

Разработка современной галереи изображений на PHP с использованием HTML, CSS и JavaScript

Екимова Яна Сергеевна

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

В данной статье рассматривается процесс разработки веб-галереи на PHP с использованием HTML, CSS, JavaScript и локального сервера. Актуальность разработки галерей изображений обусловлена растущим спросом на визуальный контент в интернете. Согласно исследованиям, более 60% пользователей предпочитают сайты с качественными изображениями и удобным интерфейсом. Кроме того, современные галереи должны быть адаптивными, быстрыми и безопасными, что требует использования современных технологий, таких как PHP для серверной обработки данных, HTML и CSS для создания структуры и стилей, а также JavaScript для обеспечения интерактивности. В результате работы была создана веб-галерея на PHP с использованием HTML, CSS, JavaScript и локального сервера.

Ключевые слова: HTML, CSS, JavaScript, Open Server, локальный сервер, php, localhost.

Development of a modern image gallery in PHP using HTML, CSS and JavaScript

Ekimova Yana Sergeevna

Sholom Aleichem Priamurskiy State University

Student

Abstract

This article discusses the process of developing a web gallery in PHP using HTML, CSS, JavaScript and a local server. The urgency of developing image galleries is due to the growing demand for visual content on the Internet. According to research, more than 60% of users prefer websites with high-quality images and user-friendly interface. In addition, modern galleries must be adaptive, fast, and secure, which requires the use of modern technologies such as PHP for server-side data processing, HTML and CSS for creating structure and styles, and JavaScript for interactivity. As a result, a web gallery was created in PHP using HTML, CSS, JavaScript and a local server.

Keywords: HTML, CSS, JavaScript, Open Server, Local server, php, localhost.

1 Введение

1.1 Актуальность

В современном мире веб-технологий галереи изображений являются неотъемлемой частью многих сайтов, начиная от портфолио фотографов и заканчивая интернет-магазинами. Они позволяют пользователям удобно просматривать, загружать и управлять визуальным контентом. Однако создание функциональной и интуитивно понятной галереи требует не только навыков программирования, но и учета современных требований к пользовательскому опыту. В данной статье рассматривается процесс разработки современной галереи изображений на PHP с использованием HTML, CSS и JavaScript. Изучим основные этапы создания системы, включая загрузку, отображение, просмотр и удаление изображений, а также реализацию интерактивных элементов, таких как модальное окно и листание фотографий.

1.2 Обзор исследований

К.В. Гарпушкина, Г.Л. Карасёва разработали web-сайт библиотеки с использованием php, html, css и javascript [1]. Разработала web - приложение для оператора сотовой связи с возможностью подбора тарифного плана А. В. Иванова [2]. А.Х. Эль-гади, В. В. Орлов разработали адаптивную часть веб-сайта «Стоматологическая клиника» с использованием HTML 5, CSS 3, PHP и JAVASCRIPT[3]. Разработали адаптивную часть веб-сайта «Станция технического обслуживания автомобилей» с использованием html 5, css 3, php и javascript С.Н. Лузан В.В Орлов[4]. А.А. Бабенко С.Н. Ушакова разработали web-приложение " Кинотеатра" с использованием JavaScript, PHP и MYSQL [5].

1.3 Цель исследования

Целью исследования является разработка современной галереи изображений на PHP с использованием HTML, CSS и JavaScript.

2 Материалы и методы

Для создания приложения используется язык программирования php. В качестве среды разработки используется текстовый редактор Notepad++, оптимизированный для языков программирования и программное обеспечение Open Server.

3 Результаты

В этой статье рассмотрим, как создать простую, но функциональную галерею на PHP. Разберем структуру проекта, работу с загрузкой и удалением изображений, а также реализацию модального окна для просмотра и управления изображениями.

Перед началом работы необходимо установить и настроить локальный сервер Open Server. Программное обеспечение Open Server доступно для

загрузки по ссылке: https://drive.google.com/file/d/1XdJSG4kVwELmT4l-0-m7PSsAe_jHYyd/view?usp=sharing.

После завершения установки требуется выполнить настройку сервера. Для этого необходимо перейти в директорию установки Open Server (по умолчанию это папка «OSPanel» на диске С или другом указанном при установке диске) и запустить исполняемый файл «Open Server Panel.exe» (рис. 1).

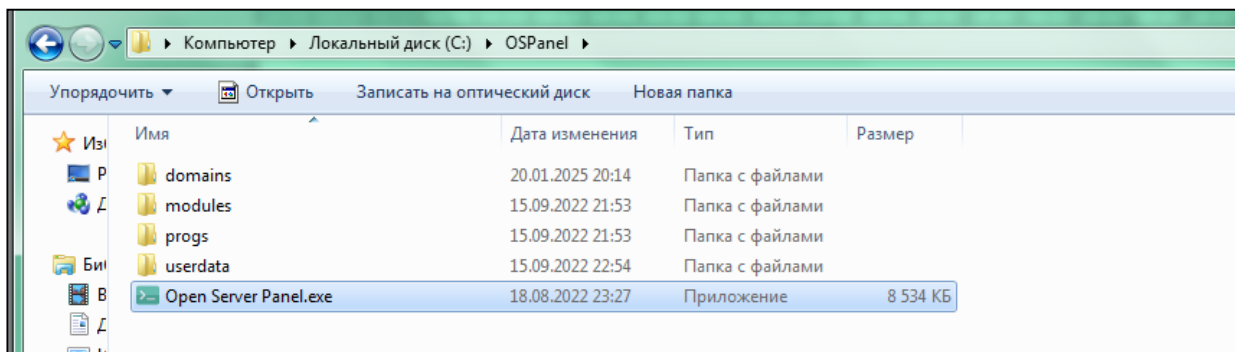


Рисунок 1. Установка «Open Server Panel.exe»

После запуска Open Server на панели управления отображается индикатор в виде красного флажка (рис. 2). Для настройки параметров сервера необходимо навести курсор на данный индикатор, кликнуть правой кнопкой мыши и выбрать пункт меню «Настройки» (рис. 3).

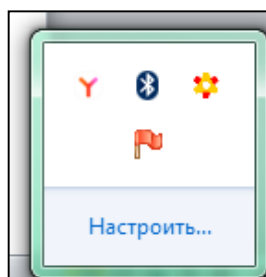


Рисунок 2. Индикатор флажка

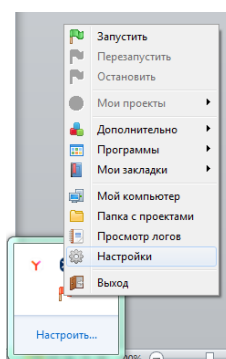


Рисунок 3. Выбор пункта Настройки

На следующем этапе необходимо перейти во вкладку «Модули» и выполнить настройку параметров в соответствии с указанными значениями

(см. пример ниже). После завершения настройки требуется сохранить изменения, нажав соответствующую кнопку.

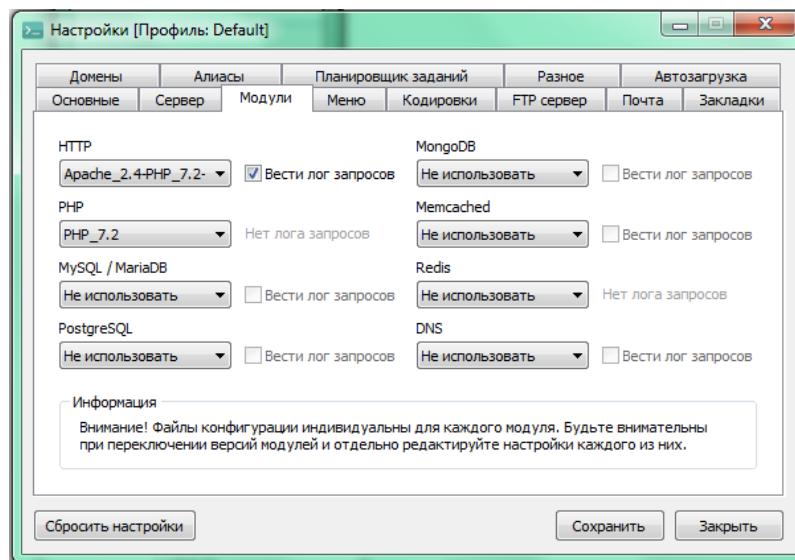


Рисунок 4. Переход во вкладку "Модули"

На следующем этапе необходимо перейти в директорию «OSPanel», после чего открыть папку «domains». В данной папке создается новая директория с именем «gallery», предназначенная для размещения файлов будущего проекта (см. рис. 5).

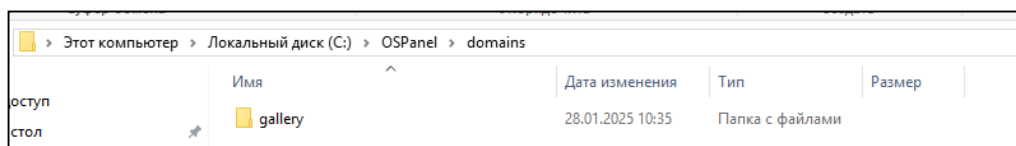


Рисунок 5. Переход к нужной директории

Теперь необходимо перейти в раздел настроек, где требуется настроить параметры домена. Для этого в соответствующие поля вводятся данные, аналогичные представленным на рисунке (рис. 6). После завершения ввода данных необходимо нажать кнопку «Добавить» и сохранить внесенные изменения.

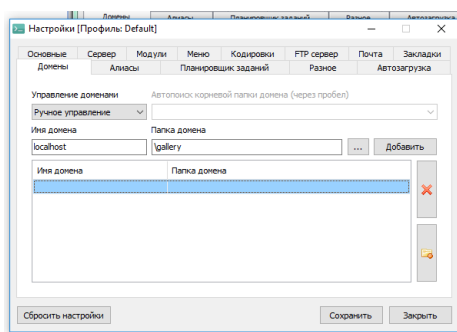


Рисунок 6. Настройка домена

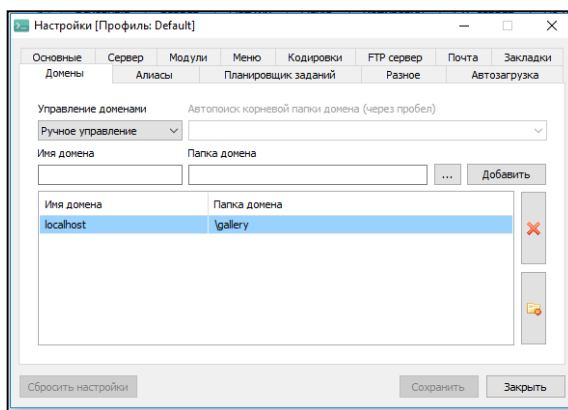


Рисунок 7. Сохранение добавленного домена

Проект будет состоять из нескольких файлов:

- **index.php** — главная страница галереи, которая отображает загруженные изображения и форму для загрузки новых.
- **upload.php** — обработчик загрузки изображений.
- **delete.php** — обработчик удаления изображений.
- **style.css** — файл стилей для оформления галереи.
- **script.js** — скрипт для работы с модальным окном и взаимодействия с изображениями.

Создадим главную страницу `index.php`, которая будет отвечать за отображение загруженных изображений и предоставлять форму для загрузки новых файлов. Для этого в текстовом редакторе Notepad++ создается новый файл, который сохраняется в директории `gallery` (рис.8). Он начинается с PHP-кода, который определяет папку для хранения изображений (`uploads/`) и получает список всех изображений в этой папке с помощью функции `glob`. Далее создается HTML-структура страницы, включающая заголовок, форму для загрузки новых изображений и блок для отображения галереи. Если изображения есть, они выводятся в виде сетки с помощью цикла `foreach`. Если изображений нет, отображается сообщение об их отсутствии. Внизу страницы будет добавлено модальное окно для просмотра изображений в увеличенном виде, которое управляется JavaScript.

```
<?php
// Папка для хранения изображений
$upload_dir = 'uploads/';

// Получаем список файлов в папке с расширениями jpg, jpeg, png, gif
$images = glob($upload_dir . "*. {jpg,jpeg,png,gif}", GLOB_BRACE);
?>

<!DOCTYPE html>
<html lang="ru">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Веб-галерея</title>
  <link rel="stylesheet" href="css/style.css"> <!-- Подключаем файл стилей -->
</head>
<body>
```

```

<div class="container">
  <h1>Моя веб-галерея</h1> <!-- Заголовок страницы -->

  <!-- Форма для загрузки изображений -->
  <form action="upload.php" method="post" enctype="multipart/form-data" class="upload-form">
    <input type="file" name="image" accept="image/*" required> <!-- Поле для выбора файла -->
    <button type="submit">Загрузить</button> <!-- Кнопка для отправки формы -->
  </form>

  <!-- Галерея изображений -->
  <div class="gallery">
    <?php if (count($images) > 0): ?> <!-- Проверяем, есть ли изображения -->
      <?php foreach ($images as $index => $image): ?> <!-- Цикл для отображения изображений -->
        <div class="gallery-item" data-index="<?php echo $index; ?>">
           <!-- Вывод изображения -->
        </div>
      <?php endforeach; ?>
    <?php else: ?> <!-- Если изображений нет -->
      <p>В галерее пока нет изображений.</p>
    <?php endif; ?>
  </div>

  <!-- Модальное окно для просмотра изображений -->
  <div id="modal" class="modal">
    <span class="close">&times;</span> <!-- Кнопка закрытия модального окна -->
    <span class="prev">&lt;</span> <!-- Кнопка "Предыдущее изображение" -->
    <span class="next">&gt;</span> <!-- Кнопка "Следующее изображение" -->
    <img id="modal-image" src="" alt="Изображение"> <!-- Место для увеличенного изображения -->
    <button id="delete-btn">Удалить</button> <!-- Кнопка удаления изображения -->
  </div>
</div>

<script src="js/script.js"></script> <!-- Подключаем JavaScript для работы с модальным окном -->
</body>
</html>

```

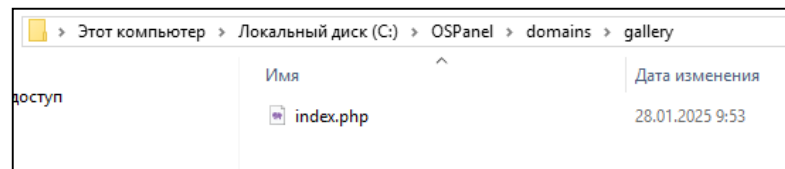


Рисунок 9. Создание файла index.php

```

1 <?php
2 // Папка для хранения изображений
3 $upload_dir = 'uploads/';
4
5 // Получаем список файлов в папке с расширениями jpg, jpeg, png, gif
6 $images = glob($upload_dir . "{.jpg,.jpeg,.png,.gif}", GLOB_BRACE);
7
8
9 <DOCTYPE html>
10 <html lang="ru">
11 <head>
12 <meta charset="UTF-8">
13 <meta name="viewport" content="width=device-width, initial-scale=1.0">
14 <title>Веб-галерея</title>
15 <link rel="stylesheet" href="css/style.css"> <!-- Подключаем файл стилей -->
16 </head>
17 <body>
18 <div class="container">
19 <h1>Моя веб-галерея</h1> <!-- Заголовок страниц -->
20
21 <!-- Форма для загрузки изображений -->
22 <form action="upload.php" method="post" enctype="multipart/form-data" class="upload-form">
23 <input type="file" name="image" accept="image/*" required> <!-- Поле для выбора файла -->
24 <button type="submit">Загрузить</button> <!-- Кнопка для отправки формы -->
25 </form>
26
27 <!-- Галерея изображений -->
28 <div class="gallery">
29 <?php if (count($images) > 0): <!-- Проверим, есть ли изображения -->
30 <?php foreach ($images as $index => $image): <!-- Цикл для отображения изображений -->
31 <div class="gallery-item" data-index="<?php echo $index; ?>">
32  <!-- Вывод изображения -->
33 </div>
34 <?php endforeach; ?>
35 <?php else: ?> <!-- Если изображений нет -->
36 <p>В галерее пока нет изображений.</p>
37 <?php endif; ?>
38 </div>
39
40 <!-- Модальное окно для просмотра изображений -->
41 <div id="modal" class="modal">
42 <span class="close">&times;</span> <!-- Кнопка закрытия модального окна -->
43 <span class="prev">&lt;</span> <!-- Кнопка "Предыдущее изображение" -->
44 <span class="next">&gt;</span> <!-- Кнопка "Следующее изображение" -->
45 <img id="modal-image" src="" alt="Изображение" /> <!-- Место для увеличенного изображения -->
46 <button id="delete-btn">Удалить</button> <!-- Кнопка удаления изображения -->
47 </div>
48
49 <script src="js/script.js"></script> <!-- Подключаем JavaScript для работы с модальным окном -->
50 </body>
51 </html>

```

Рисунок 10. Создание файла index.php

Для проверки корректности работы созданного файла необходимо запустить локальный сервер Open Server. Для этого на панели управления Open Server выбираем соответствующий значок и активируем запуск сервера.

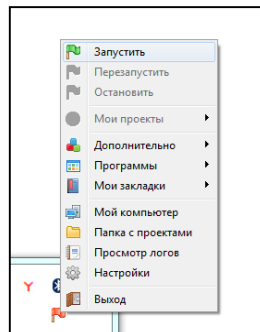


Рисунок 11. Запуск Open Server

После запуска сервера необходимо перейти во вкладку «Мои проекты» и выбрать пункт «localhost». В результате откроется веб-интерфейс, где будет доступен созданный сайт для дальнейшего тестирования и взаимодействия.

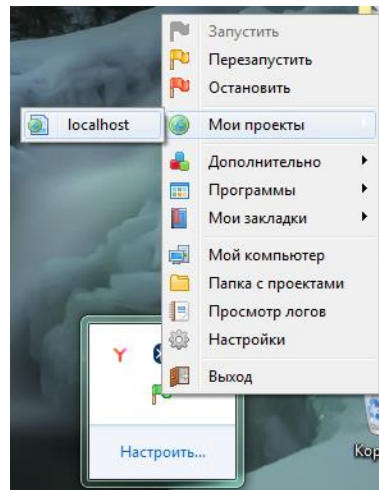


Рисунок 12. Переход на вкладку localhost

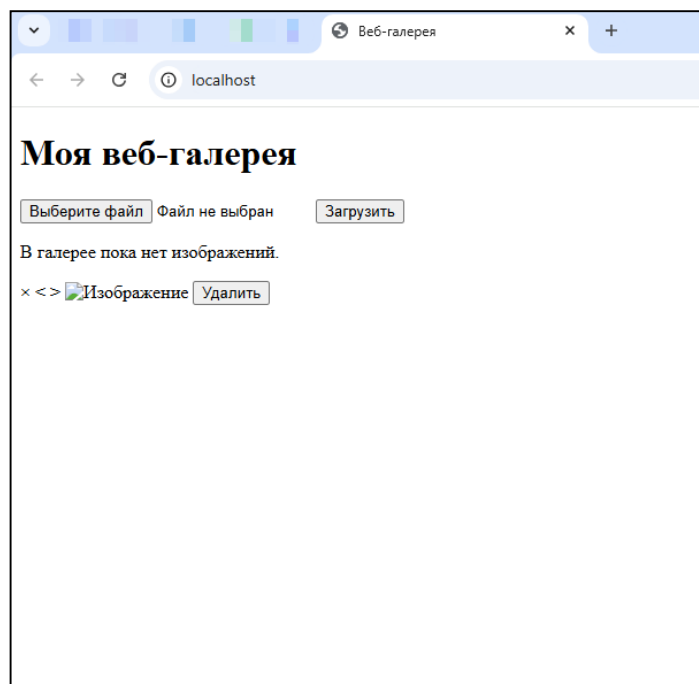


Рисунок 13. Страница index.php

Теперь создадим файл `upload.php`, который будет отвечать за обработку загрузки изображений на сервер. Он проверяет, был ли файл успешно загружен через форму на главной странице. Если ошибок нет, файл перемещается из временной папки в папку `uploads/` с помощью функции `move_uploaded_file`. После успешной загрузки пользователь перенаправляется на главную страницу (`index.php`). Если возникают ошибки, например, файл не был загружен, выводится соответствующее сообщение.

```
<?php
// Папка для загрузки изображений
$upload_dir = 'uploads/';

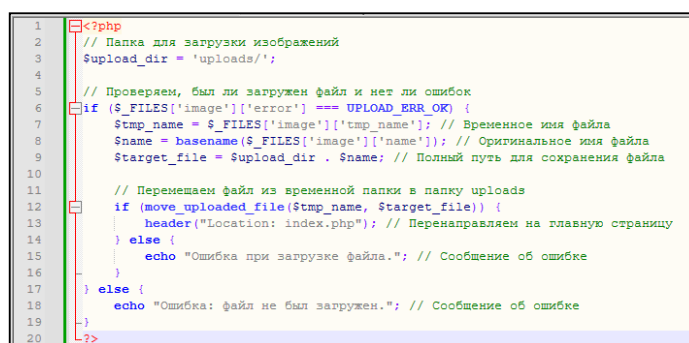
// Проверяем, был ли загружен файл и нет ли ошибок
```

```

if ($_FILES['image']['error'] === UPLOAD_ERR_OK) {
    $tmp_name = $_FILES['image']['tmp_name']; // Временное имя файла
    $name = basename($_FILES['image']['name']); // Оригинальное имя файла
    $target_file = $upload_dir . $name; // Полный путь для сохранения файла

    // Перемещаем файл из временной папки в папку uploads
    if (move_uploaded_file($tmp_name, $target_file)) {
        header("Location: index.php"); // Перенаправляем на главную страницу
    } else {
        echo "Ошибка при загрузке файла."; // Сообщение об ошибке
    }
} else {
    echo "Ошибка: файл не был загружен."; // Сообщение об ошибке
}
?>

```



```

1 <?php
2 // Папка для загрузки изображений
3 $upload_dir = 'uploads/';
4
5 // Проверяем, был ли загружен файл и нет ли ошибок
6 if ($_FILES['image']['error'] === UPLOAD_ERR_OK) {
7     $tmp_name = $_FILES['image']['tmp_name']; // Временное имя файла
8     $name = basename($_FILES['image']['name']); // Оригинальное имя файла
9     $target_file = $upload_dir . $name; // Полный путь для сохранения файла
10
11     // Перемещаем файл из временной папки в папку uploads
12     if (move_uploaded_file($tmp_name, $target_file)) {
13         header("Location: index.php"); // Перенаправляем на главную страницу
14     } else {
15         echo "Ошибка при загрузке файла."; // Сообщение об ошибке
16     }
17 } else {
18     echo "Ошибка: файл не был загружен."; // Сообщение об ошибке
19 }
20 ?>

```

Рисунок 14. Код файла upload.php

Проверим работу файла, нажимаем на кнопку «Выбрать изображение», а затем на кнопку «Загрузить» файл.

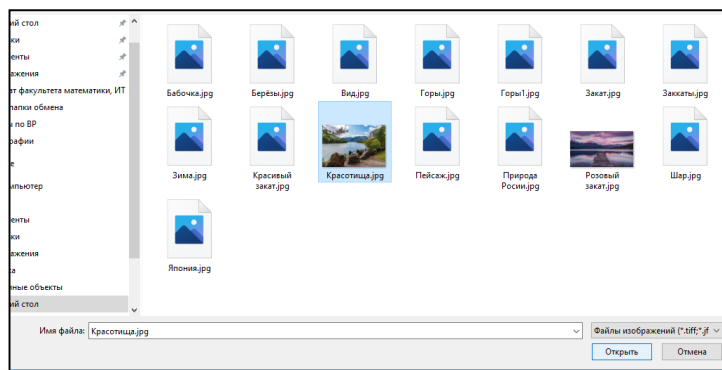


Рисунок 15. Загрузка картинки

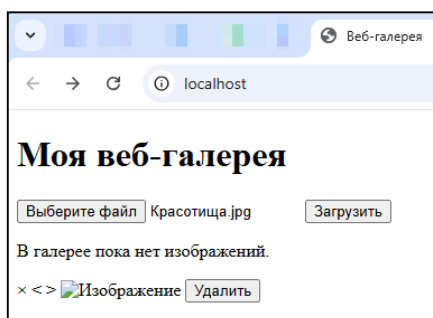


Рисунок 16. Загрузка картинки

После загрузки картинки, появляется ошибка, приступим к её устранению.

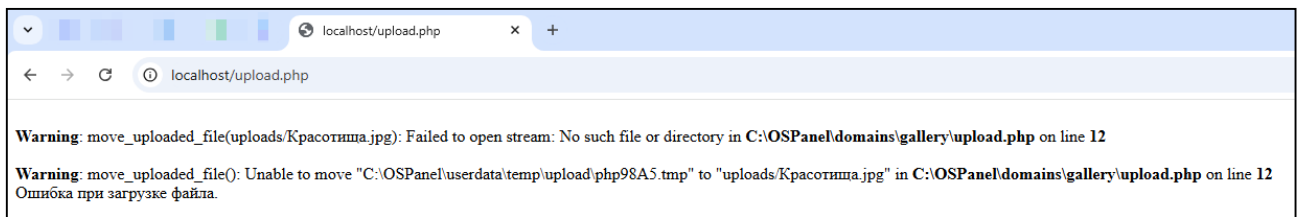


Рисунок 17. Ошибка загрузки картинки

Эта ошибка возникает из-за проблем с перемещением загруженного файла из временной папки в целевую папку uploads. Создадим её:

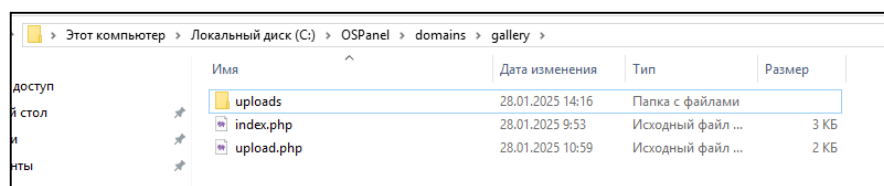


Рисунок 18. Ошибка загрузки изображения

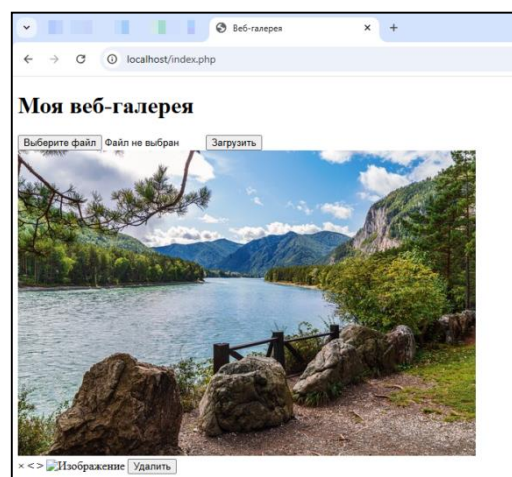


Рисунок 19. Загрузка изображения

Создадим файл `delete.php`, который будет обрабатывать удаление изображений. Он проверяет, был ли передан параметр `file` (имя файла для удаления). Если файл существует в папке `uploads/`, он удаляется с помощью функции `unlink`. После удаления пользователь перенаправляется на главную страницу. Этот файл вызывается из JavaScript при нажатии кнопки "Удалить" в модальном окне.

```
<?php
// Папка с изображениями
$upload_dir = 'uploads/';

// Проверяем, был ли передан параметр file (имя файла для удаления)
if (isset($_GET['file'])) {
```

```

$file = $upload_dir . $_GET['file']; // Полный путь к файлу

// Проверяем, существует ли файл
if (file_exists($file)) {
    unlink($file); // Удаляем файл
}

// Перенаправляем на главную страницу
header("Location: index.php");
?>

```

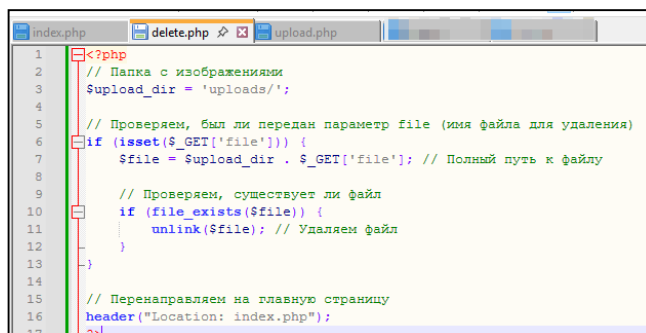


Рисунок 20. Код файла delete.php

Затем создадим файл `style.css`, который будет содержать стили для оформления галереи. Он задает общие стили для страницы, такие как шрифты, цвета и отступы. Форма загрузки стилизована с помощью кнопки и поля для выбора файла. Галерея изображений отображается в виде сетки с использованием CSS Grid. Каждое изображение имеет эффект увеличения при наведении. Модальное окно для просмотра изображений стилизовано так, чтобы занимать весь экран, с кнопками для закрытия, листания и удаления изображений.

```

/* Основные стили для страницы */
body {
    font-family: Arial, sans-serif; /* Шрифт */
    background-color: #f4f4f9; /* Цвет фона */
    color: #333; /* Цвет текста */
    margin: 0; /* Убираем отступы */
    padding: 0;
}

.container {
    max-width: 1200px; /* Максимальная ширина контейнера */
    margin: 0 auto; /* Центрирование контейнера */
    padding: 20px; /* Внутренние отступы */
    text-align: center; /* Выравнивание текста по центру */
}

h1 {
    color: #444; /* Цвет заголовка */
}

```

```
margin-bottom: 20px; /* Отступ снизу */
}

/* Стили для формы загрузки */
.upload-form {
margin-bottom: 20px; /* Отступ снизу */
}

.upload-form input[type="file"] {
padding: 10px; /* Внутренние отступы */
border: 1px solid #ccc; /* Граница */
border-radius: 5px; /* Скругление углов */
margin-right: 10px; /* Отступ справа */
}

.upload-form button {
padding: 10px 20px; /* Внутренние отступы */
background-color: #28a745; /* Цвет фона кнопки */
color: white; /* Цвет текста */
border: none; /* Убираем границу */
border-radius: 5px; /* Скругление углов */
cursor: pointer; /* Курсор в виде указателя */
}

.upload-form button:hover {
background-color: #218838; /* Цвет фона при наведении */
}

/* Стили для галереи */
.gallery {
display: grid; /* Используем CSS Grid для сетки */
grid-template-columns: repeat(auto-fill, minmax(200px, 1fr)); /* Колонки с минимальной
шириной 200px */
gap: 15px; /* Расстояние между элементами */
margin-bottom: 20px; /* Отступ снизу */
}

.gallery-item {
position: relative; /* Относительное позиционирование */
overflow: hidden; /* Скрываем выходящие за пределы элементы */
border-radius: 10px; /* Скругление углов */
box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1); /* Тень */
transition: transform 0.3s ease, box-shadow 0.3s ease; /* Анимация при наведении */
}

.gallery-item:hover {
transform: scale(1.05); /* Увеличение при наведении */
box-shadow: 0 6px 12px rgba(0, 0, 0, 0.2); /* Увеличение тени */
}

.gallery-item img {
width: 100%; /* Ширина изображения */
```

```
height: 200px; /* Высота изображения */
object-fit: cover; /* Обрезка изображения по размеру */
display: block; /* Блочное отображение */
cursor: pointer; /* Курсор в виде указателя */
}

/* Стили для модального окна */
.modal {
display: none; /* Скрываем модальное окно по умолчанию */
position: fixed; /* Фиксированное позиционирование */
z-index: 1000; /* Высокий z-index для отображения поверх других элементов */
left: 0;
top: 0;
width: 100%;
height: 100%;
background-color: rgba(0, 0, 0, 0.9); /* Полупрозрачный черный фон */
justify-content: center; /* Центрирование по горизонтали */
align-items: center; /* Центрирование по вертикали */
}

.modal img {
max-width: 90%; /* Максимальная ширина изображения */
max-height: 90%; /* Максимальная высота изображения */
border-radius: 10px; /* Скругление углов */
}

.close {
position: absolute; /* Абсолютное позиционирование */
top: 20px;
right: 30px;
color: white; /* Цвет текста */
font-size: 40px; /* Размер текста */
cursor: pointer; /* Курсор в виде указателя */
}

.prev, .next {
position: absolute; /* Абсолютное позиционирование */
top: 50%; /* Позиционирование по вертикали */
color: white; /* Цвет текста */
font-size: 40px; /* Размер текста */
cursor: pointer; /* Курсор в виде указателя */
user-select: none; /* Запрет выделения текста */
}

.prev {
left: 30px; /* Позиционирование слева */
}

.next {
right: 30px; /* Позиционирование справа */
}
```

```
#delete-btn {
    position: absolute; /* Абсолютное позиционирование */
    bottom: 20px; /* Позиционирование снизу */
    left: 50%; /* Центрирование по горизонтали */
    transform: translateX(-50%); /* Сдвиг для точного центрирования */
    padding: 10px 20px; /* Внутренние отступы */
    background-color: #dc3545; /* Цвет фона */
    color: white; /* Цвет текста */
    border: none; /* Убираем границу */
    border-radius: 5px; /* Скругление углов */
    cursor: pointer; /* Курсор в виде указателя */
}

#delete-btn:hover {
    background-color: #c82333; /* Цвет фона при наведении */
}
```

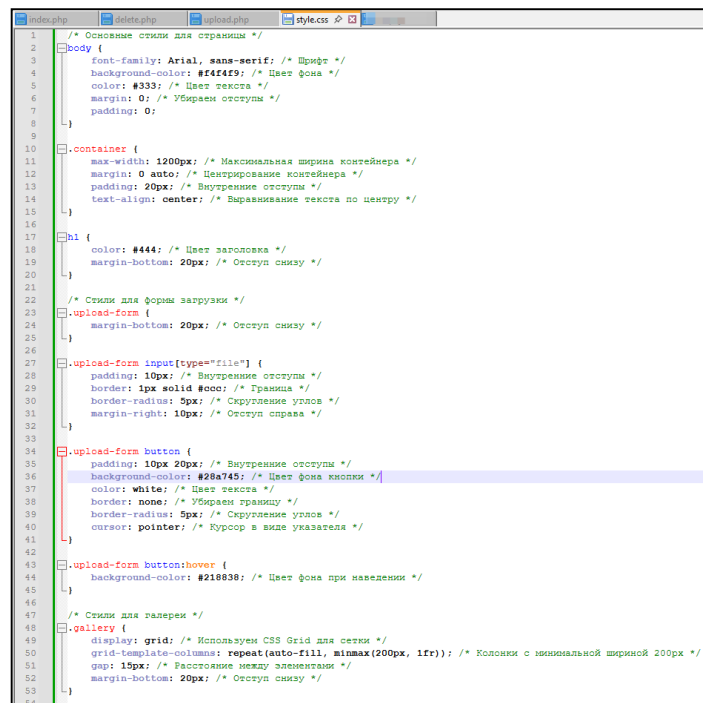


Рисунок 21. Код файла style.css

```

index.php delete.php upload.php style.css
55 .gallery-item {
56     position: relative; /* Относительное позиционирование */
57     overflow: hidden; /* Скрываем выходящие за пределы элементы */
58     border-radius: 10px; /* Скругление углов */
59     box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1); /* Тень */
60     transition: transform 0.3s ease, box-shadow 0.3s ease; /* Анимация при наведении */
61 }
62
63 .gallery-item:hover {
64     transform: scale(1.05); /* Увеличение при наведении */
65     box-shadow: 0 6px 12px rgba(0, 0, 0, 0.2); /* Увеличение тени */
66 }
67
68 .gallery-item img {
69     width: 100%; /* Ширина изображения */
70     height: 200px; /* Высота изображения */
71     object-fit: cover; /* Обрезка изображения по размеру */
72     display: block; /* Блочное отображение */
73     cursor: pointer; /* Курсор в виде указателя */
74 }
75
76 /* Стили для модального окна */
77 .modal {
78     display: none; /* Скрываем модальное окно по умолчанию */
79     position: fixed; /* Фиксированное позиционирование */
80     z-index: 1000; /* Высокий z-index для отображения поверх других элементов */
81     left: 0;
82     top: 0;
83     width: 100%;
84     height: 100%;
85     background-color: rgba(0, 0, 0, 0.9); /* Полупрозрачный черный фон */
86     justify-content: center; /* Центрирование по горизонтали */
87     align-items: center; /* Центрирование по вертикали */
88 }
89
90 .modal img {
91     max-width: 90%; /* Максимальная ширина изображения */
92     max-height: 90%; /* Максимальная высота изображения */
93     border-radius: 10px; /* Скругление углов */
94 }
95
96 .close {
97     position: absolute; /* Абсолютное позиционирование */
98     top: 20px;
99     right: 30px;
100     color: white; /* Цвет текста */
101     font-size: 40px; /* Размер текста */
102     cursor: pointer; /* Курсор в виде указателя */
103 }
104
105 .prev, .next {
106     position: absolute; /* Абсолютное позиционирование */
107     top: 50%; /* Позиционирование по вертикали */
108     color: white; /* Цвет текста */

```

Рисунок 22. Код файла style.css

```

index.php delete.php upload.php style.css script.js
88
89
90 .modal img {
91     max-width: 90%; /* Максимальная ширина изображения */
92     max-height: 90%; /* Максимальная высота изображения */
93     border-radius: 10px; /* Скругление углов */
94 }
95
96 .close {
97     position: absolute; /* Абсолютное позиционирование */
98     top: 20px;
99     right: 30px;
100     color: white; /* Цвет текста */
101     font-size: 40px; /* Размер текста */
102     cursor: pointer; /* Курсор в виде указателя */
103 }
104
105 .prev, .next {
106     position: absolute; /* Абсолютное позиционирование */
107     top: 50%; /* Позиционирование по вертикали */
108     color: white; /* Цвет текста */
109     font-size: 40px; /* Размер текста */
110     cursor: pointer; /* Курсор в виде указателя */
111     user-select: none; /* Запрет выделения текста */
112 }
113
114 .prev {
115     left: 30px; /* Позиционирование слева */
116 }
117
118 .next {
119     right: 30px; /* Позиционирование справа */
120 }
121
122 #delete-btn {
123     position: absolute; /* Абсолютное позиционирование */
124     bottom: 20px; /* Позиционирование снизу */
125     left: 50%; /* Центрирование по горизонтали */
126     transform: translateX(-50%); /* Сдвиг для точного центрирования */
127     padding: 10px 20px; /* Внутренние отступы */
128     background-color: #dc3545; /* Цвет фона */
129     color: white; /* Цвет текста */
130     border: none; /* Убираем границу */
131     border-radius: 5px; /* Скругление углов */
132     cursor: pointer; /* Курсор в виде указателя */
133 }
134
135 #delete-btn:hover {
136     background-color: #c82333; /* Цвет фона при наведении */
137 }

```

Рисунок 23. Код файла style.css

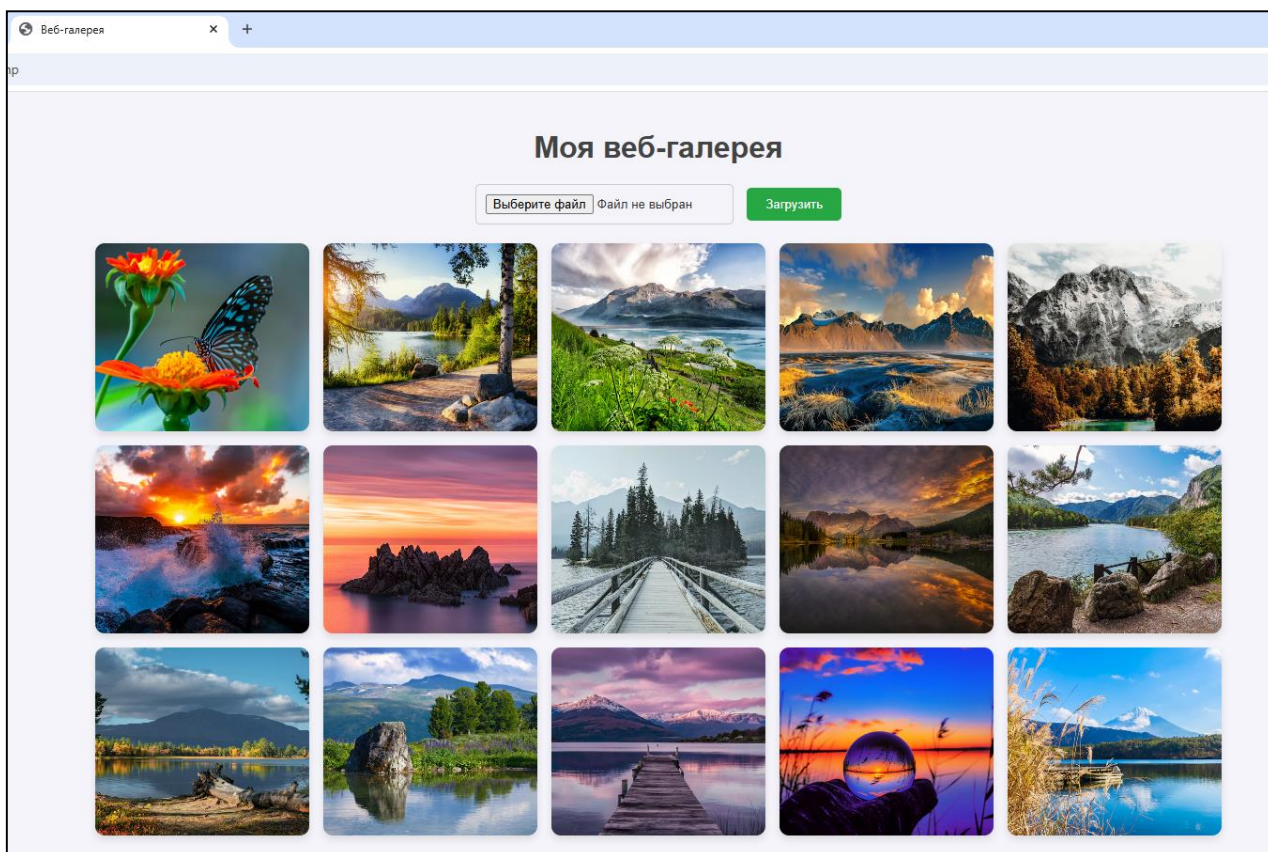


Рисунок 24. Главная страница с применением CSS стилей

Далее создадим файл `script.js`, который будет добавлять интерактивность галерее. Он отслеживает клики по изображениям, открывая модальное окно с увеличенной версией изображения. В модальном окне можно листать изображения с помощью кнопок «Вперед» и «Назад». Также добавлена кнопка «Удалить», которая вызывает подтверждение и, если пользователь согласен, отправляет запрос на удаление изображения через `delete.php`. После удаления страница автоматически обновляется, чтобы отразить изменения в галерее.

// Получаем элементы DOM

```
const galleryItems = document.querySelectorAll('.gallery-item'); // Все элементы галереи
const modal = document.getElementById('modal'); // Модальное окно
const modalImage = document.getElementById('modal-image'); // Изображение в модальном окне
const closeBtn = document.querySelector('.close'); // Кнопка закрытия
const prevBtn = document.querySelector('.prev'); // Кнопка "Предыдущее изображение"
const nextBtn = document.querySelector('.next'); // Кнопка "Следующее изображение"
const deleteBtn = document.getElementById('delete-btn'); // Кнопка удаления
```

```
let currentIndex = 0; // Текущий индекс изображения
```

```
// Открытие модального окна при клике на изображение
galleryItems.forEach((item, index) => {
  item.addEventListener('click', () => {
    currentIndex = index; // Устанавливаем текущий индекс
```

```
        updateModalImage(); // Обновляем изображение в модальном окне
        modal.style.display = 'flex'; // Показываем модальное окно
    });
});

// Закрытие модального окна
closeBtn.addEventListener('click', () => {
    modal.style.display = 'none'; // Скрываем модальное окно
});

// Листание влево (предыдущее изображение)
prevBtn.addEventListener('click', () => {
    currentIndex = (currentIndex > 0) ? currentIndex - 1 : galleryItems.length - 1; // Обновляем
индекс
    updateModalImage(); // Обновляем изображение
});

// Листание вправо (следующее изображение)
nextBtn.addEventListener('click', () => {
    currentIndex = (currentIndex < galleryItems.length - 1) ? currentIndex + 1 : 0; // Обновляем
индекс
    updateModalImage(); // Обновляем изображение
});

// Удаление изображения
deleteBtn.addEventListener('click', () => {
    const imageSrc = modalImage.src.split('/').pop(); // Получаем имя файла
    if (confirm('Вы уверены, что хотите удалить это изображение?')) {
        window.location.href = `delete.php?file=${imageSrc}`; // Перенаправляем на delete.php
    }
});

// Обновление изображения в модальном окне
function updateModalImage() {
    const imageSrc = galleryItems[currentIndex].querySelector('img').src; // Получаем src
текущего изображения
    modalImage.src = imageSrc; // Устанавливаем src для модального окна
}
```

```

1 // Получен элементы DOM
2 const galleryItems = document.querySelectorAll('.gallery-item'); // Все элементы галереи
3 const modal = document.getElementById('modal'); // Модальное окно
4 const modalImage = document.getElementById('modal-image'); // Изображение в модальном окне
5 const closeButton = document.querySelector('.close'); // Кнопка закрытия
6 const prevBtn = document.querySelector('.prev'); // Кнопка "Предыдущее изображение"
7 const nextBtn = document.querySelector('.next'); // Кнопка "Следующее изображение"
8 const deleteBtn = document.getElementById('delete-btn'); // Кнопка удаления
9
10 let currentIndex = 0; // Текущий индекс изображения
11
12 // Открытие модального окна при клике на изображение
13 galleryItems.forEach((item, index) => {
14   item.addEventListener('click', () => {
15     currentIndex = index; // Устанавливаем текущий индекс
16     updateModalImage(); // Обновляем изображение в модальном окне
17     modal.style.display = 'flex'; // Показываем модальное окно
18   });
19 });
20
21 // Закрытие модального окна
22 closeButton.addEventListener('click', () => {
23   modal.style.display = 'none'; // Скрываем модальное окно
24 });
25
26 // Листание влево (предыдущее изображение)
27 prevBtn.addEventListener('click', () => {
28   currentIndex = (currentIndex > 0) ? currentIndex - 1 : galleryItems.length - 1; // Обновляем индекс
29   updateModalImage(); // Обновляем изображение
30 });
31
32 // Листание вправо (следующее изображение)
33 nextBtn.addEventListener('click', () => {
34   currentIndex = (currentIndex < galleryItems.length - 1) ? currentIndex + 1 : 0; // Обновляем индекс
35   updateModalImage(); // Обновляем изображение
36 });
37
38 // Удаление изображения
39 deleteBtn.addEventListener('click', () => {
40   const imageUrl = modalImage.src.split('/').pop(); // Получаем имя файла
41   if (confirm('Вы уверены, что хотите удалить это изображение?')) {
42     window.location.href = 'delete.php?file=' + imageUrl; // Перенаправляем на delete.php
43   }
44 });
45
46 // Обновление изображения в модальном окне
47 function updateModalImage() {
48   const imageUrl = galleryItems[currentIndex].querySelector('img').src; // Получаем src текущего изображения
49   modalImage.src = imageUrl; // Устанавливаем src для модального окна
50 }

```

Рисунок 25. Код файла script.js

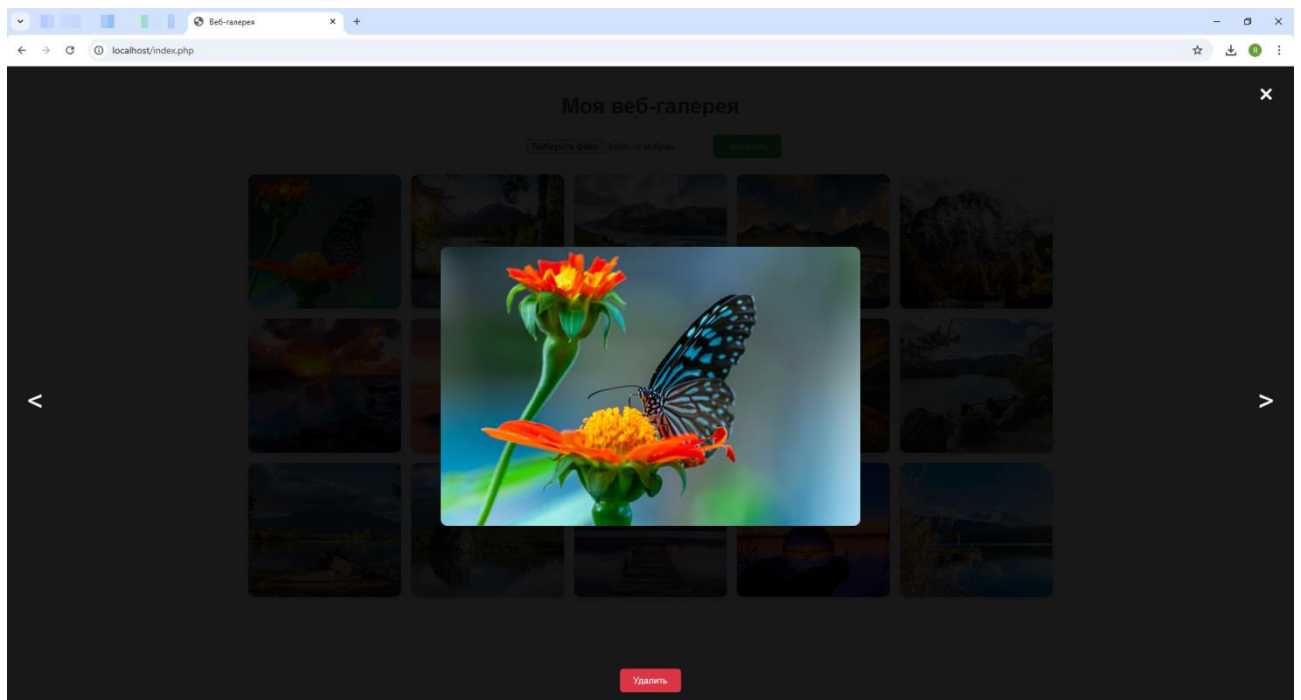


Рисунок 26. Результат добавления script.js

Теперь можно листать фотографии влево и вправо, а также удалять ненужные изображения. После нажатия на кнопку «Удалить» появляется подтверждение, действительно ли вы хотите удалить изображение.

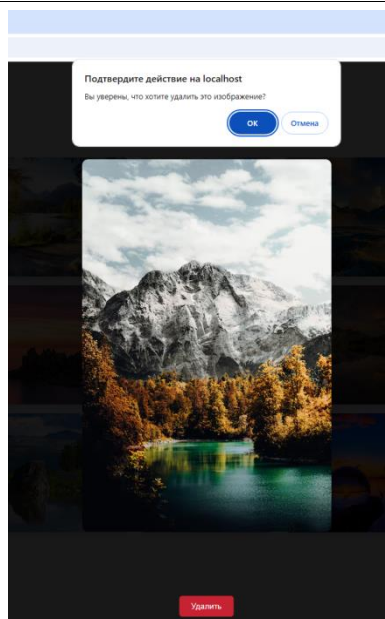


Рисунок 27. Подтверждение удаления изображения

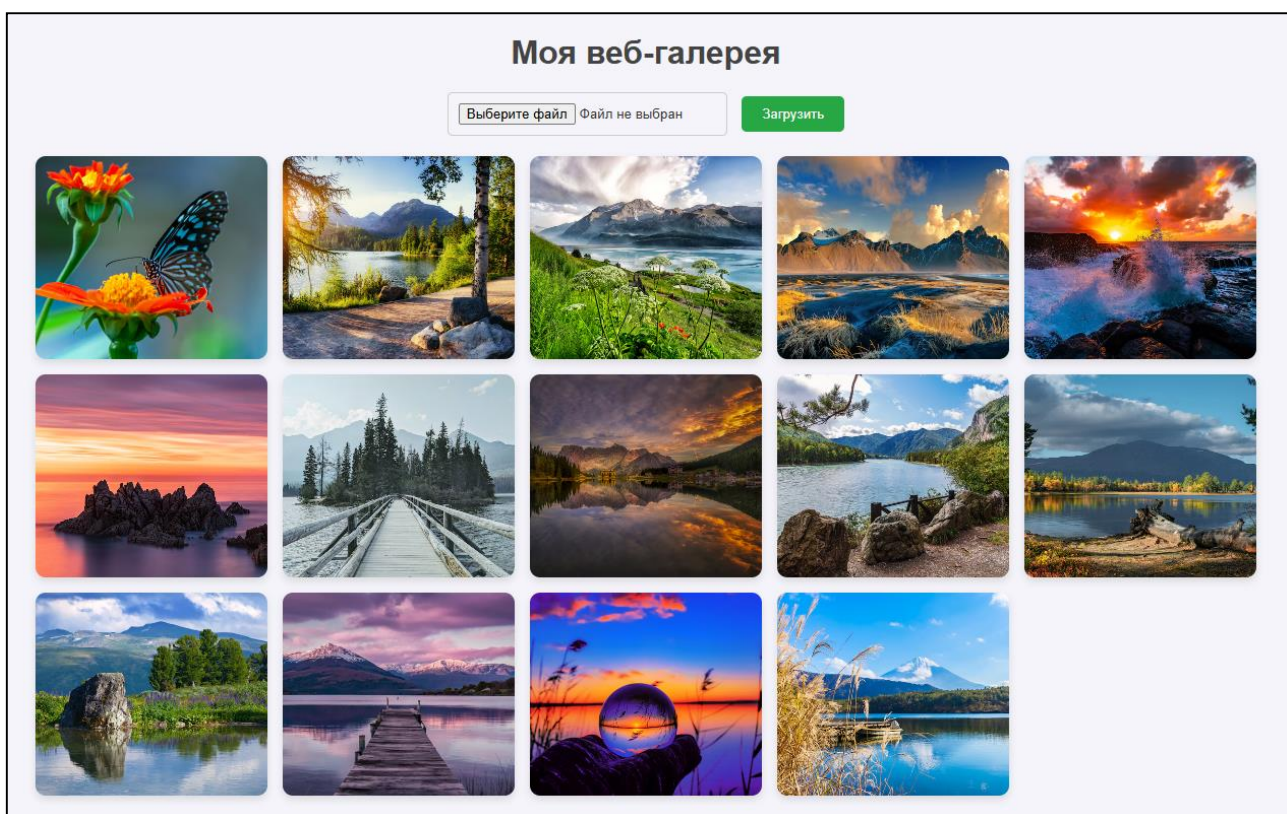


Рисунок 28. После удаления изображения

4 Выводы

В данной работе подробно рассмотрели, как создать современную галерею с изображениями на PHP с использованием HTML, CSS и JavaScript. Были реализованы ключевые функции, такие как загрузка, отображение, просмотр и удаление изображений, а также добавлены интерактивные элементы, улучшающие пользовательский опыт.

Использование PHP обеспечило надежную обработку данных на стороне сервера, HTML и CSS позволили создать адаптивный и визуально привлекательный интерфейс, а JavaScript добавил интерактивность, включая модальное окно для просмотра изображений и листание фотографий.

Данная работа может быть использована в образовательных учреждениях как методическое пособие для студентов.

Библиографический список

1. Гарпушкина К. В., Карасёва Г. Л. Разработка web-сайта библиотека с использованием php, html, css и javascript //Новые математические методы и компьютерные технологии в проектировании, производстве и научных исследованиях. 2018. С. 211-211.
2. Иванова А. В. Разработка web-приложения для оператора сотовой связи с возможностью подбора тарифного плана //Новая наука: Опыт, традиции, инновации. 2016. №. 5-2. С. 109-111.
3. Эль-гади А. Х., Орлов В. В. Разработка адаптивной части веб-сайта «Стоматологическая клиника» с использованием html 5, css 3, php и javascript. 2022.
4. Лузан С. Н., Орлов В. В. Разработка адаптивной части веб-сайта «Станция технического обслуживания автомобилей» с использованием html 5, css 3, php и javascript. 2022.
5. Бабенко А. А., Ушакова С. Н. Разработка web-приложения" Кинотеатра" с использованием JavaScript, PHP и MYSQL //Мировая наука. 2018. №. 12 (21). С. 140-144.