**Рис.5.8** Редактируемая таблица для хранения информации о клиентах

1. **Управление услугами:**
   * Отображение доступных услуг для фотосессий.
   * Возможность добавления новых услуг.
   * Удаление услуг, которые больше не предоставляются.



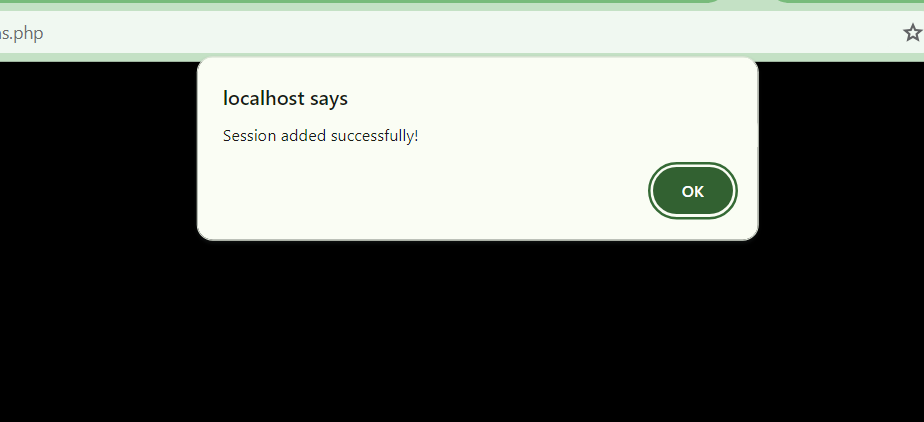
**Рис.5.9** Редактируемая таблица для хранения информации о сервисах

1. **Управление фотографами:**
   * Просмотр списка фотографов с соответствующей информацией.
   * Возможность добавления новых фотографов.
   * Удаление данных о фотографах.



**Рис.5.10** Редактируемая таблица для хранения информации о фотографах

При выполнении добавления, редактирования или удаления записи появляется всплывающее окно .



**Рис.5.11** Всплывающее окно

Top of Form

# **Глава VI (Листинг веб-приложений)**

## **PHP файлы:**

/\*studio.php\*/

<?php

/\*

\* Подключаем файл для получения соединения к базе данных (PhpMyAdmin, MySQL)

\*/

require\_once 'connect.php';

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<link rel="stylesheet" href="stylemain1.css">

<title>You+Baby Studio</title>

</head>

<body>

<header class="header">

<div class="container">

<div class="header\_\_inner">

<div class="header\_\_logo" > You+Baby Studio</div>

<nav class="nav">

<a class="nav\_\_link" href="#">Home</a>

<a class="nav\_\_link" href="details.php">Details</a>

<a class="nav\_\_link" href="booking.php">Booking</a>

<a class="nav\_\_link" href="login.php">Admin</a>

</nav>

</div>

</div>

</header>

<section class="main-section">

<div class="container">

<div class="main-section\_\_content">

<div class="main-section\_\_image">

<!-- Путь к вашему изображению -->

<img src="family.png" alt="Studio Image">

</div>

<div class="main-section\_\_text">

<h1 style="font-family: 'Georgia', sans-serif; font-size: 40px;">Welcome to <br>

You+Baby Studio</h1>

<p style="font-family: 'Georgia', sans-serif; font-size: 25px;">Capture beautiful moments with our professional photographers.</p>

<a class="booking-button" href="booking.php">Book Now</a>

</div>

</div>

</div>

</section>

</body>

</html>

/\*details.php\*/

<?php

/\*

\* Подключаем файл для получения соединения к базе данных (PhpMyAdmin, MySQL)

\*/

require\_once 'connect.php';

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<link rel="stylesheet" href="stylemain1.css">

<title>You+Baby Studio</title>

</head>

<body>

<header class="header">

<div class="container">

<div class="header\_\_inner">

<div class="header\_\_logo" > You+Baby Studio</div>

<nav class="nav">

<a class="nav\_\_link" href="#">Home</a>

<a class="nav\_\_link" href="details.php">Details</a>

<a class="nav\_\_link" href="booking.php">Booking</a>

<a class="nav\_\_link" href="login.php">Admin</a>

</nav>

</div>

</div>

</header>

<section class="main-section">

<div class="container">

<div class="main-section\_\_content">

<div class="main-section\_\_image">

<!-- Путь к вашему изображению -->

<img src="family.png" alt="Studio Image">

</div>

<div class="main-section\_\_text">

<h1 style="font-family: 'Georgia', sans-serif; font-size: 40px;">Welcome to <br>

You+Baby Studio</h1>

<p style="font-family: 'Georgia', sans-serif; font-size: 25px;">Capture beautiful moments with our professional photographers.</p>

<a class="booking-button" href="booking.php">Book Now</a>

</div>

</div>

</div>

</section>

</body>

</html>

<?php

require\_once 'connect.php';

// Получаем данные из таблицы Services

$services\_query = "SELECT \* FROM services";

$services\_result = mysqli\_query($connection, $services\_query);

// Получаем данные из таблицы Photographers

$photographers\_query = "SELECT \* FROM photographers";

$photographers\_result = mysqli\_query($connection, $photographers\_query);

// Закрываем соединение с базой данных

mysqli\_close($connection);

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Services and Photographers</title>

<link rel="stylesheet" href="styledetails1.css">

</head>

<body>

<header class="header">

<div class="container">

<div class="header\_\_inner">

<div class="header\_\_logo" > You+Baby Studio</div>

<nav class="nav">

<a class="nav\_\_link" href="studio.php">Home</a>

<a class="nav\_\_link" href="details.php">Details</a>

<a class="nav\_\_link" href="booking.php">Booking</a>

</nav>

</div>

</div>

</header>

<h2>Services</h2>

<table>

<tr>

<th>Service Name</th>

<th>Cost</th>

<th>Number of Photos</th>

<th>Location</th>

</tr>

<?php

while ($row = mysqli\_fetch\_assoc($services\_result)) {

echo "<tr>";

echo "<td>{$row['service\_name']}</td>";

echo "<td>{$row['cost']}</td>";

echo "<td>{$row['number\_photos']}</td>";

echo "<td>{$row['location']}</td>";

echo "</tr>";

}

?>

</table>

<h2>Photographers</h2>

<table>

<tr>

<th>Name</th>

<th>Photography Style</th>

<th>Contact</th>

</tr>

<?php

while ($row = mysqli\_fetch\_assoc($photographers\_result)) {

echo "<tr>";

echo "<td>{$row['name']}</td>";

echo "<td>{$row['photography\_style']}</td>";

echo "<td>{$row['contact']}</td>";

echo "</tr>";

}

?>

</table>

<div class="booking-button-container">

<a class="booking-button" href="booking.php">Book Now</a>

</div>

</body>

</html>

/\*booking.php\*/

<?php

/\*

\* Подключаем файл для получения соединения к базе данных (PhpMyAdmin, MySQL)

\*/

require\_once 'connect.php';

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Booking Form</title>

<link rel="stylesheet" href="stylebooking.css">

<!-- Подключение библиотеки jQuery для выбора даты из календаря -->

<script src="https://code.jquery.com/jquery-3.6.4.min.js"></script>

<link rel="stylesheet" href="https://code.jquery.com/ui/1.12.1/themes/base/jquery-ui.css">

<script src="https://code.jquery.com/ui/1.12.1/jquery-ui.js"></script>

<script>

$( function() {

// Инициализация календаря

$( "#datepicker" ).datepicker();

} );

</script>

</head>

<body>

<header class="header">

<div class="container">

<div class="header\_\_inner">

<div class="header\_\_logo" > You+Baby Studio</div>

<nav class="nav">

<a class="nav\_\_link" href="studio.php">Home</a>

<a class="nav\_\_link" href="details.php">Details</a>

<a class="nav\_\_link" href="booking.php">Booking</a>

</nav>

</div>

</div>

</header>

<h2>Booking Form</h2>

<form action="process\_booking.php" method="post">

<label for="client\_name">Client Name:</label>

<input type="text" name="client\_name" required>

<label for="client\_contact">Client Contact:</label>

<input type="text" name="client\_contact" required>

<label for="service\_name">Service:</label>

<select name="service\_name" required>

<!-- Здесь нужно добавить PHP-код для вывода вариантов услуг из таблицы services -->

<?php

// Подключение к базе данных (замените параметры подключения на ваши)

$connection = mysqli\_connect("localhost", "root", "", "studio");

// Проверка соединения

if (!$connection) {

die("Connection failed: " . mysqli\_connect\_error());

}

// Запрос для получения вариантов услуг из таблицы services

$service\_query = "SELECT service\_name FROM services";

$service\_result = mysqli\_query($connection, $service\_query);

// Вывод вариантов услуг в выпадающий список

while ($row = mysqli\_fetch\_assoc($service\_result)) {

echo "<option value='" . $row['service\_name'] . "'>" . $row['service\_name'] . "</option>";

}

// Закрытие соединения

mysqli\_close($connection);

?>

</select>

<label for="photographer\_name">Photographer:</label>

<select name="photographer\_name" required>

<!-- Здесь нужно добавить PHP-код для вывода вариантов фотографов из таблицы photographers -->

<?php

// Подключение к базе данных (замените параметры подключения на ваши)

$connection = mysqli\_connect("localhost", "root", "", "studio");

// Проверка соединения

if (!$connection) {

die("Connection failed: " . mysqli\_connect\_error());

}

// Запрос для получения вариантов фотографов из таблицы photographers

$photographer\_query = "SELECT name FROM photographers";

$photographer\_result = mysqli\_query($connection, $photographer\_query);

// Вывод вариантов фотографов в выпадающий список

while ($row = mysqli\_fetch\_assoc($photographer\_result)) {

echo "<option value='" . $row['name'] . "'>" . $row['name'] . "</option>";

}

// Закрытие соединения

mysqli\_close($connection);

?>

</select>

<label for="date">Date:</label>

<input type="date" name="date" min="<?php echo date('Y-m-d'); ?>" required>

<br>

<br>

<input type="submit" value="Submit">

</form>

</body>

</html>

/\*succes.php\*/

<?php

require\_once 'connect.php'; // Подключение к базе данных

// Получение информации о последней сессии клиента

$last\_session\_query = "SELECT sessions.date, services.service\_name, services.cost, services.number\_photos, services.location, photographers.name AS photographer\_name

FROM sessions

INNER JOIN services ON sessions.service\_id = services.service\_id

INNER JOIN photographers ON sessions.photographer\_id = photographers.photographer\_id

ORDER BY sessions.date DESC

LIMIT 1";

$last\_session\_result = mysqli\_query($connection, $last\_session\_query);

if ($last\_session\_result && mysqli\_num\_rows($last\_session\_result) > 0) {

$last\_session\_data = mysqli\_fetch\_assoc($last\_session\_result);

$session\_date = $last\_session\_data['date'];

$service\_name = $last\_session\_data['service\_name'];

$cost = $last\_session\_data['cost'];

$number\_photos = $last\_session\_data['number\_photos'];

$location = $last\_session\_data['location'];

$photographer\_name = $last\_session\_data['photographer\_name'];

} else {

// Обработка случая, если нет данных о последней сессии

$session\_date = "N/A";

$service\_name = "N/A";

$cost = "N/A";

$number\_photos = "N/A";

$location = "N/A";

$photographer\_name = "N/A";

}

// Закрытие соединения с базой данных

mysqli\_close($connection);

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Success Page</title>

<link rel="stylesheet" href="stylebooking.css"> <!-- Замените на ваш файл стилей -->

<style >

body {

text-align: center;

margin: 50px;

}

h2 {

color: #333;

}

.success-info {

background-color: #f8f8f8;

padding: 20px;

border-radius: 10px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

display: inline-block;

}

.info-item {

margin: 10px 0;

}

.booking-button {

display: inline-block;

background-color: #333;

color: #fff;

padding: 10px 20px;

text-decoration: none;

border-radius: 5px;

font-weight: bold;

transition: background-color 0.3s ease;

margin-top: 20px;

}

.booking-button:hover {

background-color: #555;

}

</style>

</head>

<body>

<h2>Success! Your Session Information:</h2>

<div class="success-info">

<p class="info-item"><strong>Date:</strong> <?php echo $session\_date; ?></p>

<p class="info-item"><strong>Service:</strong> <?php echo $service\_name; ?></p>

<p class="info-item"><strong>Cost:</strong> $<?php echo $cost; ?></p>

<p class="info-item"><strong>Number of Photos:</strong> <?php echo $number\_photos; ?></p>

<p class="info-item"><strong>Location:</strong> <?php echo $location; ?></p>

<p class="info-item"><strong>Photographer:</strong> <?php echo $photographer\_name; ?></p>

<p>Thank you for booking a session! Get ready to meet <?php echo $photographer\_name; ?> at <?php echo $location; ?>. Enjoy a relaxed experience, capturing <?php echo $number\_photos; ?> beautiful moments for $<?php echo $cost; ?>.</p>

<a href="studio.php" class="booking-button">Home</a>

</div>

</body>

</html>

/\*login.php\*/

<?php

session\_start();

if (isset($\_POST['login'])) {

$username = $\_POST['username'];

$password = $\_POST['password'];

// Проверка пароля (замените на свою логику проверки)

if ($username === 'admin' && $password === '5555') {

$\_SESSION['admin'] = true;

header('Location: admin\_sessions.php');

exit;

} else {

$error = "Invalid username or password";

}

}

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<style>

body {

font-family: Arial, sans-serif;

background-color: #f8f8f8;

color: #333;

margin: 0;

padding: 0;

display: flex;

flex-direction: column;

align-items: center;

justify-content: center;

height: 100vh; /\* Use 100% of the viewport height \*/

}

h2 {

color: #333;

text-align: center; /\* Center the heading \*/

margin-bottom: 20px; /\* Add some bottom margin for spacing \*/

}

form {

max-width: 300px;

width: 100%; /\* Adjust the form width if needed \*/

padding: 20px;

background-color: #fff;

border: 1px solid #ddd;

border-radius: 5px;

box-sizing: border-box;

}

label {

display: block;

margin-bottom: 10px;

}

input {

width: 100%;

padding: 8px;

margin-bottom: 15px;

box-sizing: border-box;

}

input[type="submit"] {

background-color: #000000;

color: #fff;

cursor: pointer;

}

input[type="submit"]:hover {

background-color: #000000;

}

p {

color: red;

}

</style>

</head>

<body>

<h2>Login to Your Account</h2>

<?php if (isset($error)) { echo "<p style='color:red;'>$error</p>"; } ?>

<form action="login.php" method="post">

<label for="username">Username:</label>

<input type="text" name="username" required>

<label for="password">Password:</label>

<input type="password" name="password" required>

<input type="submit" name="login" value="Login">

</form>

</body>

</html>

/\*admin\_sessions.php\*/

<?php

session\_start();

// Проверка, вошел ли пользователь в систему

if (!isset($\_SESSION['admin'])) {

header('Location: login.php');

exit;

}

// Подключение к базе данных

require\_once 'connect.php';

// Функция для очистки и защиты данных от SQL-инъекций

function sanitizeInput($input) {

global $connection;

return mysqli\_real\_escape\_string($connection, htmlspecialchars(strip\_tags($input)));

}

// Добавление новой сессии

if (isset($\_POST['add\_session'])) {

$client\_id = sanitizeInput($\_POST['client\_id']);

$photographer\_id = sanitizeInput($\_POST['photographer\_id']);

$date = sanitizeInput($\_POST['date']);

$service\_id = sanitizeInput($\_POST['service\_id']);

$insert\_query = "INSERT INTO sessions (client\_id, photographer\_id, date, service\_id, payment\_id, photo\_id)

VALUES ('$client\_id', '$photographer\_id', '$date', '$service\_id', 1, 1)";

if (mysqli\_query($connection, $insert\_query)) {

echo "<script>alert('Session added successfully!');</script>";

} else {

echo "Error adding session: " . mysqli\_error($connection);

}

}

// Удаление сессии

if (isset($\_GET['delete\_session'])) {

$session\_id = sanitizeInput($\_GET['delete\_session']);

$delete\_query = "DELETE FROM sessions WHERE session\_id = '$session\_id'";

if (mysqli\_query($connection, $delete\_query)) {

echo "<script>alert('Session deleted successfully!');</script>";

} else {

echo "Error deleting session: " . mysqli\_error($connection);

}

}

/// Обработка редактирования сессии

if (isset($\_POST['update\_session'])) {

$edit\_session\_id = sanitizeInput($\_POST['edit\_session\_id']);

$edit\_client\_id = sanitizeInput($\_POST['edit\_client\_id']);

$edit\_photographer\_id = sanitizeInput($\_POST['edit\_photographer\_id']);

$edit\_date = sanitizeInput($\_POST['edit\_date']);

$edit\_service\_id = sanitizeInput($\_POST['edit\_service\_id']);

$update\_query = "UPDATE sessions

SET client\_id = '$edit\_client\_id',

photographer\_id = '$edit\_photographer\_id',

date = '$edit\_date',

service\_id = '$edit\_service\_id'

WHERE session\_id = '$edit\_session\_id'";

if (mysqli\_query($connection, $update\_query)) {

echo "<script>alert('Session updated successfully!');</script>";

} else {

echo "Error updating session: " . mysqli\_error($connection);

}

}

// Получение всех сессий

$sessions\_query = "SELECT \* FROM sessions";

$sessions\_result = mysqli\_query($connection, $sessions\_query);

// Добавление нового клиента

if (isset($\_POST['add\_client'])) {

$client\_name = sanitizeInput($\_POST['client\_name']);

$client\_contact = sanitizeInput($\_POST['client\_contact']);

$insert\_query = "INSERT INTO clients (name, contact) VALUES ('$client\_name', '$client\_contact')";

if (mysqli\_query($connection, $insert\_query)) {

echo "<script>alert('Client added successfully!');</script>";

} else {

echo "Error adding client: " . mysqli\_error($connection);

}

}

// Удаление клиента

if (isset($\_GET['delete\_client'])) {

$client\_id = sanitizeInput($\_GET['delete\_client']);

$delete\_query = "DELETE FROM clients WHERE client\_id = '$client\_id'";

if (mysqli\_query($connection, $delete\_query)) {

echo "<script>alert('Client deleted successfully!');</script>";

} else {

echo "Error deleting client: " . mysqli\_error($connection);

}

}

// Обновление клиента

if (isset($\_POST['update\_client'])) {

$client\_id = sanitizeInput($\_POST['client\_id']);

$client\_name = sanitizeInput($\_POST['client\_name']);

$client\_contact = sanitizeInput($\_POST['client\_contact']);

$update\_query = "UPDATE clients SET name = '$client\_name', contact = '$client\_contact' WHERE client\_id = '$client\_id'";

if (mysqli\_query($connection, $update\_query)) {

echo "<script>alert('Client updated successfully!');</script>";

} else {

echo "Error updating client: " . mysqli\_error($connection);

}

}

// Получение всех клиентов

$clients\_query = "SELECT \* FROM clients";

$clients\_result = mysqli\_query($connection, $clients\_query);

// Добавление нового фотографа

if (isset($\_POST['add\_photographer'])) {

$photographer\_name = sanitizeInput($\_POST['photographer\_name']);

$photographer\_contact = sanitizeInput($\_POST['photographer\_contact']);

$insert\_query = "INSERT INTO photographers (name, contact) VALUES ('$photographer\_name', '$photographer\_contact')";

if (mysqli\_query($connection, $insert\_query)) {

echo "<script>alert('Photographer added successfully!');</script>";

} else {

echo "Error adding photographer: " . mysqli\_error($connection);

}

}

// Удаление фотографа

if (isset($\_GET['delete\_photographer'])) {

$photographer\_id = sanitizeInput($\_GET['delete\_photographer']);

$delete\_query = "DELETE FROM photographers WHERE photographer\_id = '$photographer\_id'";

if (mysqli\_query($connection, $delete\_query)) {

echo "<script>alert('Photographer deleted successfully!');</script>";

} else {

echo "Error deleting photographer: " . mysqli\_error($connection);

}

}

// Обновление фотографа

if (isset($\_POST['update\_photographer'])) {

$photographer\_id = sanitizeInput($\_POST['photographer\_id']);

$photographer\_name = sanitizeInput($\_POST['photographer\_name']);

$photographer\_contact = sanitizeInput($\_POST['photographer\_contact']);

$update\_query = "UPDATE photographers SET name = '$photographer\_name', contact = '$photographer\_contact' WHERE photographer\_id = '$photographer\_id'";

if (mysqli\_query($connection, $update\_query)) {

echo "<script>alert('Photographer updated successfully!');</script>";

} else {

echo "Error updating photographer: " . mysqli\_error($connection);

}

}

// Получение всех фотографов

$photographers\_query = "SELECT \* FROM photographers";

$photographers\_result = mysqli\_query($connection, $photographers\_query);

// Добавление новой услуги

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

$service\_name = sanitizeInput($\_POST['service\_name']);

$insert\_query = "INSERT INTO services (service\_name) VALUES ('$service\_name')";

if (mysqli\_query($connection, $insert\_query)) {

echo "<script>alert('Service added successfully!');</script>";

} else {

echo "Error adding service: " . mysqli\_error($connection);

}

}

// Удаление услуги

if (isset($\_GET['delete\_service'])) {

$service\_id = sanitizeInput($\_GET['delete\_service']);

$delete\_query = "DELETE FROM services WHERE service\_id = '$service\_id'";

if (mysqli\_query($connection, $delete\_query)) {

echo "<script>alert('Service deleted successfully!');</script>";

} else {

echo "Error deleting service: " . mysqli\_error($connection);

}

}

// Обработка редактирования услуги

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

$edit\_service\_id = sanitizeInput($\_POST['edit\_service\_id']);

$edit\_service\_name = sanitizeInput($\_POST['edit\_service\_name']);

$update\_service\_query = "UPDATE services

SET service\_name = '$edit\_service\_name'

WHERE service\_id = '$edit\_service\_id'";

if (mysqli\_query($connection, $update\_service\_query)) {

echo "<script>alert('Service updated successfully!');</script>";

} else {

echo "Error updating service: " . mysqli\_error($connection);

}

}

// Получение всех услуг

$services\_query = "SELECT \* FROM services";

$services\_result = mysqli\_query($connection, $services\_query);

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Admin Sessions</title>

<link rel="stylesheet" href="styledetails1.css">

<style>

body {

background-color: #000000

}

.header {

background-color: #000000;

}

h2 {

color: #fff;

text-align: center;

}

h3 {

color: #fff;

text-align: center;

}

a.delete-btn, a.edit-btn {

color: #B22222;

text-decoration: none; /

}

a.delete-btn:hover, a.edit-btn:hover {

color: #FF0000;

}

</style>

<script>

// Функция открытия модального окна редактирования сессии

function openEditSessionModal(sessionId) {

// Ваш код для получения данных о сессии по sessionId и заполнения формы

// Например, можно использовать AJAX для отправки запроса на сервер

// и получения данных о сессии в формате JSON

// После получения данных, заполните форму редактирования

// Пример:

document.getElementById('edit\_client\_id').value = 'значение\_клиента';

// Заполните аналогичным образом остальные поля формы

// Открываем модальное окно

document.getElementById('editSessionModal').style.display = 'block';

}

// Функция закрытия модального окна редактирования сессии

function closeEditSessionModal() {

document.getElementById('editSessionModal').style.display = 'none';

}

</script>

</head>

<body>

<header class="header">

<div class="container">

<div class="header\_\_inner">

<div class="header\_\_logo" > You+Baby Studio</div>

<nav class="nav">

<a class="nav\_\_link" href="#sessions">Sessions</a>

<a class="nav\_\_link" href="#clients">Clients</a>

<a class="nav\_\_link" href="#photographers">Photographers</a>

<a class="nav\_\_link" href="#services">Services</a>

<a class="nav\_\_link" href="studio.php">Exit</a>

</nav>

</div>

</div>

</header>

<section id="sessions">

<h2>Sessions</h2>

<table border="1">

<tr>

<th>Session ID</th>

<th>Client Name</th>

<th>Photographer Name</th>

<th>Date</th>

<th>Service</th>

<th>Action</th>

</tr>

<?php

$sessions\_query = "SELECT sessions.session\_id, clients.name AS client\_name, photographers.name AS photographer\_name, sessions.date, services.service\_name

FROM sessions

INNER JOIN clients ON sessions.client\_id = clients.client\_id

INNER JOIN photographers ON sessions.photographer\_id = photographers.photographer\_id

INNER JOIN services ON sessions.service\_id = services.service\_id";

$sessions\_result = mysqli\_query($connection, $sessions\_query);

$sessions\_result = mysqli\_query($connection, $sessions\_query);

while ($row = mysqli\_fetch\_assoc($sessions\_result)) { ?>

<tr>

<td><?php echo $row['session\_id']; ?></td>

<td><?php echo $row['client\_name']; ?></td>

<td><?php echo $row['photographer\_name']; ?></td>

<td><?php echo $row['date']; ?></td>

<td><?php echo $row['service\_name']; ?></td>

<td>

<a class="delete-btn" href="admin\_sessions.php?delete\_session=<?php echo $row['session\_id']; ?>">Delete</a>

<a class="edit-btn" href="#" onclick="openEditSessionModal(<?php echo $row['session\_id']; ?>)">Edit</a>

</td>

</tr>

<?php } ?>

</table>

<h3>Add New Session</h3>

<form action="admin\_sessions.php" method="post">

<label for="client\_id">Client:</label>

<select name="client\_id" required>

<?php

$clients\_query = "SELECT \* FROM clients";

$clients\_result = mysqli\_query($connection, $clients\_query);

while ($client\_row = mysqli\_fetch\_assoc($clients\_result)) {

echo "<option value='{$client\_row['client\_id']}'>{$client\_row['name']}</option>";

}

?>

</select>

<label for="photographer\_id">Photographer:</label>

<select name="photographer\_id" required>

<?php

$photographers\_query = "SELECT \* FROM photographers";

$photographers\_result = mysqli\_query($connection, $photographers\_query);

while ($photographer\_row = mysqli\_fetch\_assoc($photographers\_result)) {

echo "<option value='{$photographer\_row['photographer\_id']}'>{$photographer\_row['name']}</option>";

}

?>

</select>

<label for="date">Date:</label>

<input type="text" name="date" required>

<label for="service\_id">Service:</label>

<select name="service\_id" required>

<?php

$services\_query = "SELECT \* FROM services";

$services\_result = mysqli\_query($connection, $services\_query);

while ($service\_row = mysqli\_fetch\_assoc($services\_result)) {

echo "<option value='{$service\_row['service\_id']}'>{$service\_row['service\_name']}</option>";

}

?>

</select>

<input type="submit" name="add\_session" value="Add Session">

</form>

<div id="editSessionModal" class="modal">

<div class="modal-content">

<span class="close" onclick="closeEditSessionModal()">&times;</span>

<h3>Edit Session</h3>

<form id="editSessionForm" action="admin\_sessions.php" method="post">

<!-- Поле для редактирования клиента -->

<label for="edit\_client\_id">Client:</label>

<select name="edit\_client\_id" id="edit\_client\_id" required>

<?php

// Получение клиентов

$clients\_query = "SELECT \* FROM clients";

$clients\_result = mysqli\_query($connection, $clients\_query);

while ($client\_row = mysqli\_fetch\_assoc($clients\_result)) {

$selected = ($client\_row['client\_id'] == $row['client\_id']) ? 'selected' : '';

echo "<option value='{$client\_row['client\_id']}' $selected>{$client\_row['name']}</option>";

}

?>

</select>

<!-- Поле для редактирования фотографа -->

<label for="edit\_photographer\_id">Photographer:</label>

<select name="edit\_photographer\_id" id="edit\_photographer\_id" required>

<?php

// Получение фотографов

$photographers\_query = "SELECT \* FROM photographers";

$photographers\_result = mysqli\_query($connection, $photographers\_query);

while ($photographer\_row = mysqli\_fetch\_assoc($photographers\_result)) {

$selected = ($photographer\_row['photographer\_id'] == $row['photographer\_id']) ? 'selected' : '';

echo "<option value='{$photographer\_row['photographer\_id']}' $selected>{$photographer\_row['name']}</option>";

}

?>

</select>

<!-- Поле для редактирования даты -->

<label for="edit\_date">Date:</label>

<input type="text" name="edit\_date" id="edit\_date" value="<?php echo isset($row['date']) ? $row['date'] : ''; ?>" required> <br />

<!-- Поле для редактирования услуги -->

<label for="edit\_service\_id">Service:</label>

<select name="edit\_service\_id" id="edit\_service\_id" required>

<?php

// Получение услуг

$services\_query = "SELECT \* FROM services";

$services\_result = mysqli\_query($connection, $services\_query);

while ($service\_row = mysqli\_fetch\_assoc($services\_result)) {

$selected = ($service\_row['service\_id'] == $row['service\_id']) ? 'selected' : '';

echo "<option value='{$service\_row['service\_id']}' $selected>{$service\_row['service\_name']}</option>";

}

?>

</select>

<input type="hidden" name="edit\_session\_id" value="<?php echo $row['session\_id']; ?>">

<input type="submit" name="update\_session" value="Update Session">

</form>

</div>

</div>

</section>

<section id="clients">

<h2>Clients</h2>

<table border="1">

<tr>

<th>Client ID</th>

<th>Name</th>

<th>Contact</th>

<th>Action</th>

</tr>

<?php

mysqli\_data\_seek($clients\_result, 0);

while ($row = mysqli\_fetch\_assoc($clients\_result)) { ?>

<tr>

<td><?php echo $row['client\_id']; ?></td>

<td><?php echo $row['name']; ?></td>

<td><?php echo $row['contact']; ?></td>

<td>

<a class="delete-btn" href="admin\_sessions.php?delete\_client=<?php echo $row['client\_id']; ?>">Delete</a>

<a class="edit-btn" href="edit\_client.php?client\_id=<?php echo $row['client\_id']; ?>">Edit</a>

</td>

</tr>

<?php } ?>

</table>

<h3>Add New Client</h3>

<form action="admin\_sessions.php" method="post">

<label for="client\_name">Name:</label>

<input type="text" name="client\_name" required>

<label for="client\_contact">Contact:</label>

<input type="text" name="client\_contact" required>

<br>

<input type="submit" name="add\_client" value="Add Client">

</form>

</section>

<section id="photographers">

<h2>Admin Photographers</h2>

<table border="1">

<tr>

<th>Photographer ID</th>

<th>Name</th>

<th>Contact</th>

<th>Action</th>

</tr>

<?php

mysqli\_data\_seek($photographers\_result, 0);

while ($row = mysqli\_fetch\_assoc($photographers\_result)) { ?>

<tr>

<td><?php echo $row['photographer\_id']; ?></td>

<td><?php echo $row['name']; ?></td>

<td><?php echo $row['contact']; ?></td>

<td>

<a class="delete-btn" href="admin\_sessions.php?delete\_photographer=<?php echo $row['photographer\_id']; ?>">Delete</a>

<a class="edit-btn" href="edit\_photographer.php?photographer\_id=<?php echo $row['photographer\_id']; ?>">Edit</a>

</td>

</tr>

<?php } ?>

</table>

<h3>Photographer</h3>

<form action="admin\_sessions.php" method="post">

<label for="photographer\_name">Name:</label>

<input type="text" name="photographer\_name" required>

<label for="photographer\_contact">Contact:</label>

<input type="text" name="photographer\_contact" required>

<br>

<input type="submit" name="add\_photographer" value="Add Photographer">

</form>

</section>

<section id="services">

<h2>Services</h2>

<table border="1">

<tr>

<th>Service ID</th>

<th>Service Name</th>

<th>Cost</th>

<th>Number of Photos</th>

<th>Location</th>

<th>Action</th>

</tr>

<?php

mysqli\_data\_seek($services\_result, 0);

while ($row = mysqli\_fetch\_assoc($services\_result)) { ?>

<tr>

<td><?php echo $row['service\_id']; ?></td>

<td><?php echo $row['service\_name']; ?></td>

<td><?php echo $row['cost']; ?></td>

<td><?php echo $row['number\_photos']; ?></td>

<td><?php echo $row['location']; ?></td>

<td>

<a class="delete-btn" href="admin\_sessions.php?delete\_service=<?php echo $row['service\_id']; ?>">Delete</a>

<a class="edit-btn" href="edit\_service.php?service\_id=<?php echo $row['service\_id']; ?>">Edit</a>

</td>

</tr>

<?php } ?>

</table>

<h3>Add Service</h3>

<form action="admin\_sessions.php" method="post">

<label for="service\_name">Service Name:</label>

<input type="text" name="service\_name" required>

<label for="cost">Cost:</label>

<input type="text" name="cost" required>

<label for="number\_photos">Number of Photos:</label>

<input type="text" name="number\_photos" required>

<label for="location">Location:</label>

<input type="text" name="location" required>

<br>

<input type="submit" name="add\_service" value="Add Service">

</form>

</section>

<a href="logout.php">Logout</a>

</body>

</html>

<?php

// Закрытие соединения с базой данных

mysqli\_close($connection);

?>

/\*process\_booking.php\*/

<?php

require\_once 'connect.php';

// Получение данных из формы

$client\_name = $\_POST['client\_name'];

$client\_contact = $\_POST['client\_contact'];

$service\_name = $\_POST['service\_name'];

$session\_date = $\_POST['date'];

$photographer\_name = $\_POST['photographer\_name'];

// Вставляем данные в таблицу clients и получаем последний вставленный client\_id

$insert\_client\_query = "INSERT INTO clients (name, contact) VALUES ('$client\_name', '$client\_contact')";

mysqli\_query($connection, $insert\_client\_query);

$client\_id = mysqli\_insert\_id($connection);

// Получение service\_id по имени услуги

$service\_query = "SELECT service\_id FROM services WHERE service\_name = '$service\_name'";

$service\_result = mysqli\_query($connection, $service\_query);

$row = mysqli\_fetch\_assoc($service\_result);

$service\_id = $row['service\_id'];

// Получение photographer\_id по имени фотографа

$photographer\_query = "SELECT photographer\_id FROM photographers WHERE name = '$photographer\_name'";

$photographer\_result = mysqli\_query($connection, $photographer\_query);

$row = mysqli\_fetch\_assoc($photographer\_result);

$photographer\_id = $row['photographer\_id'];

// Вставка данных в таблицу sessions

$insert\_query = "INSERT INTO sessions ( client\_id, photographer\_id, date, service\_id, payment\_id, photo\_id)

VALUES ( '$client\_id', '$photographer\_id', '$session\_date', '$service\_id', 1, 1)";

if (mysqli\_query($connection, $insert\_query)) {

// Вывод всплывающего окна с сообщением

echo "<script>alert('Session added successfully!');</script>";

// После успешной записи, установите session\_id и перенаправьте на страницу успеха

$\_SESSION['client\_id'] = $client\_id;

header('Location: success.php');

exit;

}

else {

echo "Error: " . mysqli\_error($connection);

}

// Закрытие соединения

mysqli\_close($connection);

?>

/\*connect.php\*/

<?php

$host = "localhost"; // Имя хоста

$username = "root"; // Имя пользователя

$password = ""; // Пароль

$database = "studio"; // Имя базы данных

// Подключение к базе данных

$connection = mysqli\_connect($host, $username, $password, $database);

// Проверка соединения

if (!$connection) {

die("Ошибка подключения к базе данных: " . mysqli\_connect\_error());

}

## **CSS файлы:**

/\* stylebooking.css\*/

body {

font-family: Georgia, sans-serif;

background-color: #f2f2f2;

color: #333;

margin: 0;

padding: 0;

}

.header {

background-color: #333;

color: #fff;

padding: 20px 0;

}

.container {

max-width: 1200px;

margin: 0 auto;

}

.header\_\_inner {

display: flex;

justify-content: space-between;

align-items: center;

}

.header\_\_logo {

font-size: 24px;

font-weight: bold;

}

.nav {

display: flex;

}

.nav\_\_link {

color: #fff;

text-decoration: none;

margin-left: 20px;

}

.nav\_\_link:hover {

text-decoration: underline;

}

body {

font-family:Georgia , sans-serif;

background-color: #f2f2f2;

color: #333;

margin: 0;

}

.container {

max-width: 800px;

margin: 0 auto;

}

h2 {

color: #333;

text-align: center;

}

form {

background-color: #fff;

padding: 20px;

border-radius: 5px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

width: 33%; /\* Set the width to one-third of the parent container \*/

margin: 0 auto; /\* Center the form horizontally \*/

box-sizing: border-box;

text-align: center;

}

label {

display: block;

margin-bottom: 8px;

color: #333;

}

input[type="text"],

select {

width: 100%;

padding: 8px;

margin-bottom: 16px;

box-sizing: border-box;

}

select {

appearance: none;

-webkit-appearance: none;

-moz-appearance: none;

background: url('arrow-down.png') no-repeat right center; /\* замените 'arrow-down.png' на вашу стрелку \*/

padding-right: 30px;

}

input[type="submit"] {

background-color: #333;

color: #fff;

padding: 10px 15px;

border: none;

border-radius: 3px;

cursor: pointer;

display: inline-block;

}

input[type="submit"]:hover {

background-color: #555;

}

#date {

width: calc(100% - 16px); /\* учитывайте padding в ширине \*/

padding: 8px;

}

/\* stylemain1.css\*/

body {

font-family: Georgia, sans-serif;

margin: 0;

padding: 0;

}

.header {

background-color: #333;

color: #fff;

padding: 30px 0;

}

.container {

max-width: 1200px;

margin: 0 auto;

}

.header\_\_inner {

display: flex;

justify-content: space-between;

align-items: center;

}

.header\_\_logo {

font-size: 28px;

font-weight: bold;

}

.nav {

display: flex;

}

.nav\_\_link {

color: #fff;

text-decoration: none;

margin-left: 20px;

}

.nav\_\_link:hover {

text-decoration: underline;

}

.main-section {

background-color: #fff;

padding: 30px 0;

}

.main-section\_\_content {

display: flex;

justify-content: space-between;

align-items: center;

}

.main-section\_\_image img {

max-width: 100%;

height: auto;

border-radius: 10px;

}

.main-section\_\_text {

width: 50%;

}

.booking-button {

display: inline-block;

background-color: #333;

color: #fff;

padding: 10px 20px;

text-decoration: none;

border-radius: 5px;

font-weight: bold;

transition: background-color 0.3s ease;

}

.booking-button:hover {

background-color: #555;

}

# **Выводы**

При выполнении данной работы мной были успешно применены различные технологии и методы для эффективной обработки, организации и манипуляции данными. Использование инструментов, таких как AJAX, jQuery, MySQL, PHPMyAdmin, PHP, CSS и HTML, совместно способствовало созданию полностью функционального проекта, способного эффективно управлять данными на входе, выходе и хранить информацию для различных нужд, таких как статистика и вычисления.

Важно отметить, что, подобно любому другому проекту, данное веб-приложение также прошло через различные этапы разработки, начиная с проектирования и создания логических связей между таблицами и заканчивая формированием графических элементов и ограничений с использованием асинхронных сценариев.

Неизбежно любой проект может содержать некоторое количество ошибок или недоразумений, но в целом он стабилен и позволяет пользователям, гостям и администраторам получать услуги и хранить необходимые данные для эффективного управления бизнесом.

Выполнение поставленных задач позволило мне повысить свой профессиональный уровень, взаимодействуя с механизмами системы управления базами данных (СУБД) и создавая различные виды запросов. Применение методов передачи данных и их предварительной обработки перед передачей на сервер, с целью ограничения нерационального использования базы данных, также было успешно осуществлено. Были также использованы методы защиты данных, включая шифрование (md5) и удаление суперглобальной переменной SESSION, содержащей личные данные пользователей. Полученный опыт и уровень навыков предоставят мне возможность успешно развиваться и расширять свои знания в данной области.

# **Библиография**

1. W3Schools. (<https://www.w3schools.com/>)
   * Интернет-ресурс с обширными учебными материалами по веб-технологиям, таким как HTML, CSS, JavaScript, PHP, и другие.
2. PHP: Hypertext Preprocessor. (<https://www.php.net/>)
   * Официальный сайт PHP, предоставляющий документацию, руководства и ресурсы для разработчиков.
3. jQuery. (<https://jquery.com/>)
   * Сайт jQuery, содержащий документацию и ресурсы для использования библиотеки jQuery в веб-разработке.
4. MySQL Documentation. (https://dev.mysql.com/doc/)
   * Официальная документация MySQL с руководствами и ресурсами для работы с системой управления базами данных MySQL.
5. MDN Web Docs. (https://developer.mozilla.org/)
   * Ресурс от Mozilla с обширной документацией по веб-технологиям, включая HTML, CSS, JavaScript и другие.
6. Stack Overflow. (<https://stackoverflow.com/>)
   * Сайт вопросов и ответов, где разработчики обсуждают проблемы и делятся опытом в веб-разработке.
7. GitHub. (<https://github.com/>)
   * Платформа для хранения кода и совместной разработки. Репозитории на GitHub могут содержать исходный код веб-проектов.