

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ

«МИСИС»

Институт компьютерных наук
Кафедра Инфокоммуникационных технологий

КУРСОВАЯ РАБОТА

по дисциплине

Разработка клиент-серверных приложений

на тему:

«Разработать скрипты фотоальбома»

Выполнил студент группы
БИСТ-21-3:
Картвелишвили К.И.

Подпись _____

Проверил:
Парфенова Е. В.

Оценка _____

Подпись _____

Москва 2024

СОДЕРЖАНИЕ

Задание по РКСП:	3
Листинг файла auth.php	4
Листинг файла album.php	6
Листинг файла view_all.php	9
Листинг файла config.php	10
Листинг файла logout.php	10
Листинг файла style.css	10
Блок-схема всего проекта	13
Блок-схема album.php	14
Блок-схема view_all.php	14
Инструкции по запуску проекта	15
Пример выполнения программы	16
Заключение	19
Список литературы	20

Задание по РКСП:

Разработать скрипты фотоальбома. Скрипты должны позволять пользователю (после авторизации) добавлять, удалять и изменять подписи к фотографиям, загруженным этим пользователем, и просматривать фотографии других пользователей

Требования к разрабатываемому ПО:

1. Все скрипты должны быть написаны на языке программирования PHP. Допускается обоснованное использование фрагментов кода на языке SQL и HTML.
2. Все скрипты должны работать на локальном сервере без выхода в сеть Интернет (кроме заранее согласованных с преподавателем вариантов).
3. Допускается использование любых методов, функций, классов и т.д., которые не противоречат настоящему заданию.
4. Не допускается использование разработанных другими авторами блоков кода.
5. При разработке допускается использование любого вспомогательного ПО при условии лицензионной чистоты этого ПО.
6. Все данные, если это не указано отдельно, должны храниться в БД (СУБД MySQL).
7. Для разработки интерфейса допускается использование любых языков и технологий локального программирования (Си#, Java, Java Script, VB Script, AJAX и т.д.). При этом основное задание должно быть выполнено средствами серверного программирования.
8. Должно быть реализовано клиентское ПО (например, на Си#).

Требования к оформлению:

Отчет содержать следующие разделы:

- титульный лист;
- задание;
- блок-схемы всех скриптов;
- пример выполнения программы;
- листинг;
- выводы;
- список используемой литературы.

Листинг файла auth.php

```
<?php
session_start(); // Начало сессии PHP
require 'config.php'; // Подключение файла конфигурации для доступа к базе данных

$error = ""; // Переменная для хранения сообщений об ошибках
$success = ""; // Переменная для хранения сообщений об успешных операциях

// Обработка формы при отправке
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $action = $_POST['action']; // Получение действия (login или register) из формы
    $username = $mysqli->real_escape_string($_POST['username']); // Экранирование имени пользователя
    // для безопасного использования в SQL-запросе
    $password = $_POST['password']; // Получение пароля из формы

    if ($action == 'register') {
        // Хэширование пароля
        $hashed_password = password_hash($password, PASSWORD_BCRYPT); // Хэширование пароля с
        // использованием алгоритма BCRYPT

        // Проверка на существование пользователя
        $result = $mysqli->query("SELECT id FROM users WHERE username='$username'"); // SQL-запрос
        // для проверки существования пользователя

        if ($result->num_rows > 0) {
            $error = "Username already exists!"; // Установка сообщения об ошибке, если пользователь с
            // таким именем уже существует
        } else {
            // Вставка нового пользователя
            $mysqli->query("INSERT INTO users (username, password) VALUES ('$username',
            '$hashed_password')"); // SQL-запрос для вставки нового пользователя
            $success = "Registration successful!"; // Установка сообщения об успешной регистрации
        }
    } elseif ($action == 'login') {
        // Проверка учетных данных
        $result = $mysqli->query("SELECT * FROM users WHERE username='$username'"); // SQL-запрос
        // для получения данных пользователя
        $user = $result->fetch_assoc(); // Получение данных пользователя из результата запроса

        if ($user && password_verify($password, $user['password'])) { // Проверка пароля пользователя
            $_SESSION['user_id'] = $user['id']; // Сохранение ID пользователя в сессии
            header("Location: album.php"); // Перенаправление на страницу альбома
            exit(); // Прекращение выполнения скрипта
        } else {
            $error = "Invalid username or password"; // Установка сообщения об ошибке, если имя
            // пользователя или пароль неверны
        }
    }
}
?>
```

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="style.css"> <!-- Подключение CSS-файла -->
  <title>Auth</title> <!-- Заголовок страницы -->
</head>

<body>
  <div class="container">
    <h2>Authentication</h2>
    <?php if ($error) : ?> <!-- Проверка наличия сообщения об ошибке -->
      <div class="error"><?= $error ?></div> <!-- Отображение сообщения об ошибке -->
    <?php endif; ?>
    <?php if ($success) : ?> <!-- Проверка наличия сообщения об успешной операции -->
      <div class="success"><?= $success ?></div> <!-- Отображение сообщения об успешной операции -->
    <?php endif; ?>
    <!-- Форма авторизации и регистрации -->
    <form method="post">
      <input type="text" name="username" placeholder="Username" required> <!-- Поле для ввода имени
пользователя -->
      <input type="password" name="password" placeholder="Password" required> <!-- Поле для ввода
пароля -->
      <button type="submit" name="action" value="login">Login</button> <!-- Кнопка для входа -->
      <button type="submit" name="action" value="register">Register</button> <!-- Кнопка для
регистрации -->
    </form>
  </div>
</body>

</html>
```

Листинг файла album.php

```
<?php
session_start(); // Начало сессии
if (!isset($_SESSION['user_id'])) { // Проверка, авторизован ли пользователь
    header("Location: auth.php"); // Перенаправление на страницу авторизации, если пользователь не
    авторизован
    exit(); // Прекращение выполнения скрипта
}

require 'config.php'; // Подключение файла конфигурации для подключения к базе данных

$user_id = $_SESSION['user_id']; // Получение ID пользователя из сессии

// Обработка загрузки фотографии
if ($_SERVER['REQUEST_METHOD'] === "POST" && isset($_FILES["photo"])) { // Проверка, что
    запрос POST и файл был отправлен
        $caption = $mysqli->real_escape_string($_POST['caption']); // Экранирование строки для безопасного
        использования в SQL-запросе
        $target_dir = "uploads/"; // Директория для загрузки фотографий
        $target_file = $target_dir . basename($_FILES["photo"]["name"]); // Полный путь к файлу
        $uploadOk = 1; // Флаг успешности загрузки
        $imageFileType = strtolower(pathinfo($target_file, PATHINFO_EXTENSION)); // Получение
        расширения файла

        // Проверка на изображение
        $check = getimagesize($_FILES["photo"]["tmp_name"]); // Проверка, является ли файл изображением
        if ($check !== false) {
            $uploadOk = 1; // Файл является изображением
        } else {
            echo "File is not an image."; // Сообщение об ошибке
            $uploadOk = 0; // Файл не является изображением
        }

        // Проверка на существование файла
        if (file_exists($target_file)) {
            echo "Sorry, file already exists."; // Сообщение об ошибке
            $uploadOk = 0; // Файл уже существует
        }

        // Проверка размера файла
        if ($_FILES["photo"]["size"] > 5000000) { // Максимальный размер файла 5MB
            echo "Sorry, your file is too large."; // Сообщение об ошибке
            $uploadOk = 0; // Файл слишком большой
        }

        // Разрешенные форматы файлов
        if ($imageFileType !== "jpg" && $imageFileType !== "png" && $imageFileType !== "jpeg" &&
        $imageFileType !== "gif") {
            echo "Sorry, only JPG, JPEG, PNG & GIF files are allowed."; // Сообщение об ошибке
            $uploadOk = 0; // Неразрешенный формат файла
        }
    }
}
```

```

}

// Проверка на ошибки загрузки
if ($uploadOk == 0) {
    echo "Sorry, your file was not uploaded."; // Сообщение об ошибке
} else {
    // Перемещение файла и сохранение информации в БД
    if (move_uploaded_file($_FILES["photo"]["tmp_name"], $target_file)) {
        $stmt = $mysqli->prepare("INSERT INTO photos (user_id, path, caption) VALUES (?, ?, ?)"); //
        Подготовка SQL-запроса
        $stmt->bind_param("iss", $user_id, $target_file, $caption); // Связывание параметров запроса
        $stmt->execute(); // Выполнение запроса
        $stmt->close(); // Закрытие запроса
        header("Location: album.php"); // Перенаправление на страницу альбома
        exit(); // Прекращение выполнения скрипта
    } else {
        echo "Sorry, there was an error uploading your file."; // Сообщение об ошибке
    }
}
}

// Обработка удаления фотографии
if ($_SERVER["REQUEST_METHOD"] == "POST" && isset($_POST['delete_photo'])) {
    $photo_id = $_POST['photo_id']; // Получение ID фотографии из запроса
    $photo_path = $_POST['photo_path']; // Получение пути к файлу из запроса

    if (file_exists($photo_path)) { // Проверка, существует ли файл
        unlink($photo_path); // Удаление файла
    }

    $stmt = $mysqli->prepare("DELETE FROM photos WHERE id = ? AND user_id = ?"); // Подготовка
    SQL-запроса для удаления записи из БД
    $stmt->bind_param("ii", $photo_id, $user_id); // Связывание параметров запроса
    $stmt->execute(); // Выполнение запроса
    $stmt->close(); // Закрытие запроса
    header("Location: album.php"); // Перенаправление на страницу альбома
    exit(); // Прекращение выполнения скрипта
}

// Обработка редактирования подписи
if ($_SERVER["REQUEST_METHOD"] == "POST" && isset($_POST['edit_caption'])) {
    $photo_id = $_POST['photo_id']; // Получение ID фотографии из запроса
    $new_caption = $mysqli->real_escape_string($_POST['new_caption']); // Экранирование новой подписи
    для безопасного использования в SQL-запросе

    $stmt = $mysqli->prepare("UPDATE photos SET caption = ? WHERE id = ? AND user_id = ?"); //
    Подготовка SQL-запроса для обновления подписи
    $stmt->bind_param("sii", $new_caption, $photo_id, $user_id); // Связывание параметров запроса
    $stmt->execute(); // Выполнение запроса
    $stmt->close(); // Закрытие запроса
    header("Location: album.php"); // Перенаправление на страницу альбома

```

```

    exit(); // Прекращение выполнения скрипта
}

// Получение фотографий пользователя
$photos = $mysqli->query("SELECT * FROM photos WHERE user_id = $user_id"); // Выполнение SQL-запроса для получения всех фотографий пользователя
?>

<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="style.css"> <!-- Подключение CSS-файла -->
    <title>My Album</title> <!-- Заголовок страницы -->
</head>

<body>
    <div class="container">
        <h2>My Album</h2>
        <a href="view_all.php">View All Photos</a> | <a href="logout.php">Logout</a> <!-- Ссылки для просмотра всех фотографий и выхода из системы -->

        <!-- Форма для загрузки фотографии -->
        <form action="album.php" method="post" enctype="multipart/form-data">
            <input type="file" name="photo" required> <!-- Поле для выбора файла -->
            <input type="text" name="caption" placeholder="Caption" required> <!-- Поле для ввода подписи -->
        >

        <button type="submit">Upload Photo</button> <!-- Кнопка для загрузки фотографии -->
    </form>

    <div class="album">
        <?php while ($photo = $photos->fetch_assoc()) : ?> <!-- Цикл для отображения всех фотографий пользователя -->
        <div class="photo">
             <!-- Отображение фотографии -->
            <p><? = htmlspecialchars($photo['caption']) ?></p> <!-- Отображение подписи к фотографии -->
        >

        <!-- Форма для удаления фотографии -->
        <form action="album.php" method="post">
            <input type="hidden" name="photo_id" value="<? = $photo['id'] ?>"> <!-- Скрытое поле с ID фотографии -->
            <input type="hidden" name="photo_path" value="<? = htmlspecialchars($photo['path']) ?>"> <!-- Скрытое поле с путем к файлу -->
            <button type="submit" name="delete_photo">Delete</button> <!-- Кнопка для удаления фотографии -->
        </form>

        <!-- Форма для редактирования подписи -->
        <form action="album.php" method="post">

```



```

        <input type="hidden" name="photo_id" value="<?= $photo['id'] ?>"> <!-- Скрытое поле с ID
фотографии -->
        <input type="text" name="new_caption" placeholder="New Caption" required> <!-- Поле для
ввода новой подписи -->
        <button type="submit" name="edit_caption">Edit Caption</button> <!-- Кнопка для
редактирования подписи -->
    </form>
</div>
<?php endwhile; ?>
</div>
</div>
</body>
</html>

```

Листинг файла view_all.php

```

<?php
session_start(); // Начало сессии PHP

// Проверка авторизации пользователя
if (!isset($_SESSION['user_id'])) {
    header("Location: auth.php"); // Перенаправление на страницу авторизации, если пользователь не
авторизован
    exit(); // Прекращение выполнения скрипта
}

require 'config.php'; // Подключение файла конфигурации для доступа к базе данных

// Получение всех фотографий из базы данных
$photos = $mysqli->query("SELECT photos.*, users.username FROM photos JOIN users ON photos.user_id
= users.id"); // Запрос к базе данных на получение всех фотографий с именами пользователей
?>

<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8"> <!-- Установка кодировки символов -->
    <meta name="viewport" content="width=device-width, initial-scale=1.0"> <!-- Обеспечение
адаптивности страницы на разных устройствах -->
    <link rel="stylesheet" href="style.css"> <!-- Подключение CSS-файла -->
    <title>All Photos</title> <!-- Заголовок страницы -->
</head>

<body>
    <div class="container">
        <h2>All Photos</h2> <!-- Заголовок страницы -->

```

```

    <a href="album.php">My Album</a> | <a href="logout.php">Logout</a> <!-- Ссылки на альбом
пользователя и выход из аккаунта -->
    <div class="album">
        <?php while ($photo = $photos->fetch_assoc()) : ?> <!-- Цикл для вывода всех фотографий из базы
данных -->
            <div class="photo">
                 <!-- Отображение
изображения -->
                <p><?= htmlspecialchars($photo['caption']) ?></p> <!-- Отображение подписи к фотографии --
>
                <p>Uploaded by <?= htmlspecialchars($photo['username']) ?></p> <!-- Отображение имени
пользователя, загрузившего фотографию -->
            </div>
        <?php endwhile; ?>
    </div>
</div>
</body>
</html>

```

Листинг файла config.php

```

<?php
$mysqli = new mysqli("localhost", "root", "", "photo_album");
if ($mysqli->connect_error) {
    die("Connection failed: " . $mysqli->connect_error);
}

```

Листинг файла logout.php

```

<?php
session_start();
session_destroy();
header("Location: auth.php");
exit();

```

Листинг файла style.css

```

body {
    font-family: Arial, sans-serif;
    /* Устанавливает шрифт Arial или, если он недоступен, любой шрифт без засечек */
}

.container {
    width: 80%;
    margin: 0 auto;
    padding: 20px;
    /* Задаёт ширину контейнера 80% от ширины родительского элемента,
выравнивает контейнер по центру и добавляет внутренние отступы по 20px */
}

```

```
}

h2 {
  text-align: center;
  /* Выравнивает заголовки уровня h2 по центру */
}

form {
  margin-bottom: 20px;
  /* Добавляет нижний отступ 20px для формы */
}

input[type="text"], input[type="password"], input[type="file"] {
  display: block;
  width: 100%;
  padding: 10px;
  margin-bottom: 10px;
  border: 1px solid #ccc;
  border-radius: 5px;
  /* Задаст стиль текстовым, файловым полям и полям для ввода пароля */
}

button {
  padding: 10px 20px;
  background-color: blueviolet;
  border: none;
  color: white;
  border-radius: 5px;
  cursor: pointer;
  /* Задаст стиль кнопкам */
}

button:hover {
  background-color: blueviolet;
  /* Устанавливает цвет фона синий фиолетовый (blueviolet) при наведении на кнопку */
}

.error {
  color: red;
  text-align: center;
  /* Задаст стиль для элементов с классом .error */
}

.success {
  color: blueviolet;
  text-align: center;
  /* Задаст стиль для элементов с классом .success */
}

.album {
  display: flex;
```

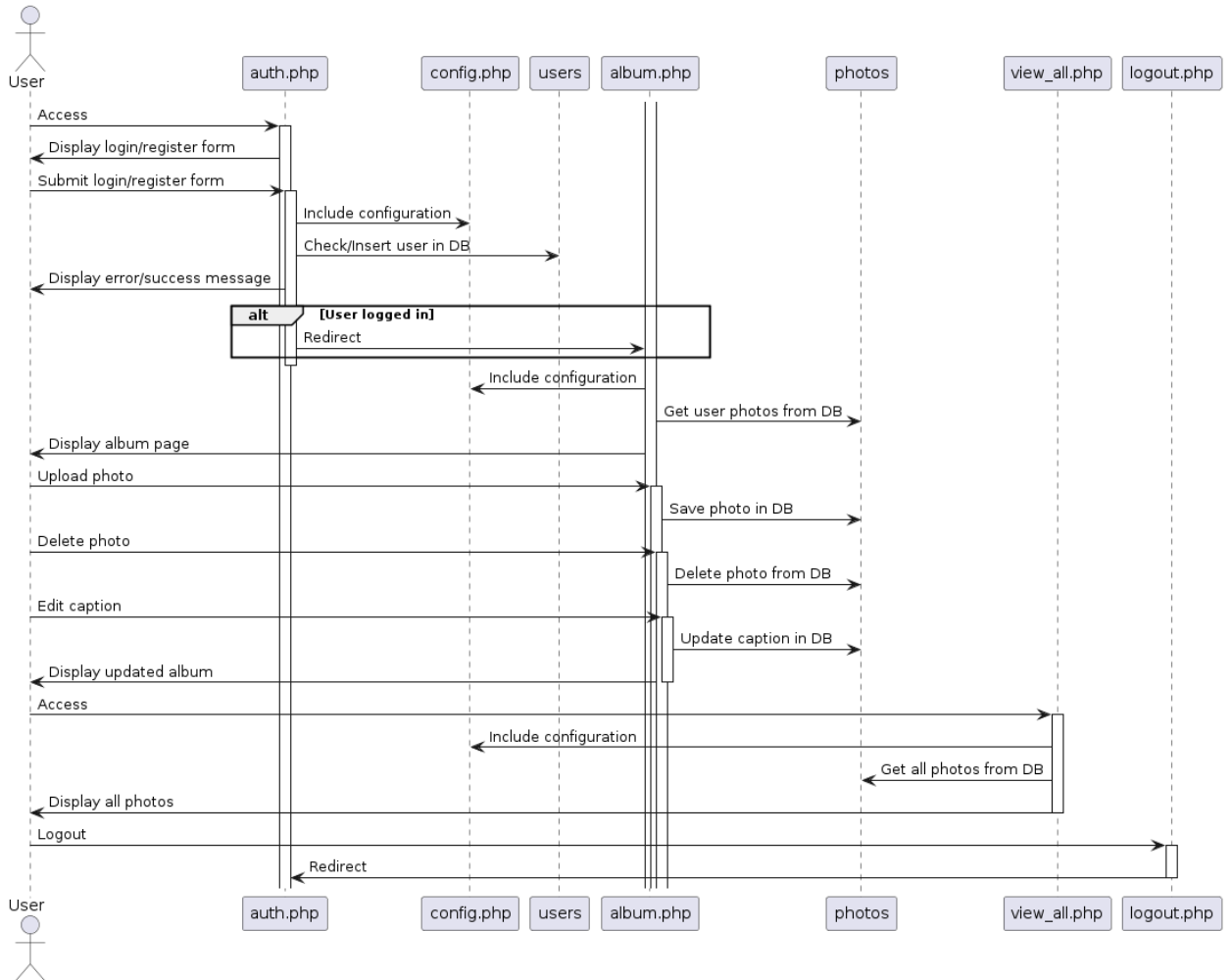
```
flex-wrap: wrap;
gap: 20px;
/* Задает стиль для контейнера альбома */
}

.photo {
  flex: 1 0 30%;
  border: 1px solid #ccc;
  border-radius: 5px;
  padding: 10px;
  text-align: center;
  /* Задает стиль для каждого фото в альбоме */
}

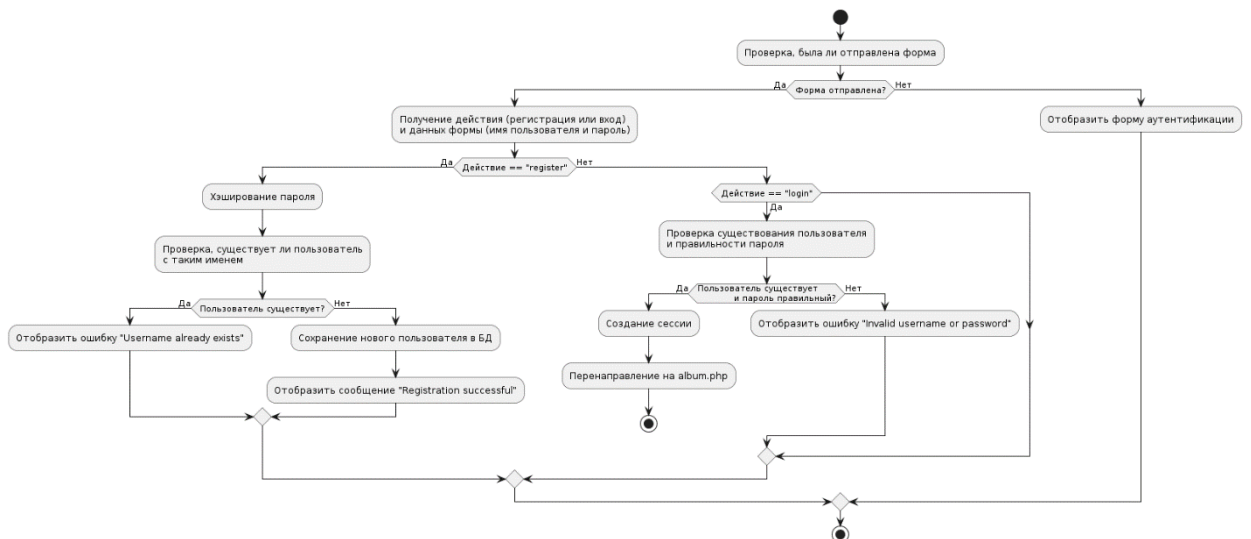
.photo img {
  max-width: 100%;
  height: auto;
  border-radius: 5px;
  /* Задает стиль изображениям */
}

.photo form {
  display: inline;
  /* Задает блочно-строчный режим отображения для форм внутри элементов с классом .photo */
}
```

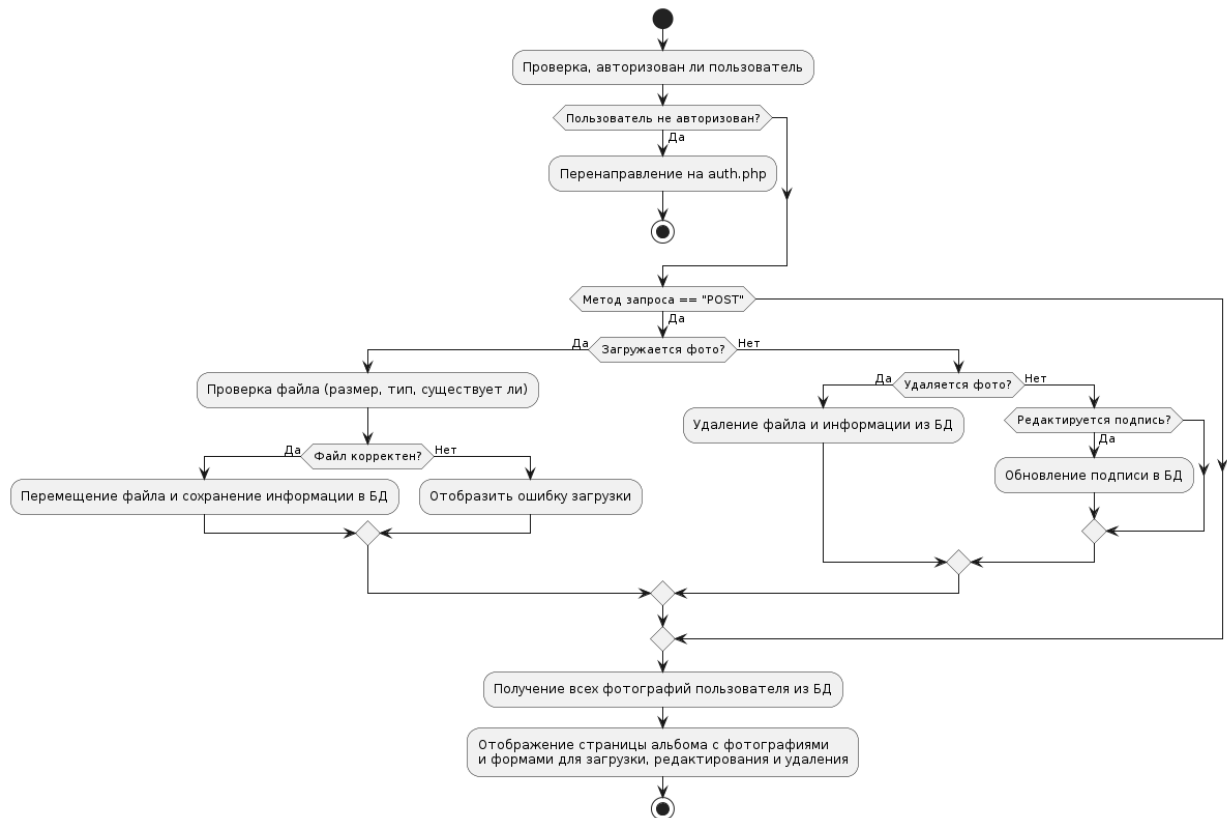
Блок-схема всего проекта



Блок-схема auth.php



Блок-схема album.php



Блок-схема view_all.php



Инструкции по запуску проекта

Установка и настройка среды разработки:

Установите XAMPP (или аналогичный локальный сервер).

Запустите Apache и MySQL.

Создание базы данных:

Откройте phpMyAdmin.

Выполните SQL скрипт из файла db.sql.

Настройка файлов проекта:

Создайте папку photo_album в директории вашего веб-сервера (htdocs для XAMPP).

Скопируйте все файлы и папки из предоставленной структуры файлов в эту директорию.

Запуск проекта:

Откройте браузер и перейдите по адресу http://localhost/photo_album/auth.php.

Зарегистрируйтесь и войдите в систему.

Добавляйте, удаляйте и редактируйте фотографии на странице album.php.

Просматривайте фотографии всех пользователей на странице view_all.php.

Проверка функционала

Регистрация и вход в систему:

Зарегистрируйтесь с новым именем пользователя и паролем.

Войдите с использованием зарегистрированных данных.

Добавление фотографий:

На странице album.php загрузите фотографии и добавьте подписи.

Просмотр и управление фотографиями:

Убедитесь, что загруженные фотографии отображаются в альбоме пользователя.

Удалите и отредактируйте подписи к фотографиям.

Просмотр фотографий других пользователей:

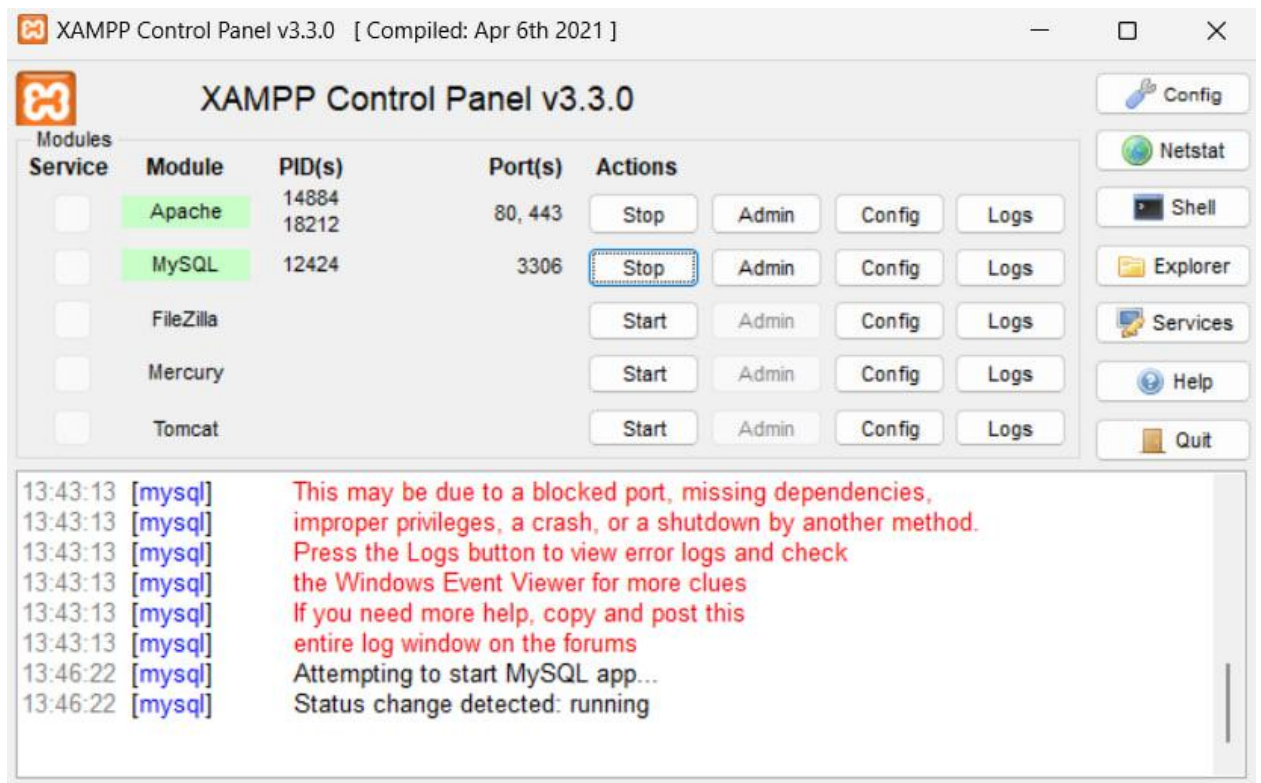
Перейдите на страницу view_all.php и убедитесь, что фотографии всех пользователей отображаются корректно.

Выход из системы:

Выйдите из системы с помощью ссылки Logout.

Пример выполнения программы

Подготовка к запуску проекта, запуск MySQL и Apache:



Регистрация пользователя:

Authentication

Выбор фотографии для добавления:

My Album

[View All Photos](#) | [Logout](#)

4c532ef88a802914b984a4caa6e1ff08.jpeg

Добавление фотографий:


[View All Photos](#) | [Logout](#)

Выберите файл

maxresdefault.jpg

Porsche 3

Upload Photo




Porsche 1

Delete

New Caption

Edit Caption



Porsche 2

Delete

New Caption

My Album


[View All Photos](#) | [Logout](#)

Выберите файл

Файл не выбран

Caption

Upload Photo




Porsche 1

Delete

Porsche 4

Edit Caption




Porsche 2

Delete

New Caption

Edit Caption



Porsche 3

Delete


New Caption

Edit Caption

Просмотр всех фотографий пользователей:


All Photos

[My Album](#) | [Logout](#)




Porsche 4

Uploaded by kornelikkk



Porsche 2

Uploaded by kornelikkk



Porsche 3

Uploaded by kornelikkk



Porsche 3
Uploaded by korneliikk



Porsche 5
Uploaded by korneliikk



Porsche 6
Uploaded by korneliikk



dron
Uploaded by korneliikk2



dron 2
Uploaded by korneliikk2



dron 3
Uploaded by korneliikk2

Заключение

Проект разработки фотоальбома на PHP и MySQL предоставил возможность изучить и применить различные аспекты веб-разработки, включая серверную и клиентскую части. В ходе реализации проекта удалось успешно интегрировать фронтенд и бэкенд, обеспечив функционал для загрузки, хранения и отображения фотографий пользователей, а также разработать интерфейс для регистрации и аутентификации. Важной частью проекта стало управление базой данных MySQL, где были созданы необходимые таблицы для хранения информации о пользователях и фотографиях, реализованы операции вставки, обновления и удаления записей, а также обеспечено связывание таблиц с использованием внешних ключей.

В итоге разработка фотоальбома на PHP и MySQL позволила приобрести ценные навыки и знания в области веб-разработки, охватывая ключевые аспекты создания веб-приложений, которые могут быть использованы для дальнейшего развития проекта и улучшения пользовательского интерфейса.

Список литературы

1. " PHP и MySQL для начинающих" - авторы: Робин Никсон.
2. " Создаем динамические веб-сайты с помощью PHP, MySQL и JavaScript" - авторы: Робин Никсон.
3. " Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5 " - автор: Робин Никсон.
4. "PHP documentation" - автор: PHP
5. "MySQL documentation" - автор: MySQL