

Лабораторна робота №3

Тема: Cookie. Session. Робота з файлами та каталогами

Мета роботи: навчитися працювати з файлами: створювати, читати, записувати і видаляти дані з файлів, навчитися працювати з каталогами, cookie, session, передавати файли через форми і приймати файли у PHP-скрипті

Хід роботи

Завдання 1: Робота з cookie (1 завдання)

Створіть на сторінці 3 посилання: «Великий шрифт», «Середній шрифт», «Маленький шрифт»

В залежності від того, на яке посилання натисне користувач, ви повинні встановити певний розмір шрифту, при цьому він повинен зберігатися і при переході на інші сторінки поточного сайту

У цьому завданні я створила веб-сторінку, яка дозволяє користувачеві вибирати розмір шрифту (великий, середній або маленький), і цей вибір зберігається в cookie-файлі. При наступних відвідинах сторінки застосовується збережений розмір шрифту.

Основні виконані частини:

1. Відображення поточної сторінки вибору розміру шрифту

(index.php): На сторінці index.php я спочатку перевіряю, чи встановлено cookie з назвою font_size. Якщо так, я отримую його значення; інакше встановлюю розмір шрифту за замовчуванням (16px). Потім я використовую це значення для динамічного встановлення стилю font-size для елемента <body>

					ІПТР.420001.123-ЗЛ						
Змн.	Арк.	№ докум.	Підпис	Дата							
Розроб.		Мартинюк Ю.І.			Звіт з лабораторної роботи			Лім.	Арк.	Аркушів	
Перевір.		Ковтун В.В.								1	
Керівник								ФІКТ, гр. ІПЗ-23-3			
Н. контр.											
Затверд.											

Головний момент: Динамічне застосування розміру шрифту на основі значення cookie при завантаженні сторінки, забезпечуючи збереження попереднього вибору користувача.

2. Обробка вибору розміру шрифту та встановлення Cookie

(process.php): На сторінці process.php я перевіряю, чи передано GET-параметр size. Якщо так, я отримую його значення та використовую оператор switch для визначення обраного розміру шрифту. Для кожного варіанту я встановлюю cookie з назвою font_size, відповідним значенням розміру шрифту (24px, 16px або 12px), часом життя в одну годину (time() + 3600) та шляхом доступу до cookie на всьому сайті (/):

```
<?php

if (isset($_GET['size'])) {
    $size = $_GET['size'];
    switch ($size) {
        case 'big':
            setcookie('font_size', '24px', time() + 3600, '/');
            break;
        case 'medium':
            setcookie('font_size', '16px', time() + 3600, '/');
            break;
        case 'small':
            setcookie('font_size', '12px', time() + 3600, '/');
            break;
    }
}
```

Головний момент: Використання функції setcookie() для збереження вибору користувача у cookie-файлі на його комп'ютері, що дозволяє запам'ятати налаштування між сесіями.

3. Перенаправлення назад на сторінку вибору: Після обробки вибору та встановлення cookie, сторінка process.php негайно перенаправляє користувача назад на сторінку index.php за допомогою функції

					ІРТР.420001.123-ЗЛ	Арк.
						2
Змн.	Арк.	№ докум.	Підпис	Дата		

header('Location: index.php'). Це призводить до оновлення сторінки index.php, і вже застосовується новий (або існуючий) розмір шрифту з cookie.

Виберіть розмір шрифту

[Великий шрифт](#) | [Середній шрифт](#) | [Маленький шрифт](#)
[Назад](#)

Рис.1 Результат виконання програми

Завдання 2: Робота з session (1 завдання)

- 1) Створіть форму авторизації з полями «Логін» та «Пароль»
- 2) Якщо був введений логін «Admin» та пароль «password», то вивести повідомлення «Добрий день, Admin!», інакше вивести повідомлення про невірність введення логіна або пароля
- 3) При кожному оновленні сторінки, якщо користувач вже пройшов успішно авторизацію, повинно виводитись привітання, а якщо користувач не пройшов авторизацію, або вийшов, то вивести форму авторизації

У цьому завданні я створила просту веб-сторінку, яка реалізує механізм авторизації користувача. Система використовує сесії для відстеження стану входу користувача.

Основні виконані частини:

1. **Ініціалізація сесії:** На початку скрипта я викликаю функцію session_start() для запуску або відновлення існуючої сесії. Це необхідно для роботи з суперглобальним масивом \$_SESSION

Головний момент: Використання session_start() є ключовим для активації механізму сесій у PHP.

2. **Перевірка стану авторизації:** Я перевіряю, чи встановлено змінну \$_SESSION['logged_in'] і чи її значення дорівнює true. Якщо це так,

					IPTP.420001.123-3Л	Арк.
						2
Змн.	Арк.	№ докум.	Підпис	Дата		

користувач вважається авторизованим, і йому виводиться привітання та посилання для виходу

Головний момент: Використання змінної сесії `$_SESSION['logged_in']` як прапора, що вказує на стан авторизації користувача.

3. **Обробка спроби входу:** Якщо користувач не авторизований, я перевіряю, чи був надісланий POST-запит (тобто чи була відправлена форма авторизації). Я отримую введені користувачем логін (`$_POST['username']`) та пароль (`$_POST['password']`) та порівнюю їх з жорстко заданими значеннями ('Admin' та 'password'). У разі успішної перевірки я встановлюю `$_SESSION['logged_in']` в true та зберігаю ім'я користувача в `$_SESSION['username']`. Потім я використовую `header('Location: index.php')` для перенаправлення користувача назад на цю ж сторінку, де вже відображається привітання:

```
<?php
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $username = $_POST['username'];
    $password = $_POST['password'];

    if ($username === 'Admin' && $password === 'password') {
        $_SESSION['logged_in'] = true;
        $_SESSION['username'] = $username;
        header('Location: index.php');
        exit();
    } else {
        echo "Невірний логін або пароль!";
    }
}
?>
```

Головний момент: Обробка POST-запиту з даними авторизації, перевірка облікових даних та встановлення змінних сесії у разі успіху.

					ІПТР.420001.123-3Л	Арк.
						2
Змн.	Арк.	№ докум.	Підпис	Дата		

Використання header('Location: ...') для перенаправлення користувача після успішної авторизації.

4. **Відображення форми авторизації:** Якщо користувач не авторизований і не було спроби входу (тобто не було POST-запиту), я відображаю HTML-форму з полями для введення логіна та пароля

Головний момент: Надання користувачеві інтерфейсу для введення своїх облікових даних.

5. **Реалізація виходу з системи:** Я перевіряю, чи передано GET-параметр logout зі значенням true. Якщо так, я використовую session_unset() для видалення всіх змінних сесії та session_destroy() для знищення самої сесії. Після цього користувач перенаправляється назад на головну сторінку, де відображається форма авторизації:

```
<?php
if (isset($_GET['logout']) && $_GET['logout'] === 'true') {
    session_unset();
    session_destroy();
    header('Location: index.php');
    exit();
}
?>
```

Головний момент: Реалізація механізму виходу користувача шляхом очищення та знищення сесії.

Форма авторизації

Логін:

Пароль:

Добрий день, Admin!
[Вийти](#)

					ІРТР.420001.123-ЗЛ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		2

Невірний логін або пароль!

Форма авторизації

Логін:

Пароль:

Рис.2-4 Результат виконання програми

Завдання 3: Робота з файлами (3 завдання)

1) Створіть форму з полями **Ім'я** та **Коментар**.

- Опрацюйте форму, записавши введені дані в файл з коментарями.
- На цій же сторінці виведіть всі поточні коментарі в таблицю, зчитавши їх з файла (1 коментар – 1 рядок).
- Примітка1: ви повинні подумати, в якому вигляді треба зберігати коментарі в файл, та чим розділяти самі коментарі, щоб потім було легко розібрати цей файл і вивести необхідну інформацію на сторінку.
- Примітка2: Можете використати функції **fgets()** та **fseek()**

2) Дано два файли зі словами, розділеними пробілами.

- Створити 3 нових файли, які будуть містити:
 - а) рядки, які зустрічаються тільки в першому файлі;
 - б) рядки, які зустрічаються в обох файлах;
 - в) рядки, які зустрічаються в кожному файлі більше двох разів.
- Створити форму, в яку користувач вводить ім'я файлу з тих що були створені вище, і цей файл має бути видалений

3) Дан файл зі словами. Впорядкувати слова за алфавітом.

1. comments/index.php (Система коментарів)

Основні виконані частини:

- **Збереження коментарів у файл (comments.txt):**
 - При отриманні POST-запиту з заповненими полями "ім'я" та "коментар", скрипт обробляє введені дані (trim(), htmlspecialchars() для безпеки).
 - Формується рядок у форматі ім'я|коментар\n.

					ІРТР.420001.123-ЗЛ	Арк.
						2
Змн.	Арк.	№ докум.	Підпис	Дата		

- За допомогою `file_put_contents($file_name, $line, FILE_APPEND)` цей рядок додається в кінець файлу `comments.txt`.
- **Головний момент:** Використання файлу як простої бази даних для зберігання коментарів. Застосування `FILE_APPEND` для додавання нових записів без перезапису існуючих.
- **Зчитування та відображення коментарів:**
 - Функція `readComments($filename)` зчитує рядки з файлу `comments.txt`.
 - Кожен рядок розділяється на ім'я та коментар за допомогою `explode('|', $line)`.
 - Створені масиви з ім'ям та коментарем додаються до масиву `$comments`.
 - У HTML-частині коментарі виводяться у вигляді таблиці. Якщо коментарів немає, відображається відповідне повідомлення.
 - **Головний момент:** Зчитування даних з файлу, їхня обробка та відображення у структурованому вигляді на веб-сторінці.
- **Форма для додавання коментарів:**
 - На сторінці розміщена HTML-форма з полями для введення імені та тексту коментаря, яка відправляє дані методом POST на цю ж сторінку.
 - **Головний момент:** Надання користувачеві інтерфейсу для взаємодії із системою коментарів.

2. files/index.php (Робота з файлами рядків)

Основні виконані частини:

- **Зчитування рядків з файлів та їх підрахунок:**
 - Функція `readLinesFromFile($filename)` відкриває вказаний файл, построчно зчитує рядки, видаляє зайві пробіли на початку та в кінці (`trim()`) та підраховує кількість кожного унікального рядка у масиві `$lines`.

					ІПТР.420001.123-ЗЛ	Арк.
						2
Змн.	Арк.	№ докум.	Підпис	Дата		

- **Головний момент:** Ефективне зчитування та обробка вмісту текстових файлів построково з підрахунком повторень.
- **Визначення унікальних рядків першого файлу:**
 - `array_diff_key($lines1, $lines2)` порівнює ключі (унікальні рядки) масивів `$lines1` та `$lines2` та повертає рядки, які є лише в першому масиві.
 - Результат записується у файл `only_in_first.txt` за допомогою `implode("\n", ...)`.
 - **Головний момент:** Використання `array_diff_key()` для знаходження унікальних елементів на основі ключів масиву.
- **Визначення спільних рядків:**
 - `array_intersect_key($lines1, $lines2)` повертає рядки, які є ключами в обох масивах.
 - Результат записується у файл `in_both.txt`.
 - **Головний момент:** Використання `array_intersect_key()` для знаходження спільних елементів на основі ключів масиву.
- **Визначення рядків, що зустрічаються більше двох разів в обох файлах:**
 - Скрипт ітерується по рядках першого файлу (`$lines1`) та перевіряє, чи цей рядок існує у другому файлі (`isset($lines2[$line])`) і чи кількість його входжень в обох файлах (`$count > 2 && $lines2[$line] > 2`) перевищує два.
 - Знайдені рядки записуються у файл `more_than_twice.txt`.
 - **Головний момент:** Логічне порівняння кількості входжень рядків у двох різних масивах.
- **Інтерфейс для видалення створених файлів:**
 - На сторінці розміщена форма з випадаючим списком для вибору одного з трьох створених файлів.
 - При відправці форми скрипт перевіряє, чи вибраний файл є одним із дозволених, і намагається його видалити за допомогою `unlink()`.

- Виводиться повідомлення про результат видалення.
- **Головний момент:** Надання базової функціональності для керування створеними файлами.

3. sort/index.php (Сортування слів)

Основні виконані частини:

- **Перевірка існування вхідного файлу:**
 - Скрипт починає з перевірки, чи існує файл words.txt за допомогою file_exists(\$file).
 - **Головний момент:** Забезпечення обробки можливої відсутності вхідного файлу.
- **Зчитування та розділення слів:**
 - Якщо файл існує, його вміст зчитується за допомогою file_get_contents(\$file).
 - Рядок вмісту розділяється на окремі слова за допомогою регулярного виразу preg_split('/\s+/', який розбиває текст по будь-яких пробільних символах. Флаг PREG_SPLIT_NO_EMPTY використовується для ігнорування порожніх елементів.
 - **Головний момент:** Використання регулярних виразів для гнучкого розділення тексту на слова.
- **Сортування слів:**
 - Масив слів \$words сортується за алфавітом за допомогою функції sort(). Прапори SORT_STRING та SORT_FLAG_CASE забезпечують сортування як рядків та ігнорування різниці між великими та малими літерами.
 - **Головний момент:** Застосування функції sort() з відповідними прапорами для виконання алфавітного сортування без урахування регістру.
- **Запис відсортованих слів у новий файл:**

					ІТР.420001.123-ЗЛ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		2

- Відсортований масив слів об'єднується назад у рядок за допомогою implode(' ', \$words), де слова розділяються пробілами.
- Цей рядок записується у файл sorted_words.txt за допомогою file_put_contents().
- Виводиться повідомлення про успішне або невдахе збереження.
- **Головний момент:** Запис оброблених даних у новий файл.
- **Відображення вхідного та відсортованого вмісту:**
 - Якщо вхідний файл існує, його вміст відображається на сторінці у тегу <pre>.
 - Якщо файл з відсортованими словами існує, його вміст також відображається у <pre>.
 - **Головний момент:** Надання користувачеві можливості переглянути вихідні та оброблені дані.

Система коментарів

Ім'я:

Христина

Коментар:

Доброго ранку!!!

Додати коментар

Коментарі

Ім'я	Коментар
Олександр	Сьогодні гарна погода
Юлія	Сьогодні поїду додому
Анна	Виконай це завдання

Система коментарів

Ім'я:

Коментар:

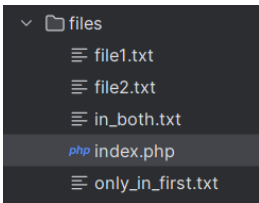
Додати коментар

Коментарі

Ім'я	Коментар
Олександр	Сьогодні гарна погода
Юлія	Сьогодні поїду додому
Анна	Виконай це завдання
Христина	Доброго ранку!!!

Робота з файлами рядків

Створені файли:



- only_in_first.txt - Рядки, які зустрічаються тільки в першому файлі
- in_both.txt - Рядки, які зустрічаються в обох файлах
- more_than_twice.txt - Рядки, які зустрічаються в кожному файлі більше двох разів

Виберіть файл для видалення:

only_in_first.txt

Видалити файл

Файл more_than_twice.txt був успішно видалений.

Сортування слів за алфавітом

Слова успішно відсортовані за алфавітом та збережені у файлі 'sorted_words.txt'.

Слова:

Ліхтар, Дзеркало, Олівець, Печиво, Вітер, Пляшка, Хмара, Ранок, Школа, Музика

Відсортовані слова:

Вітер, Дзеркало, Ліхтар, Музика Олівець, Печиво, Пляшка, Ранок, Хмара, Школа,

Рис.5-9 Результат виконання програми

Завдання 4: Передача файлів через форми і прийом файлів у PHP-скрипті (1 завдання)

Розробіть форму для завантаження зображень на сервер. Після відправлення форми, зображення повинні бути прийняті PHP-скриптом, завантажені на сервер та збережені в певному каталозі.

Код обробляє завантаження зображень користувачами на сервер. Спочатку він перевіряє, чи передано файл зображення через форму. Потім створює директорію для зберігання завантажених файлів, якщо вона відсутня. Далі код перевіряє тип та розмір завантаженого файлу, дозволяючи лише певні формати та обмежуючи розмір. У разі успішних перевірок файл завантажується на сервер. Після цього скрипт отримує список усіх завантажених зображень і відображає їх на сторінці у вигляді галереї, а також виводить повідомлення про результат завантаження.

Завантаження зображень

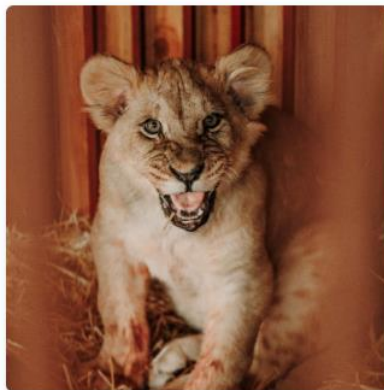
Зображення успішно завантажено!

Виберіть зображення для завантаження:

Вибір файлу f24075a3-6...f222be9.jpg

Завантажити

Галерея завантажених зображень:



					IPTP.420001.123-3Л	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		2

Завантаження зображень

Зображення успішно завантажено!

Виберіть зображення для завантаження:

Вибір файлу

Файл не вибрано

Завантажити

Галерея завантажених зображень:

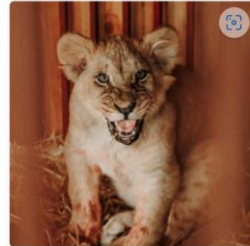


Рис.10-11 Результат виконання програми

Завдання 5: Робота з каталогами (1 завдання)

- Створіть форму з наступними полями: **Логін** та **Пароль**
- Опрацюйте форму та створіть папку з ім'ям логіна, якщо такої папки ще немає. А якщо є, то вивести повідомлення про помилку
- Всередині створеної папки створіть підпапки video, music, photo, а також декілька файлів всередині цих папок.
- Створіть окрему сторінку **delete.php** з полями **Логін** та **Пароль**
- Якщо ім'я логіна співпадає з ім'ям створеної папки, що була створена з попередньої сторінки, то вилучити цю папку з усім вмістом

index.php (Створення користувача):

Цей файл відповідає за створення нового користувача. Коли користувач заповнює форму з логіном та паролем і відправляє її, скрипт перевіряє, чи такий логін вже існує. Якщо ні, створюється нова директорія для цього користувача, в якій розміщуються підпапки для відео, музики та фото, а також створюються демонстраційні файли. Інформація про логін та хешований пароль зберігається у файлі user_info.json у директорії користувача. Користувачеві виводиться повідомлення про успішне створення або помилку.

delete.php (Видалення користувача):

					ІТР.420001.123-ЗЛ	Арк.
						2
Змн.	Арк.	№ докум.	Підпис	Дата		

Цей файл відповідає за видалення існуючого користувача. Коли користувач вводить свій логін та пароль у форму і відправляє її, скрипт перевіряє, чи існує такий логін та чи введений пароль є вірним. У разі успішної перевірки, скрипт рекурсивно видаляє директорію користувача разом з усім її вмістом. Користувачеві виводиться повідомлення про успішне видалення або помилку.

Створення користувача

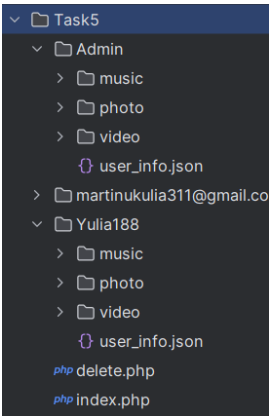
Користувача 'Yulia188' успішно створено з усіма необхідними папками та файлами!

Логін:

Пароль:

Створити користувача

Видалити користувача



Видалення користувача

Логін:

Yulia188

Пароль:

.....

Видалити користувача

Видалення користувача

Користувача 'Yulia188' та всі його дані успішно видалено!

Логін:

Пароль:

Видалити користувача

Назад до створення користувача

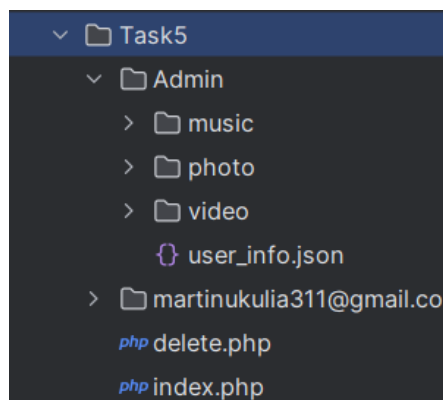


Рис.12-16 Результат виконання програми

Висновок: На сьогоднішньому занятті я навчилася працювати з файлами: створювати, читати, записувати і видаляти дані з файлів, навчитися працювати з каталогами, cookie, session, передавати файли через форми і приймати файли у PHP-скрипті

					IPTP.420001.123-3Л	Арк.
						2
Змн.	Арк.	№ докум.	Підпис	Дата		