

Лабораторна робота №7

Тема: Встановлення HTTP-статус кодів та функції керування буферизацією виводу.

Мета роботи: ознайомитися з HTTP-статусами кодів та функціями керування буферизацією виводу.

Хід роботи:

Завдання 1: Динамічне кешування сторінки з урахуванням статус-коду

Створіть PHP-скрипт `cache_page.php`, який реалізує кешування сторінки залежно від статус-коду відповіді.

Умови:

1. Якщо сторінка віддає статус 200 OK, її вміст зберігається у файлі кешу `cache.html`.
2. Якщо статус 404 Not Found, кешування не відбувається, а файл `cache.html` (якщо він існує) видаляється.
3. Якщо користувач відвідує сторінку повторно, і є кеш-файл, вміст віддається з нього без повторної генерації сторінки.

Лістинг:

`Cache_page.php`

```
<?php
$cache_file = 'cache.html';
if (file_exists($cache_file)) {
    unlink($cache_file);
}

function generatePageContent($title, $text)
{
    $content = '<h1>'. $title . '</h1>';
    $content .= '<p>'. $text . '</p>';
    return $content;
}

// http_response_code(404);
```

					Житомирська політехніка.24.121.08.000 – Лр7			
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Незнайко В.К.			Звіт з лабораторної роботи		Літ.	Арк.
Перевір.		Ковтун В.В.						1
Керівник							ФІКТ Гр. ВТк-24-1	
Н. контр.								
Зав. каф.								
							12	

```

if (file_exists($cache_file) && http_response_code() == 200) {
    echo file_get_contents($cache_file);
} else {
    ob_start();

    $content = generatePageContent("Some page", "Text");
    echo $content;

    $content = ob_get_contents();

    ob_end_clean();

    if (http_response_code() == 200) {
        file_put_contents($cache_file, $content);
        echo $content;
    } elseif (http_response_code() == 404) {
        if (file_exists($cache_file)) {
            unlink($cache_file);
        }
        echo '<h1>Сторінка не знайдена</h1>';
    }
}

```

.htaccess

ErrorDocument 404 /task1/cache_page.php

Результат виконання:

Some page

Text

Завдання 2. Система захисту від перевантаження сервера (Rate Limiting):

Створіть скрипт rate_limit.php, який обмежує кількість запитів від одного користувача та повертає відповідні статус-коди.

Умови:

- Логувати IP-адресу користувача в файл requests.log.
- Якщо користувач зробив більше 5 запитів за хвилину, встановлювати статус 429 Too Many Requests і рекомендувати спробувати пізніше.
- Якщо ліміт не перевищено, видавати 200 ОК та звичайний контент.

		Незнайко В.К.			Житомирська політехніка.24.121.08.000 – Лр7	Арк.
		Ковтун В.В				
Змн.	Арк.	№ докум.	Підпис	Дата		2

ЛІСТИНГ:

Rate_limit.php

```
<?php
$log_file = 'requests.log';
$limit = 5;
$time_window = 10;

$ip = $_SERVER['REMOTE_ADDR'];
$current_time = time();

$logs = file_exists($log_file) ? file($log_file, FILE_IGNORE_NEW_LINES |
FILE_SKIP_EMPTY_LINES) : [];

$filtered_logs = [];
$request_count = 0;

foreach ($logs as $log) {
    list($log_ip, $log_time) = explode('|', $log);
    if ($current_time - $log_time < $time_window) {
        $filtered_logs[] = $log;
        if ($log_ip === $ip) {
            $request_count++;
        }
    }
}

if ($request_count >= $limit) {
    http_response_code(429);
    header('Retry-After: 10');
    die('<h1>429 Too Many Requests</h1><p>Будь ласка, спробуйте пізніше.</p>');
}

$filtered_logs[] = "$ip|$current_time";
file_put_contents($log_file, implode("\n", $filtered_logs) . "\n");

http_response_code(200);
echo "<h1>Сторінка доступна</h1><p>Ваш запит успішно оброблено.</p>";
```

Результат виконання:

Сторінка доступна

Ваш запит успішно оброблено.

		Незнайко В.К.			Житомирська політехніка.24.121.08.000 – Лр7	Арк.
		Ковтун В.В				
Змн.	Арк.	№ докум.	Підпис	Дата		3

429 Too Many Requests

Будь ласка, спробуйте пізніше.

Завдання 3. Перехоплення помилок та перенаправлення:

Створіть скрипт `error_handler.php`, який перехоплює фатальні помилки та замість стандартного повідомлення показує кастомну сторінку з відповідним статус-кодом.

Умови:

- Використовуйте `register_shutdown_function()`, щоб обробляти `E_ERROR`.
- Якщо сталася помилка:
 - Очистити буфер `ob_clean()`
 - Встановити статус 500 Internal Server Error
 - Показати сторінку з вибаченням та часом, коли проблема буде вирішена.
- Якщо помилок немає, віддавати 200 OK.

Лістинг:

`error_handler.php`

```
<?php
ob_start();

register_shutdown_function('shutdownHandler');
function shutdownHandler()
{
    $error = error_get_last();
    if ($error !== null && $error['type'] === E_ERROR) {
        ob_clean();
        http_response_code(500);

        $fix_time = date("H:i", strtotime("+30 minutes"));

        echo "<h1>500 Internal Server Error</h1>";
        echo "<p>Вибачте, сталася помилка сервера. Очікуваний час відновлення:
{$fix_time}</p>";
    }
}
```

		Незнайко В.К.			Житомирська політехніка.24.121.08.000 – Лр7	Арк.
		Ковтун В.В				
Змн.	Арк.	№ докум.	Підпис	Дата		4

```
$a = 1/0;
```

```
echo "<h1>Сторінка працює</h1><p>Все нормально!</p>";
```

Результат виконання:

500 Internal Server Error

Вибачте, сталася помилка сервера. Очікуваний час відновлення: 17:33

Завдання 4. Гнучка система редиректів:

Створіть скрипт `redirect_manager.php`, який дозволяє керувати перенаправленнями через конфігураційний файл.

Умови:

- Є файл `redirects.json` зі списком перенаправлень:

```
{  
  "/old-page": "/new-page",  
  "/deprecated": "/404"  
}
```
- Якщо користувач заходить на `/old-page`, його потрібно перенаправити на `/new-page` зі статусом 301 Moved Permanently.
- Якщо сторінка `/deprecated`, видавати 404 Not Found та відповідне повідомлення.
- Використовувати `ob_start()`, щоб контролювати вивід.

Лістинг:

`Redirect_manager.php`

```
<?php  
ob_start();  
  
$redirects = json_decode(file_get_contents("redirects.json"), true);  
$request_url = str_replace("/task4", "", $_SERVER['REQUEST_URI']);  
  
if (isset($redirects[$request_url])) {  
    $new_url = $redirects[$request_url];
```

		Незнайко В.К.			Житомирська політехніка.24.121.08.000 – Лр7	Арк.
		Ковтун В.В				5
Змн.	Арк.	№ докум.	Підпис	Дата		

```

if ($new_url == "/404") {
    header("Location: /task4/404.php", true, 301);
} else {
    header("Location: /task4" . $new_url . ".php", true, 301);
}
exit;
}

```

```

http_response_code(200);
include("index.php");

```

new-page.php

```

<!DOCTYPE html>
<html lang="uk">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>New Page</title>
</head>
<body>
    <h1>Це нова сторінка!</h1>
    <p>Вітаємо на новій сторінці!</p>
</body>
</html>

```

404.php

```

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

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>404</title>
</head>

<body>
    <h1>404 Not Found</h1>
    <p>Ця сторінка не існує.</p>
</body>

</html>

```

index.php

```

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

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>

```

		Незнайко В.К.			Житомирська політехніка.24.121.08.000 – Лр7	Арк.
		Ковтун В.В				
Змн.	Арк.	№ докум.	Підпис	Дата		6

```

</head>

<body>
  <p><a href="/task4/old-page">Перейти на стару сторінку (old-page)</a></p>
  <p><a href="/task4/deprecated">Перейти на застарілу сторінку (deprecated)</a></p>
</body>

</html>

```

```

.htaccess

RewriteEngine On

RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
RewriteRule ^(.*)$ redirect_manager.php [L,QSA]
    Redirects.json
{
  "/old-page": "/new-page",
  "/deprecated": "/404"
}

```

Результат виконання:

[Перейти на стару сторінку \(old-page\)](#)

[Перейти на застарілу сторінку \(deprecated\)](#)

Це нова сторінка!

Вітаємо на новій сторінці!

404 Not Found

Ця сторінка не існує.

Завдання 5. Динамічне підключення заголовків без видимого виводу:

Створіть PHP-клас Response, який допомагає керувати HTTP-заголовками та буферизацією.

		Незнайко В.К.			Житомирська політехніка.24.121.08.000 – Лр7	Арк.
		Ковтун В.В				
Змн.	Арк.	№ докум.	Підпис	Дата		7

Функціонал класу:

- Метод setStatus(\$code), який встановлює HTTP-статус.
- Метод addHeader(\$header), який додає заголовок (наприклад, Content-Type).
- Метод send(\$content), який очищає буфер, додає заголовки та відправляє відповідь.

Лістинг:

Response.php

```
<?php
class Response
{
    private int $statusCode;
    private array $headers = [];

    public function setStatus(int $code): void
    {
        $this->statusCode = $code;
        header("HTTP/1.1 $code");
    }

    public function addHeader(string $header): void
    {
        $this->headers[] = $header;
    }

    public function send(string $content): void
    {
        ob_start();
        ob_clean();

        foreach ($this->headers as $header) {
            header($header);
        }

        echo $content;
        ob_end_flush();
    }
}
```

Index.php

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

		Незнайко В.К.			Житомирська політехніка.24.121.08.000 – Лр7	Арк.
		Ковтун В.В				
Змн.	Арк.	№ докум.	Підпис	Дата		8


```

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>

<body>
  <?php
    require_once 'Response.php';
    $response = new Response();
    $response->setStatus(200);
    $response->addHeader("Content-Type: text/html");
    $response->send("<h1>Вітаємо!</h1><p>Це динамічна відповідь.</p>");

    $response = new Response();
    $response->setStatus(404);
    $response->addHeader("Content-Type: text/html");
    $response->send("<h1>404</h1><p>Сторінку не знайдено.</p>");
  ?>
</body>
</html>

```

Результат виконання:

Вітаємо!

Це динамічна відповідь.

404

Сторінку не знайдено.

Завдання 6. Аналіз трафіку та відповідей сервера:

Розробіть traffic_logger.php, який записує інформацію про всі запити в базу даних.

Що зберігати в БД?

- IP-адресу
- Час запиту
- Запитаний URL
- Відповідний HTTP-статус

		Незнайко В.К.			Житомирська політехніка.24.121.08.000 – Лр7	Арк.
		Ковтун В.В				
Змн.	Арк.	№ докум.	Підпис	Дата		9

Додаткові функції:

1. Реалізувати перегляд статистики через stats.php, який рахує кількість 404 помилок за останню добу.
2. Якщо відсоток 404 помилок перевищує 10% від загальної кількості запитів, надсилати адміністратору повідомлення.

Лістинг:

Index.php

```
<?php include 'traffic_logger.php'; ?>
<!DOCTYPE html>
<html lang="uk">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Головна сторінка</title>
</head>

<body>
    <h1>Ласкаво просимо!</h1>
    <p>Це головна сторінка нашого сайту.</p>

    <div>
        <a href="about.php">Про нас</a>
        <a href="stats.php">Статистика</a>
    </div>

</body>

</html>
```

About.php

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

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>

<body>
    <h1>Про нас</h1>
    <a href='index.php'>На головну</a>
</body>

</html>
```

		Незнайко В.К.			Житомирська політехніка.24.121.08.000 – Лр7	Арк.
		Ковтун В.В				10
Змн.	Арк.	№ докум.	Підпис	Дата		

404.php

```
<?php
http_response_code(404);
include 'traffic_logger.php';
?>
<!DOCTYPE html>
<html lang="uk">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Сторінка не знайдена</title>
</head>

<body>

    <div class="container">
        <h1>404 - Сторінка не знайдена</h1>
        <p>Вибачте, але запитана сторінка не існує.</p>
        <a href="index.php">На головну</a>
    </div>

</body>

</html>
```

Stats.php

```
<?php

$db = new PDO('mysql:host=localhost;dbname=lab7;charset=utf8', 'root', '');
$db->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

$totalStmt = $db->prepare("SELECT COUNT(*) FROM logs WHERE request_time >= NOW() -
INTERVAL 1 DAY");
$totalStmt->execute();
$totalRequests = $totalStmt->fetchColumn();

$errorStmt = $db->prepare("SELECT COUNT(*) FROM logs WHERE status = 404 AND
request_time >= NOW() - INTERVAL 1 DAY");
$errorStmt->execute();
$errorRequests = $errorStmt->fetchColumn();

$percent404 = ($totalRequests > 0) ? ($errorRequests / $totalRequests) * 100 : 0;

echo "<h2>Статистика 404 помилок за останню добу</h2>";
echo "<p>Загальна кількість запитів: $totalRequests</p>";
echo "<p>Кількість 404 помилок: $errorRequests</p>";
echo "<p>Відсоток 404 помилок: " . number_format($percent404, 2) . "%</p>";
echo "<a href='index.php'>На головну</a>";
if ($percent404 > 10) {
```

		Незнайко В.К.			Житомирська політехніка.24.121.08.000 – Лр7	Арк.
		Ковтун В.В				11
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        echo "<h3>Попередження: Відсоток 404 помилок перевищив 10%! Перевірте сервер.</p>";
    }
}

```

Traffic_logger.php

```

<?php
$db = new PDO('mysql:host=localhost;dbname=lab7;charset=utf8', 'root', '');
$db->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

$ip = $_SERVER['REMOTE_ADDR'];
$url = $_SERVER['REQUEST_URI'];
$time = date('Y-m-d H:i:s');
$status = http_response_code();

$stmt = $db->prepare("INSERT INTO logs (ip, request_time, url, status) VALUES (?, ?, ?, ?)");
$stmt->execute([$ip, $time, $url, $status]);
.htaccess

```

ErrorDocument 404 /task6/404.php

Результат виконання:

Статистика 404 помилок за останню добу

Загальна кількість запитів: 63

Кількість 404 помилок: 6

Відсоток 404 помилок: 9.52%

[На головну](#)

Статистика 404 помилок за останню добу

Загальна кількість запитів: 72

Кількість 404 помилок: 11

Відсоток 404 помилок: 15.28%

[На головну](#)

Попередження: Відсоток 404 помилок перевищив 10%! Перевірте сервер.

Висновок: Під час виконання лабораторної роботи ознайомилися з HTTP-статусами кодів та функціями керування буферизацією виводу.

		Незнайко В.К.			Житомирська політехніка.24.121.08.000 – Лр7	Арк.
		Ковтун В.В				
Змн.	Арк.	№ докум.	Підпис	Дата		12