**Отчет**

Задание: Создать Web-приложение с базой данных, в котором обязательно должны присутствовать формы заполнения/редактирования/удаления записей из БД, просмотра записей, удовлетворяющих некоторому признаку в соответствии с тематикой Фото мастерская: заказчики работ, прайс работ, журнал поступления заказов, исполнители.

Решение:

Сначала создадим базу данных для хранения записей. Для этого построим схему отношений:

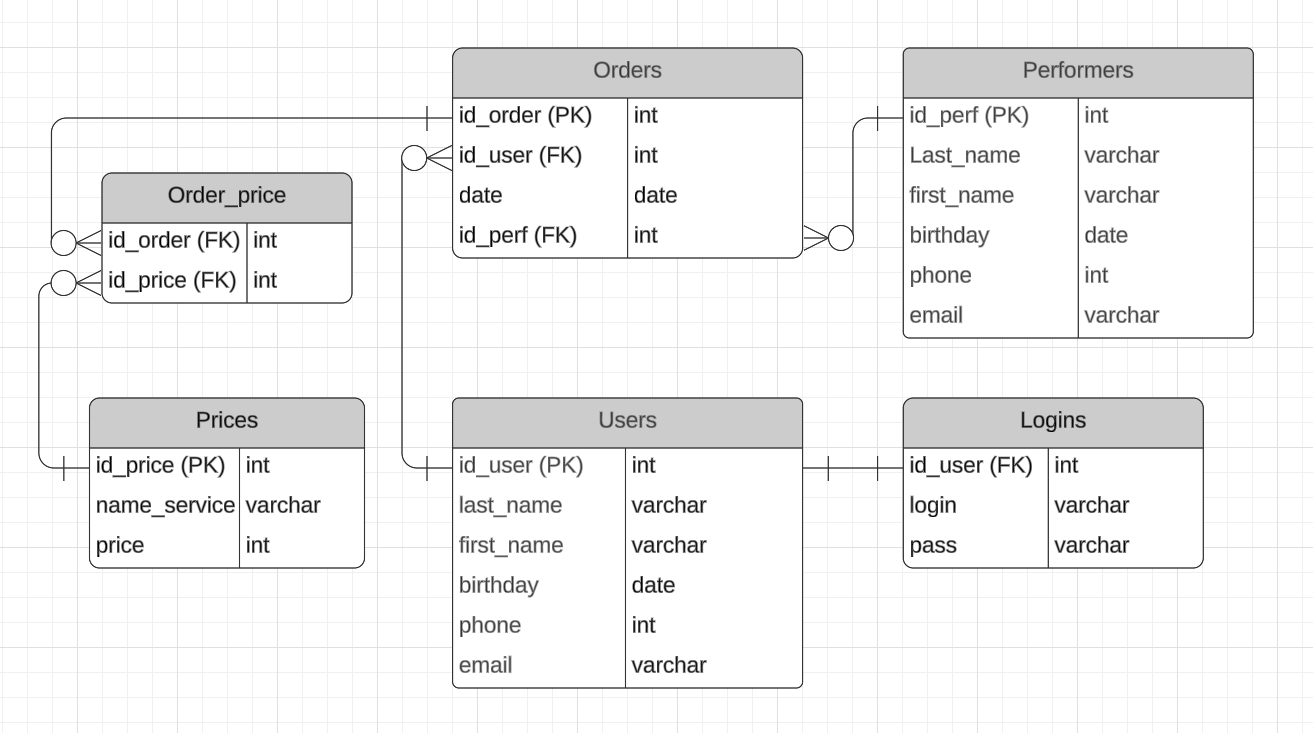


Рисунок 1 – Er-диаграмма.

где таблица Users содержит информацию о заказчиках работ, а именно уникальный идентификатор заказчика, фамилию, имя, дату рождения, номер телефона и Email. Logins содержит информацию об уникальных логинах и паролях заказчиков. Performers – информацию об исполнителях, а именно уникальный идентификатор исполнителя, фамилию, имя, дату рождения, номер телефона и Email. Prices содержит прайс работ, а именно уникальный идентификатор услуги, название услуги и цену услуги. Orders – журнал поступления заказов, содержит в себе уникальный идентификатор заказа, номер заказчика, дату создания заказа и номер исполнителя. Order\_price связывает таблицы Orders и Prices и содержит информацию о номере заказа и услуги.

Для создания базы данных используем выделенный учебный сервер и настроенную на нем базу данных MariaDB.

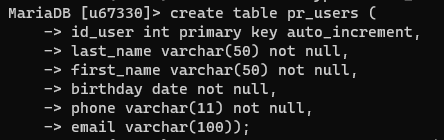


Рисунок 2 – Создание таблицы Users

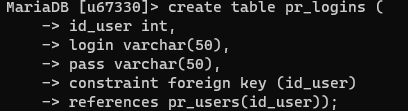


Рисунок 3 – Создание таблицы Logins

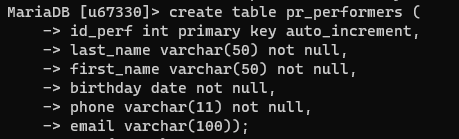


Рисунок 4 – Создание таблицы Performers

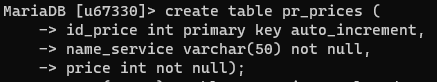


Рисунок 5 – Создание таблицы Prices

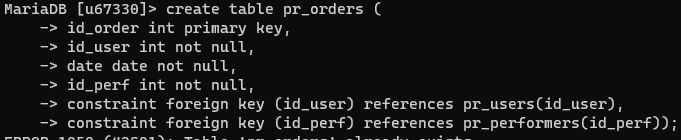


Рисунок 6 – Создание таблицы Orders

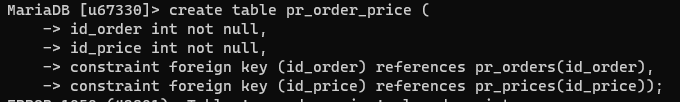


Рисунок 7 – Создание таблицы Order\_price

Для создания Web-приложения будем использовать язык PHP.

Создадим главную страницу index.php. Определим форму для внесения данных в таблицу Users. При вводе заказчиком данных и сохранении формы, значения проходят валидацию. При наличии ошибок выдается сообщение об ошибке, если ошибок нет - сохраненные значения запишутся в таблицу Users, а заказчику выдадутся уникальный логин и пароль и сохранятся в таблицу Logins.

Создадим страницу login.php, на которой заказчик может ввести ранее выданные логин и пароль для изменения ранее введенных значений. Определим форму из двух полей: логин и пароль. При сохранении формы проверяется, есть ли такой логин и правильно ли введен пароль. Если ошибок нет, то в сессию сохраняем логин и уникальный идентификатор заказчика и перенаправляем на главную страницу. На главной странице добавим кнопку «Вход», при нажатии которой происходит переадресация на login.php.

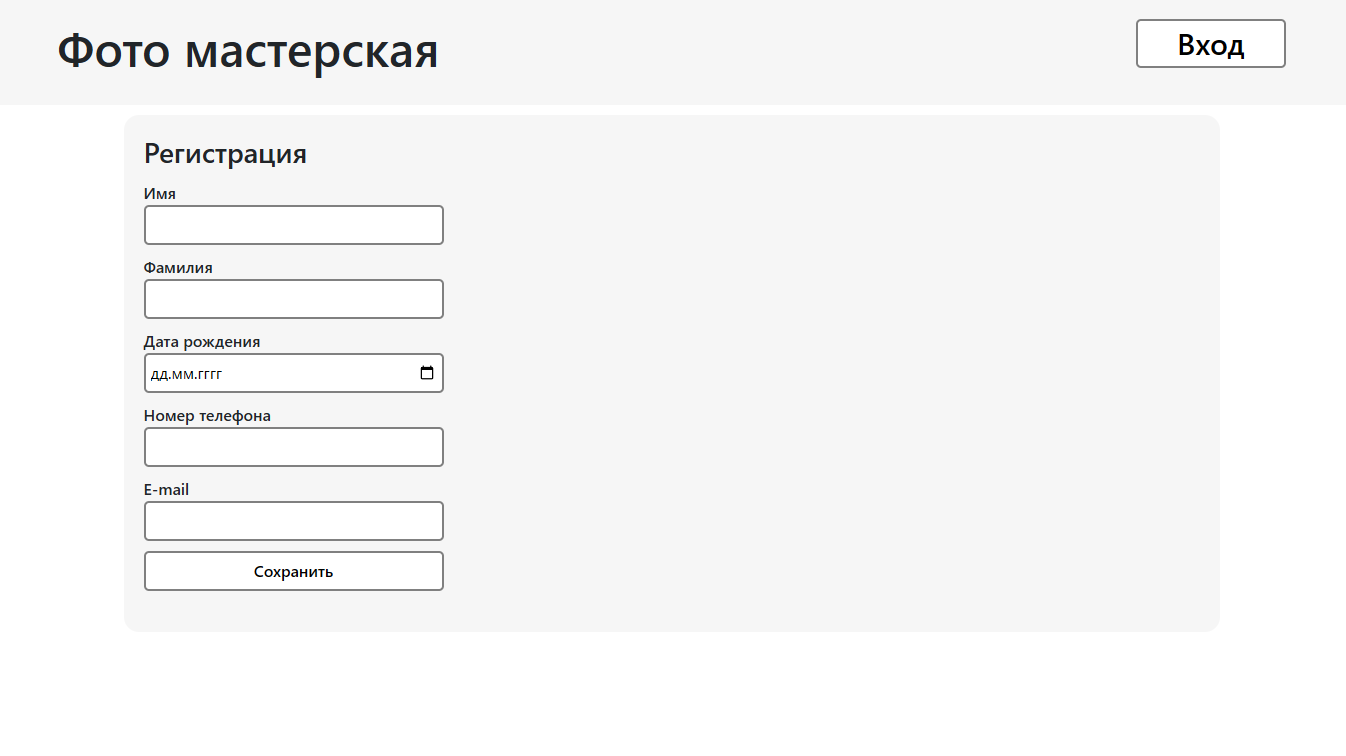


Рисунок 8 – Страница index.php

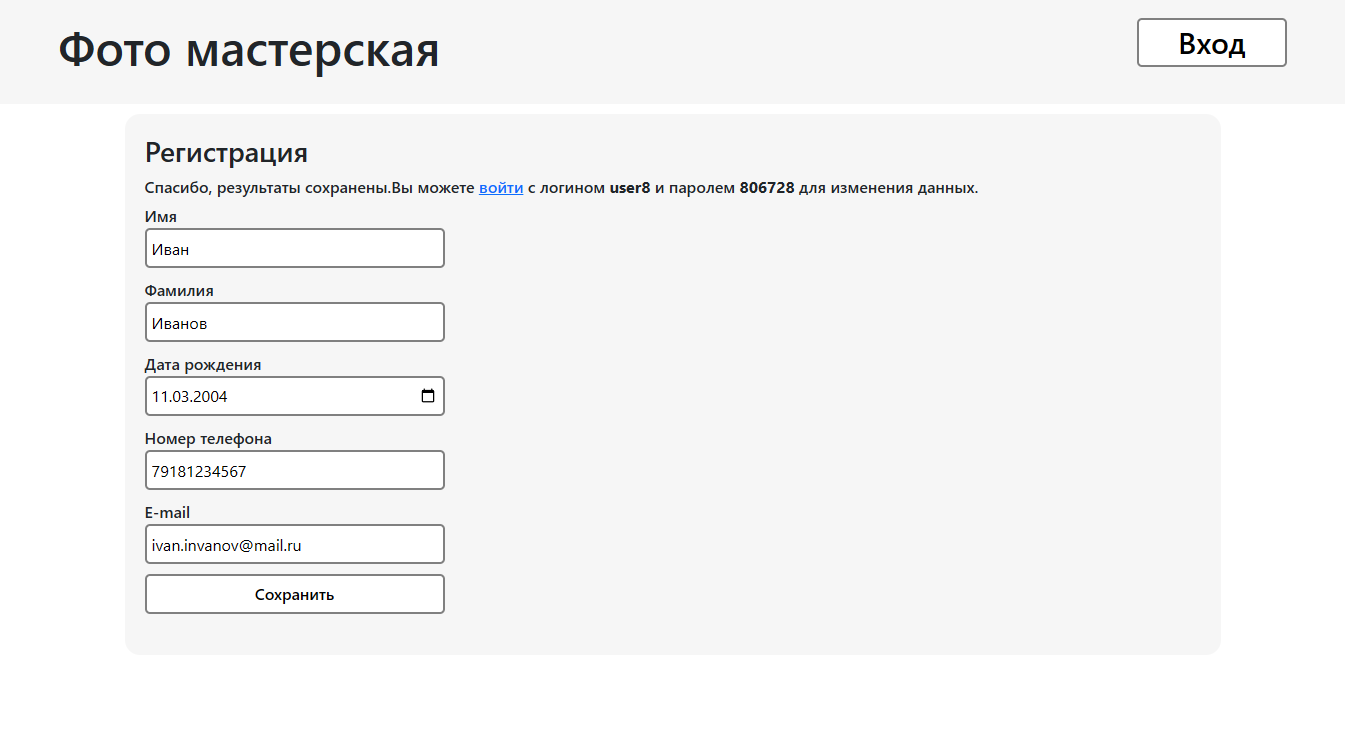


Рисунок 9 – Cтраница index.php при сохранении данных.

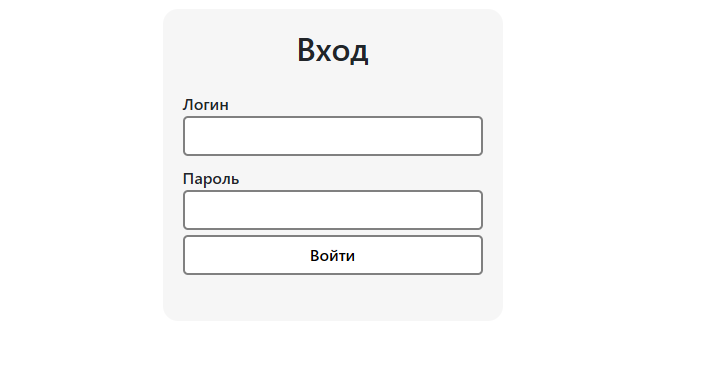
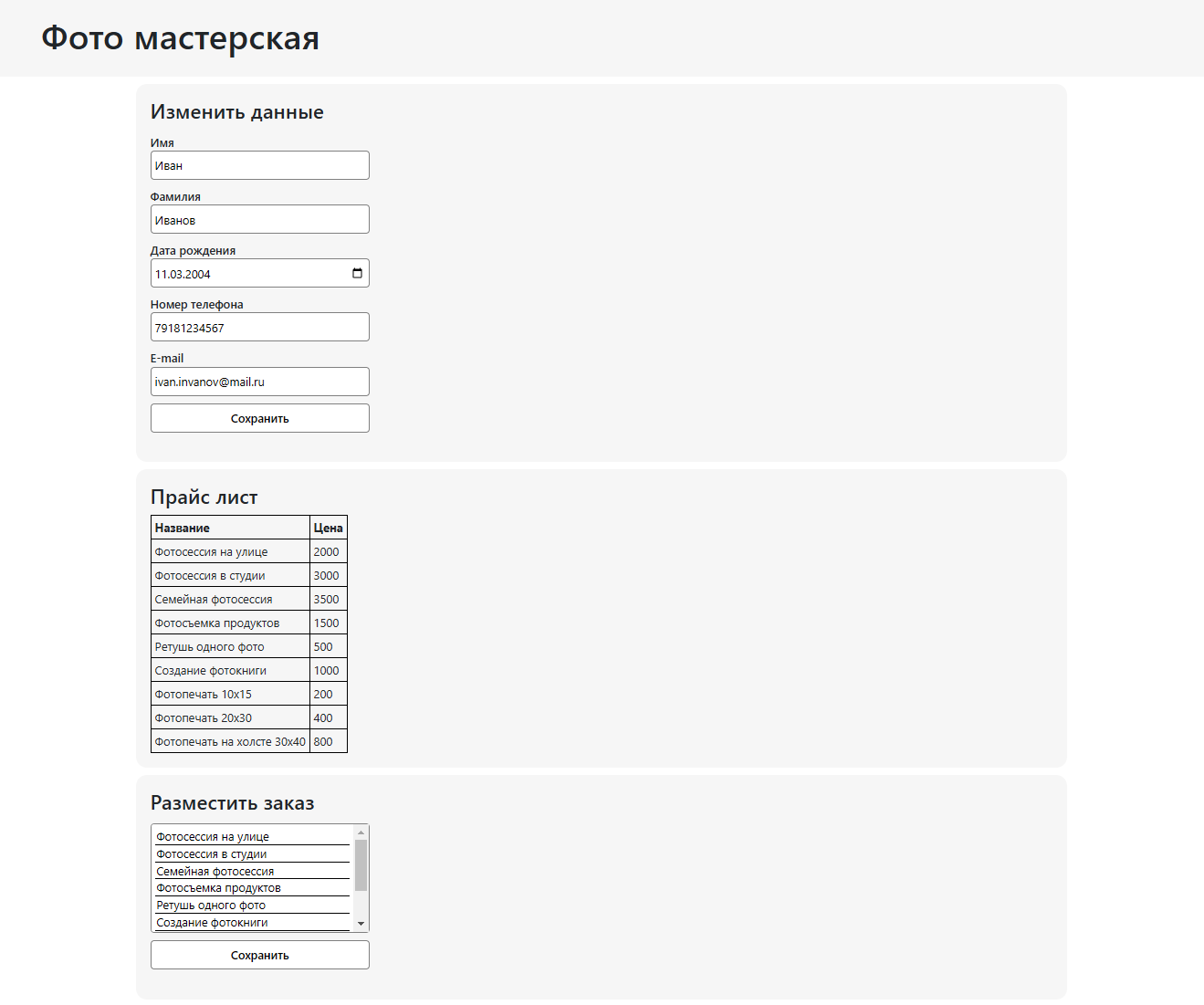


Рисунок 10 – Страница login.php.

Добавим в index.php проверку, есть ли в сессии логин, то есть заказчик вошел со своим логином и паролем. Если нет, то оставим страницу без изменений. Если есть – загрузим данные заказчика из базы данных и запишем их в форму. При изменении этих данных в форме и сохранении ее, данные заказчика проходят валидацию. Если ошибок нет – обновим данные заказчика в таблице Users. Добавим таблицу «Прайс лист», в которой будет информация о наименовании услуги и ее цене, значения будут загружаться из базы данных из таблицы Prices. Добавим форму «Разместить заказ» с полем множественного выбора. При сохранении формы в таблицу Orders запишутся уникальный идентификатор заказа, номер заказчика, текущая дата и случайный номер исполнителя из таблицы Performers. Также в таблицу Order\_price запишутся номер заказа и номера выбранных услуг. Добавим таблицу «Ваши заказы», в которой будут отображаться заказы текущего заказчика, значения будут загружаться из базы данных из таблицы Orders. Добавим форму «Удалить заказ» с полем номера заказа. При сохранении формы проверяется, есть ли такой заказ и принадлежит ли он текущему заказчику, если ошибок нет, то данный заказ удаляется из таблицы Orders. Добавим кнопку Выход, которая удаляет данные из сессии, куки и обновляет страницу.



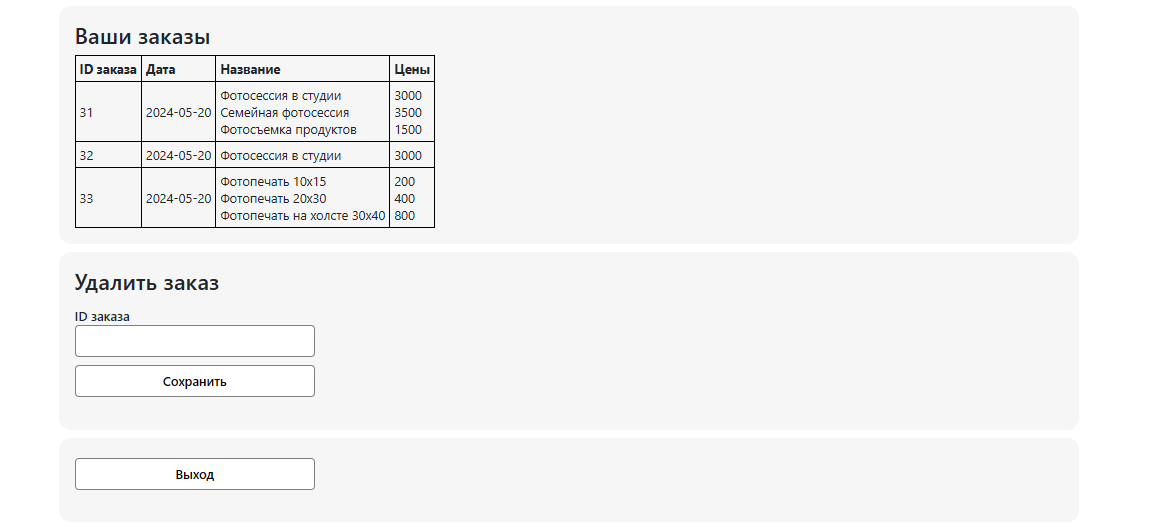


Рисунок 11 – Страница index.php при введенных заказчиком логине и пароле.

Создадим страницу admin.php, на которой будем редактировать остальные таблицы.

Добавим таблицу «Заказчики», данные которой загружаются из таблицы Users. Добавим форму «Изменение заказчиков» с полями ID заказчика, фамилия, имя, дата рождения, номер телефона, email, кнопками «Изменить» и «Удалить». При нажатии кнопки «Удалить» проверяется, есть ли такой ID в таблице Users, данные проходят валидацию. Если ошибок нет – обновим данные заказчика в таблице Users. При нажатии «Удалить» достаточно заполнить только ID, проверяется, есть ли такой ID в таблице Users. Если ошибок нет – заказчик с этим ID удаляется из таблицы Users.

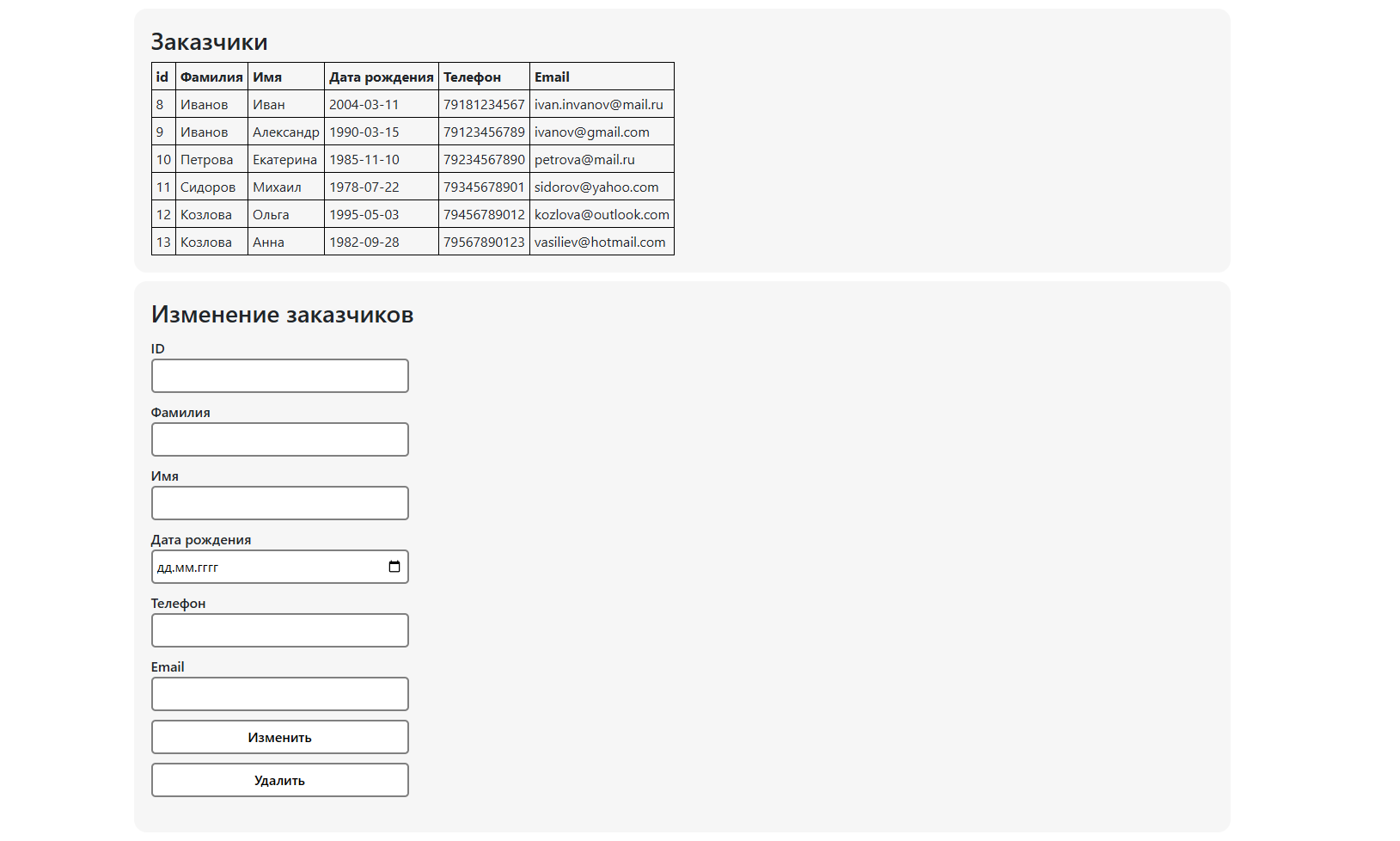


Рисунок 12 – Страница admin.php, таблица «Заказчики» и форма «Изменение заказчиков».

Создадим раздел «Заказы», где будем искать заказы по определенному признаку. Создадим форму «Найти заказ по id заказчика» с полем id и кнопкой «Найти». При нажатии кнопки проверяется, есть ли такой id заказчика в таблице Orders. Если id не найден – выводится «Заказов не найдено», иначе – выводятся все заказы данного заказчика. Создадим форму «Найти заказ по дате» с полями интервала даты и кнопкой «Найти». При нажатии кнопки проверяется, что первая дата меньше второй. Если заказы не найдены – выводится «Заказов не найдено» иначе– выводятся все заказы, даты создания которых находятся в этом интервале.

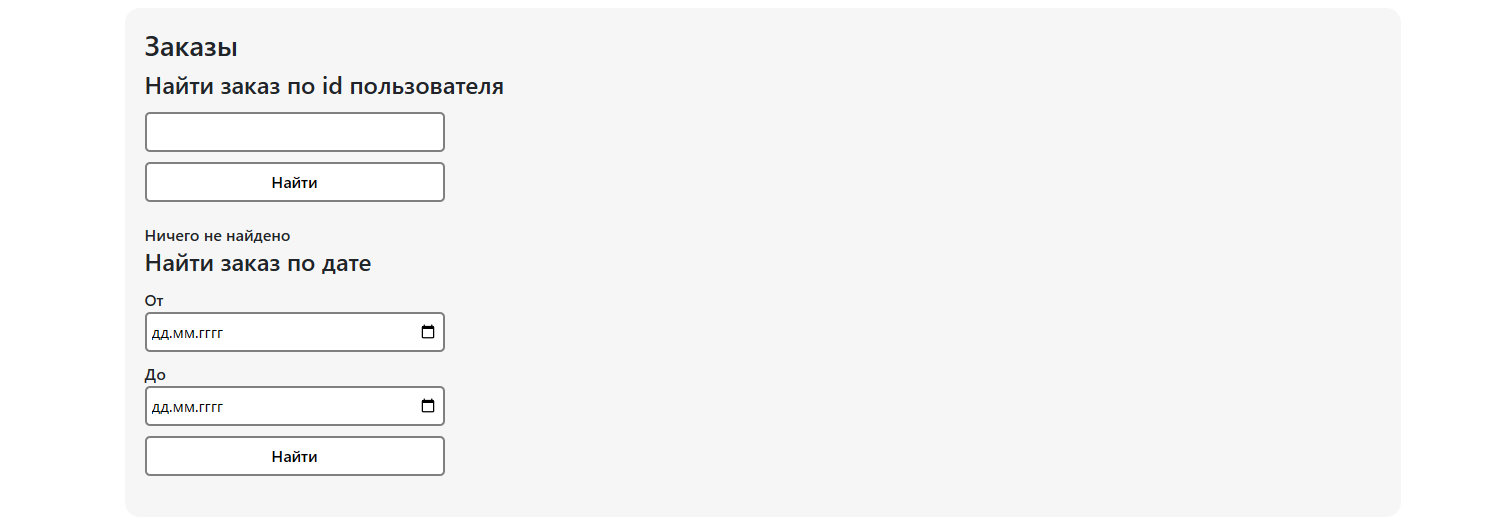


Рисунок 13 – Страница admin.php, раздел «Заказы».

Добавим таблицу «Прайс лист», в которой будет информация о наименовании услуги, ее цене и id услуги, значения будут загружаться из таблицы Prices. Добавим форму «Изменение прайс листа» с полями ID услуги, название, цена, кнопками «Добавить», «Изменить» и «Удалить». При нажатии кнопки «Добавить» данные из формы проходят валидацию. Если ошибок нет –данные добавляются в таблицу Prices. При нажатии кнопки «Изменить» данные из формы проходят валидацию и проверяется, есть ли такой ID в таблице. Если ошибок нет –данные для этого ID обновляются в таблице Prices. При нажатии «Удалить» достаточно заполнить только ID, проверяется, есть ли такой ID в таблице Prices. Если ошибок нет – запись с этим ID удаляется из таблицы Prices.

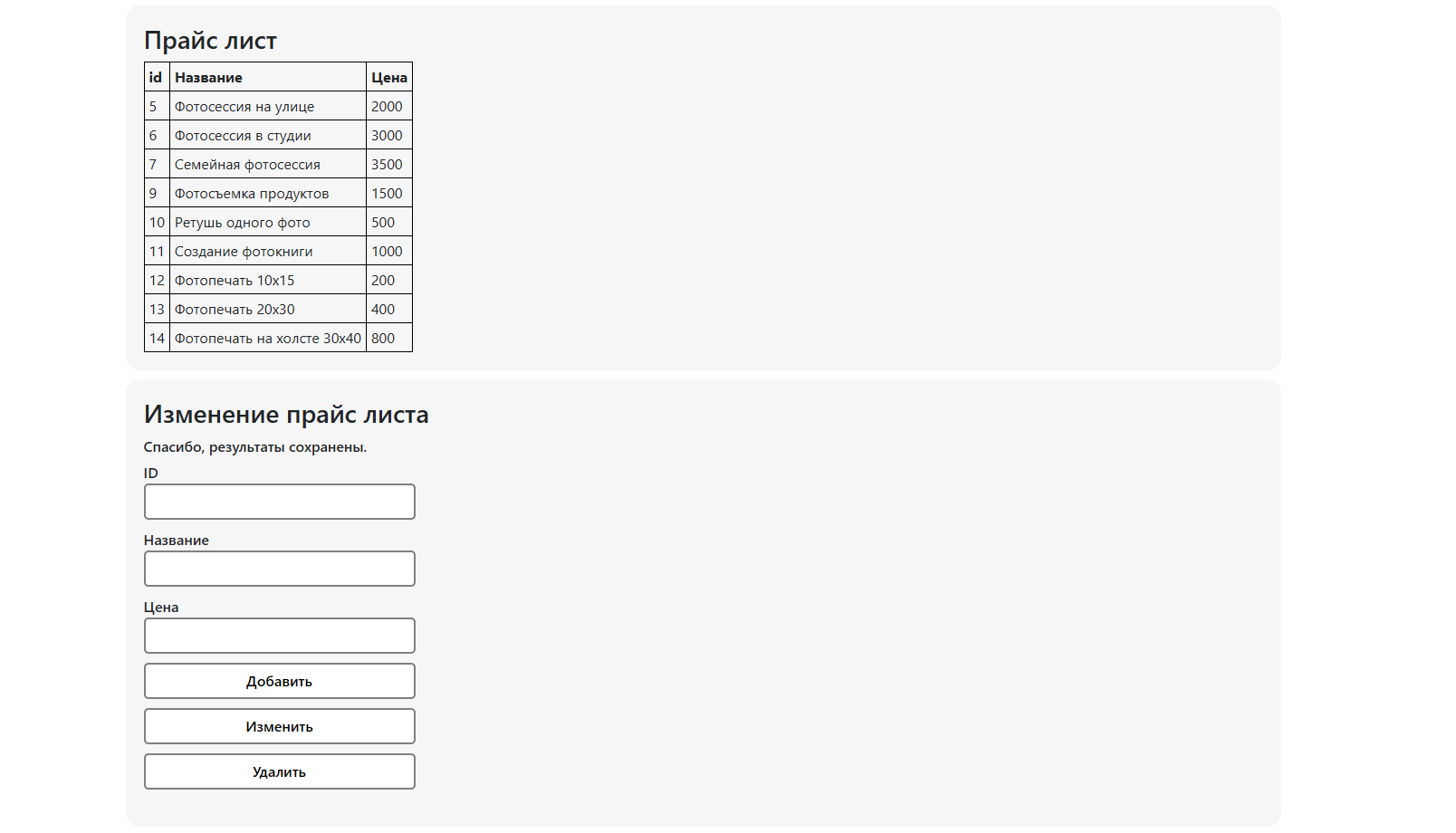


Рисунок 14 – Страница admin.php, таблица «Прайс лист» и форма «Изменение прайс листа».

Добавим таблицу «Исполнители», в которой будет информация об исполнителях, а именно ID, фамилия, имя, дата рождения, номер телефона, email, значения будут загружаться из таблицы Performers. Добавим форму «Изменение исполнителей» с полями ID, фамилия, имя, дата рождения, номер телефона, email, кнопками «Добавить», «Изменить» и «Удалить». При нажатии кнопки «Добавить» данные из формы проходят валидацию. Если ошибок нет –данные добавляются в таблицу Performers. При нажатии кнопки «Изменить» данные из формы проходят валидацию и проверяется, есть ли такой ID в таблице. Если ошибок нет –данные для этого ID обновляются в таблице Prices. При нажатии «Удалить» достаточно заполнить только ID, проверяется, есть ли такой ID в таблице Performers. Если ошибок нет – запись с этим ID удаляется из таблицы Performers.

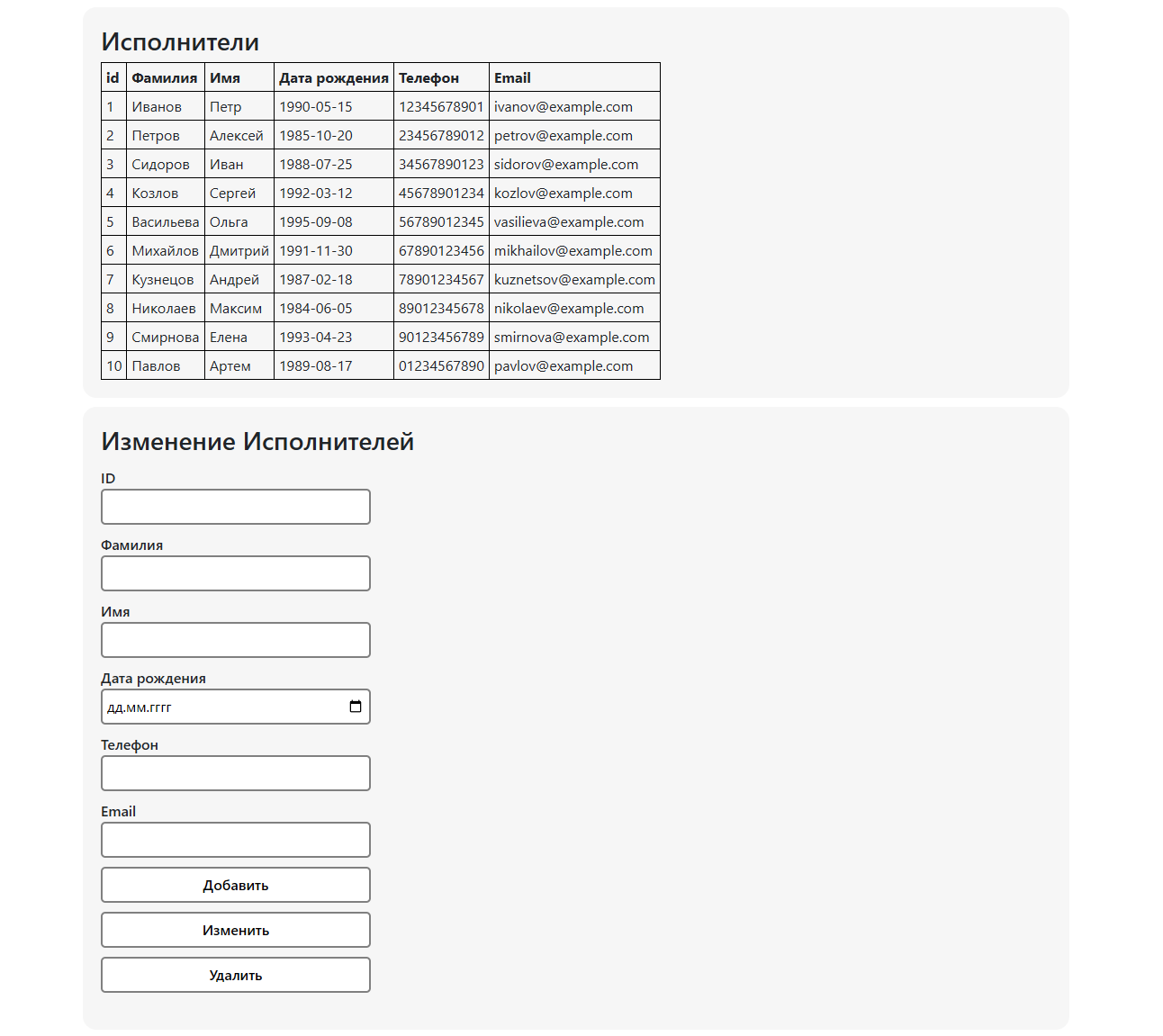


Рисунок 15 – Страница admin.php, таблица «Исполнители» и форма «Изменение исполнителей».

ПРИЛОЖЕНИЕ

Код программы

Index.php

<?php  
ob\_start();  
?>  
<html>  
<head>  
 <title>Form</title>  
 <meta charset="utf-8">  
 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-QWTKZyjpPEjISv5WaRU9OFeRpok6YctnYmDr5pNlyT2bRjXh0JMhjY6hW+ALEwIH" crossorigin="anonymous">  
 <link rel="stylesheet" href='style.css' type="text/css">  
</head>  
<body>  
<div class="header">  
 <div class="wrap">  
 <div class="title">  
 <h1>Фото мастерская</h1>  
 </div>  
 <div class="log-button">  
 <?php if(session\_start() and empty($\_SESSION['practice\_login'])) {?>  
 <form action="login.php">  
 <input type="submit" value="Вход">  
 </form>  
 <?php } ?>  
 </div>  
 </div>  
</div>