Lab 5 PHP

1. Запись и чтение из файла

 Функция fclose() важна по нескольким причинам:

* **Освобождение ресурсов**: Когда вы открываете файл, операционная система выделяет ресурсы для этого файла. Закрытие файла освобождает эти ресурсы.
* **Предотвращение ошибок**: Если файл остается открытым после завершения работы с ним, это может привести к ошибкам или повреждению данных, если другие процессы или операции попытаются использовать этот файл.
* **Сохранение данных**: В некоторых случаях, когда вы записываете данные в файл, они могут быть временно сохранены в буфере для повышения производительности. Закрытие файла гарантирует, что все данные из буфера записываются в файл.

<?php

// Создание файла

$file = fopen("file.txt", "w") or die("Ошибка создания файла!");

// Ввод данных в файл

fwrite($file, "1. William Smith, 1990, 2344455666677\n");

fwrite($file, "2. John Doe, 1988, 4445556666787\n");

fwrite($file, "3. Michael Brown, 1991, 7748956996777\n");

fwrite($file, "4. David Johnson, 1987, 5556667779999\n");

fwrite($file, "5. Robert Jones, 1992, 99933456678888\n");

// Закрытие файла

fclose($file);

// Открытие файла для добавления данных

$file = fopen("file.txt", "a") or die("Ошибка открытия файла для добавления данных!");

if (!$file) {

echo("Не был найден файл для добавления данных!");

} else {

// Добавление в файл еще 3 записей

fwrite($file, "6. James Davis, 1993, 1234567890123\n");

fwrite($file, "7. Brian Wilson, 1994, 2345678901234\n");

fwrite($file, "8. Richard Taylor, 1995, 3456789012345\n");

}

// Закрытие файла

fclose($file);

// Открытие файла для чтения из него

$file = fopen("file.txt", "r") or die("Ошибка открытия файла для чтения!");

if (!$file) {

echo("Не был найден файл для чтения данных!");

} else { ?>

<div>Данные из файла: </div>

<?php

while (!feof($file)) {

echo fgets($file); ?>

<br/>

<?php

}

fclose($file);

}

?>

2. Запись в файл с помощью функции file\_get\_contents()

<?php

// Создание файла и ввод данных в файл

file\_put\_contents("file.txt", "1. William Smith, 1990, 2344455666677\n");

file\_put\_contents("file.txt", "2. John Doe, 1988, 4445556666787\n", FILE\_APPEND);

file\_put\_contents("file.txt", "3. Michael Brown, 1991, 7748956996777\n", FILE\_APPEND);

file\_put\_contents("file.txt", "4. David Johnson, 1987, 5556667779999\n", FILE\_APPEND);

file\_put\_contents("file.txt", "5. Robert Jones, 1992, 99933456678888\n", FILE\_APPEND);

// Добавление в файл еще 3 записей

file\_put\_contents("file.txt", "6. James Davis, 1993, 1234567890123\n", FILE\_APPEND);

file\_put\_contents("file.txt", "7. Brian Wilson, 1994, 2345678901234\n", FILE\_APPEND);

file\_put\_contents("file.txt", "8. Richard Taylor, 1995, 3456789012345\n", FILE\_APPEND);

// Чтение из файла

$file = fopen("file.txt", "r") or die("Ошибка открытия файла для чтения!");

if (!$file) {

echo("Не был найден файл для чтения данных!");

} else { ?>

<div>Данные из файла: </div>

<?php

while (!feof($file)) {

echo fgets($file); ?>

<br/>

<?php

}

fclose($file);

}

?>

Отличия между fwrite() и file\_put\_contents():

* **Удобство использования**: file\_put\_contents() проще в использовании, поскольку она автоматически открывает файл, записывает данные и закрывает файл. С другой стороны, fwrite() требует явного открытия файла с помощью fopen() и закрытия файла с помощью fclose().
* **Безопасность**: file\_put\_contents() является более безопасной функцией, поскольку она блокирует файл во время записи, предотвращая одновременную запись в файл из других процессов.
* **Гибкость**: fwrite() более гибкая функция, поскольку она позволяет вам контролировать, как именно вы хотите записать данные в файл. С помощью fwrite() вы можете выбрать, хотите ли вы записать данные в начало файла, в конец файла или в любое другое место. С другой стороны, file\_put\_contents() всегда записывает данные в начало файла, если не указан флаг FILE\_APPEND.

3. Обработка форм и файлов

<?php if (!isset($\_REQUEST['start'])) { ?>

<form action="<?php echo $\_SERVER['SCRIPT\_NAME'] ?>" method="post">

<div>

<label>Ваше имя: <input name="name" type="text" size="30"></label>

</div>

<div>

<label>Ваш возраст: <input name="age" type="number" min="1" max="120"></label>

</div>

<div>

<label>Ваш E-mail: <input name="email" type="email"></label>

</div>

<div>

<label>Ваше мнение о нас напишите тут:

<textarea name="message" cols="40" rows="4" placeholder="Ваше

мнение..."></textarea>

</label>

</div>

<div>

<input type="reset" value="Стереть"/>

<input type="submit" value="Передать" name="start"/>

</div>

</form>

<?php } else {

// Данные с формы

$data = [

'name' => $\_POST['name'] ?? "",

'age' => $\_POST['age'] ?? "",

'email' => $\_POST['email'] ?? "",

'message' => $\_POST['message'] ?? "",

];

// Сохранение данных в файл

$file = fopen('messages.txt', 'a+') or die("Недоступный файл!");

foreach ($data as $field => $value) {

fwrite($file, $field . ": " . $value . "\n");

}

fwrite($file, "\n");

fclose($file);

// Вывод данных на экран

echo 'Данные были сохранены! Вот что хранится в файле: <br />';

$file = fopen("messages.txt", "r") or die("Недоступный файл!");

while (!feof($file)) {

echo fgets($file) . "<br />";

}

fclose($file);

}

4. Регистрация и авторизация пользователей

Скрипты в проекте