

Лабораторна робота №7

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

Хід роботи:

Завдання 1. Динамічне кешування сторінки з урахуванням статус-коду:

```
<?php
```

```
// Шлях до файлу кешу  
$cacheFile = 'cache.html';
```

```
// Перевіряємо, чи існує кеш і чи не запитана сторінка з помилкою  
if (file_exists($cacheFile) && !isset($_GET['error'])) {  
    // Віддаємо кешований вміст  
    echo file_get_contents($cacheFile);  
    exit; // Завершуємо виконання скрипта  
}
```

```
// Функція для генерації контенту сторінки  
function generatePageContent() {  
    // Визначаємо статус-код залежно від параметра error  
    $status = (isset($_GET['error']) && $_GET['error'] == '404') ? 404 : 200;  
    // Встановлюємо HTTP статус-код  
    http_response_code($status);  
    // Генеруємо вміст залежно від статус-коду  
    if ($status === 200) {  
        return "<h1>Вітаємо!</h1><p>Це успішно згенерована сторінка</p><p>Час генерації: " .  
            date('H:i:s') . "</p>";  
    } else {  
        return "<h1>404 Not Found</h1><p>Сторінку не знайдено</p>";  
    }  
}
```

```
// Починаємо буферизацію виводу  
ob_start();
```

```
// Генеруємо вміст сторінки  
$content = generatePageContent();  
echo $content; // Виводимо контент
```

ДУ «Житомирська політехніка».23.121.07.000 - Лр1			
Зм.	Зм.	Лім.	Арк.
Розроб.	Лютянський Дю		1
Перевір.	Дмитренко І.А		ZZ
Керівник		Звіт з лабораторної роботи	
Н. контр.			
Зав. каф.			
		ФІКТ Гр. ІПЗ-23-4[2]	

```

$statusCode = http_response_code();

// Обробляємо кешування залежно від статус-коду
if ($statusCode === 200) {
    // Отримуємо вміст буфера
    $bufferedContent = ob_get_contents();
    // Завершуємо буферизацію, виводимо вміст
    ob_end_flush();
    // Зберігаємо у файл кешу
    file_put_contents($cacheFile, $bufferedContent);
} else if ($statusCode === 404) {
    // Завершуємо буферизацію, виводимо вміст
    ob_end_flush();
    // Видаляємо файл кешу, якщо він існує
    if (file_exists($cacheFile)) {
        unlink($cacheFile);
    }
} else {
    // Для інших статус-кодів просто виводимо вміст
    ob_end_flush();
}
?>

```

Вітаємо!

Це успішно згенерована сторінка

Час генерації: 14:34:32

завдання 2

<?php

```

ob_start();

// Конфігурація
$logFile = 'requests.log';
$rateLimit = 5; // Максимальна кількість запитів
$timeWindow = 60; // Вікно часу у секундах

// Отримуємо IP користувача
$ip = $_SERVER['REMOTE_ADDR'];
$timestamp = time();

```

| | | | | | | |
|------|------|---------------|--------|------|--|------|
| | | Лютянський Дю | | | ДУ «Житомирська політехніка».23.121.07.000 – Лр6 | Арк. |
| | | Дмитренко І.А | | | | 3 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

```
// Завантажуємо існуючі записи з лог-файлу
$requests = [];
if (file_exists($logFile)) {
$logEntries = file($logFile, FILE_IGNORE_NEW_LINES | FILE_SKIP_EMPTY_LINES);
foreach ($logEntries as $entry) {
list($loggedIp, $loggedTime) = explode(' ', $entry);
if ($timestamp - $loggedTime <= $timeWindow) {
$requests[] = [$loggedIp, (int)$loggedTime];
}
}
}
```

```
// Фільтруємо записи для поточного IP
$recentRequests = array_filter($requests, function ($req) use ($ip) {
return $req[0] === $ip;
});
```

```
// Перевірка обмеження
if (count($recentRequests) >= $rateLimit) {
http_response_code(429);
header('Retry-After: ' . $timeWindow);
echo "Too Many Requests. Please try again later.";
} else {
// Додаємо новий запис
$requests[] = [$ip, $timestamp];
http_response_code(200);
echo "Request successful!";
}
if (!is_writable(dirname($logFile))) {
die("Помилка: Немає прав на запис у директорію логів!");
}
```

```
// Оновлюємо лог-файл
$logData = array_map(fn($req) => implode(' ', $req), $requests);
if (file_put_contents($logFile, implode("\n", $logData) . "\n") === false) {
die("Помилка: не вдалося записати у файл логів!");
}
```

```
ob_end_flush();
?>
```

Too Many Requests. Please try again later.

завдання 3

| | | | | | | |
|------|------|---------------|--------|------|--|------|
| | | Лютянський Дю | | | ДУ «Житомирська політехніка».23.121.07.000 – Лр6 | Арк. |
| | | Дмитренко І.А | | | | 3 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

Вітаємо на нашому сайті

Ця сторінка відображається нормально, оскільки немає помилок.

500 Internal Server Error

Перепрошуємо, на сервері сталася помилка.

Наша технічна команда вже працює над вирішенням проблеми.

Очікуваний час вирішення: 16:37

Дякуємо за розуміння.

<?php

```
// Початок буферизації виводу
ob_start();
```

```
// Реєстрація функції, яка виконається при завершенні скрипта
register_shutdown_function('handleFatalError');
```

```
// Основна функція обробки фатальних помилок
function handleFatalError() {
    // Отримуємо останню помилку
    $error = error_get_last();
    // Перевіряємо, чи є помилка і чи вона фатальна (E_ERROR)
    if ($error !== null && $error['type'] === E_ERROR) {
        // Очищуємо буфер виводу
        ob_clean();
        // Встановлюємо HTTP статус 500
        http_response_code(500);
        // Визначаємо орієнтовний час вирішення (наприклад, через 2 години)
        $resolutionTime = date('H:i', strtotime('+2 hours'));
        // Виводимо кастомну сторінку помилки
        echo '<!DOCTYPE html>
<html lang="uk">
<head>
<meta charset="UTF-8">
```

		Лютянський Дю			ДУ «Житомирська політехніка».23.121.07.000 – Лр6	Арк.
		Дмитренко І.А				3
Змн.	Арк.	№ докум.	Підпис	Дата		

```

<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>500 Internal Server Error</title>
<style>
body {
font-family: Arial, sans-serif;
background-color: #f8f8f8;
color: #333;
margin: 0;
padding: 20px;
text-align: center;
}
.container {
max-width: 600px;
margin: 50px auto;
padding: 30px;
background-color: #fff;
border-radius: 5px;
box-shadow: 0 2px 10px rgba(0, 0, 0, 0.1);
}
h1 {
color: #e74c3c;
}
.resolution-time {
font-weight: bold;
color: #3498db;
}
</style>
</head>
<body>
<div class="container">
<h1>500 Internal Server Error</h1>
<p>Перепрошуємо, на сервері сталася помилка.</p>
<p>Наша технічна команда вже працює над вирішенням проблеми.</p>
<p>Очікуваний час вирішення: <span class="resolution-time">' . $resolutionTime . '</span></p>
<p>Дякуємо за розуміння.</p>
</div>
</body>
</html>';
} else {
// Якщо немає фатальних помилок, встановлюємо статус 200 ОК
http_response_code(200);
// Завершуємо буфер, відправляємо контент
ob_end_flush();
}

// Тут може бути основний код сторінки
echo '<h1>Вітаємо на нашому сайті</h1>';
echo '<p>Ця сторінка відображається нормально, оскільки немає помилок.</p>';

```

		Лютянський Дю			ДУ «Житомирська політехніка».23.121.07.000 – Лр6	Арк.
		Дмитренко І.А				3
Змн.	Арк.	№ докум.	Підпис	Дата		

```
// Для тестування фатальної помилки, розкоментуйте наступний рядок:  
nonexistentFunction(); // Виклик неіснуючої функції спричинить E_ERROR
```

```
?>
```

завдання 4

Ласкаво просимо!

Це звичайна сторінка без перенаправлення.

Поточний URI: /project/Lab7/Task4/redirect_manager.php

<?php

```
// Розпочинаємо буферизацію виводу  
ob_start();
```

```
// Функція отримання поточного URI  
function getCurrentUri() {  
    // Отримуємо URI запиту  
    $uri = $_SERVER['REQUEST_URI'];  
    // Видаляємо параметри GET запиту, якщо вони є  
    $uri = strtok($uri, '?');  
    // Видаляємо завершальний слеш, якщо він є  
    $uri = rtrim($uri, '/');  
    return $uri;  
}
```

```
// Функція обробки перенаправлень  
function handleRedirects() {  
    // Шлях до файлу з перенаправленнями  
    $redirectsFile = 'redirects.json';  
    // Перевіряємо, чи існує файл  
    if (!file_exists($redirectsFile)) {  
        // Якщо файлу немає, виводимо повідомлення про помилку  
        http_response_code(500);  
        echo "<h1>500 Internal Server Error</h1>";  
        echo "<p>Файл конфігурації перенаправлень відсутній.</p>";  
        return false;  
    }  
    try {  
        // Зчитуємо та декодуємо JSON файл з перенаправленнями  
        $redirects = json_decode(file_get_contents($redirectsFile), true);  
        // Перевіряємо, чи вдалося декодувати JSON
```

		Лютянський Дю			ДУ «Житомирська політехніка».23.121.07.000 – Лр6	Арк.
		Дмитренко І.А				3
Змн.	Арк.	№ докум.	Підпис	Дата		

```

if ($redirects === null && json_last_error() !== JSON_ERROR_NONE) {
throw new Exception("Помилка декодування JSON: " . json_last_error_msg());
}
// Отримуємо поточний URI
$currentUri = getCurrentUri();
// Перевіряємо, чи є поточний URI у списку перенаправлень
if (isset($redirects[$currentUri])) {
$target = $redirects[$currentUri];
// Перевіряємо тип перенаправлення
if ($target === '/404') {
// Для спеціального випадку '/404' віддаємо 404 статус
http_response_code(404);
echo "<h1>404 Not Found</h1>";
echo "<p>Запитана сторінка не знайдена або більше не доступна.</p>";
} else {
// Для звичайного перенаправлення
// Очищуємо буфер перед відправкою заголовків
ob_clean();
// Відправляємо заголовок перенаправлення 301
header("Location: {$target}", true, 301);
exit; // Завершуємо виконання скрипта
}
return true;
}
// Якщо перенаправлення не знайдено, відображаємо нормальний контент
return false;
} catch (Exception $e) {
// Обробляємо помилку, якщо щось пішло не так
http_response_code(500);
echo "<h1>500 Internal Server Error</h1>";
echo "<p>Виникла помилка при обробці перенаправлень: " . $e->getMessage() . "</p>";
return false;
}
}

// Обробляємо перенаправлення
$redirected = handleRedirects();

// Якщо перенаправлення не відбулося, виводимо нормальний контент
if (!$redirected) {
// Встановлюємо статус 200 OK
http_response_code(200);
echo "<h1>Ласкаво просимо!</h1>";
echo "<p>Це звичайна сторінка без перенаправлення.</p>";
echo "<p>Поточний URI: " . getCurrentUri() . "</p>";
}

// Завершуємо буферизацію та відправляємо вміст
ob_end_flush();
?>

```

		Лютянський Дю			ДУ «Житомирська політехніка».23.121.07.000 – Лр6	Арк.
		Дмитренко І.А				3
Змн.	Арк.	№ докум.	Підпис	Дата		

завдання 5

Вітаємо!

Це динамічна відповідь від сервера.

Поточний час: 14:40:32 03.04.2025

<?php

```
// Підключаємо файл з класом Response
require_once 'Response.php';

// Створюємо новий екземпляр класу Response
$response = new Response();

// Налаштовуємо відповідь
$response->setStatus(200)
->addHeader("Content-Type: text/html; charset=UTF-8")
->addHeader("X-Powered-By: PHP Response Class");

// Готуємо контент
$content = "<!DOCTYPE html>
<html>
<head>
<title>Відповідь від сервера</title>
<style>
body {
font-family: Arial, sans-serif;
max-width: 800px;
margin: 0 auto;
padding: 20px;
}
h1 { color: #336699; }
</style>
</head>
<body>
<h1>Вітаємо!</h1>
<p>Це динамічна відповідь від сервера.</p>
<p>Поточний час: " . date('H:i:s d.m.Y') . "</p>
</body>
</html>";
```

		Лютянський Дю			ДУ «Житомирська політехніка».23.121.07.000 – Лр6	Арк.
		Дмитренко І.А				3
Змн.	Арк.	№ докум.	Підпис	Дата		


```

// Відправляємо відповідь
$response->send($content);
<?php
/**
 * Клас Response для керування HTTP-заголовками та буферизацією
 */
class Response {
/**
 * Зберігає HTTP-статус код
 * @var int
 */
private $statusCode = 200;
/**
 * Зберігає масив HTTP-заголовків
 * @var array
 */
private $headers = [];
/**
 * Встановлює HTTP-статус код
 *
 * @param int $code HTTP-статус код
 * @return Response Поточний об'єкт (для ланцюжкових викликів)
 */
public function setStatus($code) {
    $this->statusCode = (int)$code;
    return $this;
}
/**
 * Додає HTTP-заголовок
 *
 * @param string $header HTTP-заголовок
 * @return Response Поточний об'єкт (для ланцюжкових викликів)
 */
public function addHeader($header) {
    $this->headers[] = $header;
    return $this;
}
/**
 * Відправляє відповідь з заголовками та контентом
 *
 * @param string $content Контент для відправки
 */
public function send($content) {
    // Запускаємо буферизацію, якщо вона ще не запущена
    if (ob_get_level() == 0) {
        ob_start();
    }
    // Очищаємо буфер виводу
    ob_clean();

```

		Лютянський Дю			ДУ «Житомирська політехніка».23.121.07.000 – Лр6	Арк.
		Дмитренко І.А				3
Змн.	Арк.	№ докум.	Підпис	Дата		

```
// Встановлюємо HTTP-статус код
http_response_code($this->statusCode);
// Додаємо всі заголовки
foreach ($this->headers as $header) {
    header($header);
}
// Виводимо контент
echo $content;
// Відправляємо буфер клієнту
ob_end_flush();
}
```

Завдання 6

Traffic Statistics (Last 24 Hours)

Overview

Total Requests: 7

404 Errors: 0

404 Error Percentage: 0%

Top 404 URLs

No 404 errors in the last 24 hours.

```
<?php
```

```
// Database configuration (same as in traffic_logger.php)
define('DB_HOST', 'localhost');
define('DB_USER', 'root');
define('DB_PASS', '36044170Sql');
define('DB_NAME', 'traffic_logs');

// Function to get 404 statistics
function get404Stats() {
    try {
        $pdo = new PDO("mysql:host=".DB_HOST.";dbname=".DB_NAME, DB_USER, DB_PASS);
        $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
        $oneDayAgo = date('Y-m-d H:i:s', strtotime('-1 day'));
        // Get total requests
        $stmt = $pdo->prepare("SELECT COUNT(*) as total FROM traffic_logs WHERE request_time
        >= :one_day_ago");
        $stmt->execute([':one_day_ago' => $oneDayAgo]);
        $totalRequests = $stmt->fetchColumn();
        // Get 404 errors
        $stmt = $pdo->prepare("SELECT COUNT(*) as errors FROM traffic_logs
        WHERE request_time >= :one_day_ago AND http_status = 404");
        $stmt->execute([':one_day_ago' => $oneDayAgo]);
        $errorCount = $stmt->fetchColumn();
        // Get top 404 URLs
```

```

$stmt = $pdo->prepare("SELECT requested_url, COUNT(*) as count
FROM traffic_logs
WHERE request_time >= :one_day_ago AND http_status = 404
GROUP BY requested_url
ORDER BY count DESC
LIMIT 10");
$stmt->execute([':one_day_ago' => $oneDayAgo]);
$top404Urls = $stmt->fetchAll(PDO::FETCH_ASSOC);
return [
'total_requests' => $totalRequests,
'error_count' => $errorCount,
'error_percentage' => $totalRequests > 0 ? ($errorCount / $totalRequests) * 100 : 0,
'top_404_urls' => $top404Urls
];
} catch (PDOException $e) {
die("Database error: " . $e->getMessage());
}
}

```

```

// Get statistics
$stats = get404Stats();
?>

```

```

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Traffic Statistics</title>
<style>
body { font-family: Arial, sans-serif; margin: 20px; }
.stats { background: #f5f5f5; padding: 20px; border-radius: 5px; margin-bottom: 20px; }
table { width: 100%; border-collapse: collapse; }
th, td { padding: 8px; text-align: left; border-bottom: 1px solid #ddd; }
th { background-color: #f2f2f2; }
.alert { color: #d8000c; background-color: #ffd2d2; padding: 10px; border-radius: 5px; }
</style>
</head>
<body>
<h1>Traffic Statistics (Last 24 Hours)</h1>
<div class="stats">
<h2>Overview</h2>
<p>Total Requests: <?php echo number_format($stats['total_requests']); ?></p>
<p>404 Errors: <?php echo number_format($stats['error_count']); ?></p>
<p>404 Error Percentage: <?php echo round($stats['error_percentage'], 2); ?>%</p>
<?php if ($stats['error_percentage'] > 10): ?>
<div class="alert">
Warning: 404 error rate exceeds 10% threshold!
</div>
<?php endif; ?>

```

		Лютянський Дю			ДУ «Житомирська політехніка».23.121.07.000 – Лр6	Арк.
		Дмитренко І.А				3
Змн.	Арк.	№ докум.	Підпис	Дата		

```

</div>
<div class="stats">
<h2>Top 404 URLs</h2>
<?php if (!empty($stats['top_404_urls'])): ?>
<table>
<tr>
<th>URL</th>
<th>Count</th>
</tr>
<?php foreach ($stats['top_404_urls'] as $url): ?>
<tr>
<td><?php echo htmlspecialchars($url['requested_url']); ?></td>
<td><?php echo $url['count']; ?></td>
</tr>
<?php endforeach; ?>
</table>
<?php else: ?>
<p>No 404 errors in the last 24 hours.</p>
<?php endif; ?>
</div>
</body>
</html>
<?php
// Database configuration
define('DB_HOST', 'localhost');
define('DB_USER', 'root');
define('DB_PASS', '36044170Sql');
define('DB_NAME', 'traffic_logs');

// Administrator email for alerts
define('ADMIN_EMAIL', 'admin@example.com');

// Function to log traffic
function logTraffic() {
try {
// Get request details
$ipAddress = $_SERVER['REMOTE_ADDR'];
$requestTime = date('Y-m-d H:i:s');
$requestUrl = $_SERVER['REQUEST_URI'];
$httpStatus = http_response_code();
// Connect to database
$pdo = new PDO("mysql:host=".DB_HOST.";dbname=".DB_NAME, DB_USER, DB_PASS);
$pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
// Insert log entry
$stmt = $pdo->prepare("INSERT INTO traffic_logs (ip_address, request_time, requested_url, http_status)
VALUES (:ip_address, :request_time, :requested_url, :http_status)");
$stmt->execute([
'ip_address' => $ipAddress,
'request_time' => $requestTime,

```

		Лютянський Дю			ДУ «Житомирська політехніка».23.121.07.000 – Лр6	Арк.
		Дмитренко І.А				3
Змн.	Арк.	№ докум.	Підпис	Дата		

```

':requested_url' => $requestedUrl,
':http_status' => $httpStatus
]);
// Check if we need to send an alert (do this only occasionally to avoid spamming)
if (rand(1, 100) <= 10) { // 10% chance to check
checkForErrorSpike($pdo);
}
} catch (PDOException $e) {
// Log error to file if database connection fails
error_log("Traffic logger error: " . $e->getMessage(), 3, "traffic_logger_errors.log");
}
}

// Function to check for error spikes
function checkForErrorSpike($pdo) {
$oneDayAgo = date('Y-m-d H:i:s', strtotime('-1 day'));
// Get total requests in last 24 hours
$stmt = $pdo->prepare("SELECT COUNT(*) as total FROM traffic_logs WHERE request_time
>= :one_day_ago");
$stmt->execute([':one_day_ago' => $oneDayAgo]);
$totalRequests = $stmt->fetchColumn();
if ($totalRequests > 0) {
// Get 404 errors in last 24 hours
$stmt = $pdo->prepare("SELECT COUNT(*) as errors FROM traffic_logs
WHERE request_time >= :one_day_ago AND http_status = 404");
$stmt->execute([':one_day_ago' => $oneDayAgo]);
$errorCount = $stmt->fetchColumn();
$errorPercentage = ($errorCount / $totalRequests) * 100;
if ($errorPercentage > 10) {
// Send email alert
$subject = "High 404 Error Rate Alert";
$message = "Warning: 404 errors account for " . round($errorPercentage, 2) . "% of total requests in
the last 24 hours.\n\n";
$message .= "Total requests: $totalRequests\n";
$message .= "404 errors: $errorCount";
mail(ADMIN_EMAIL, $subject, $message);
}
}
}

// Log this request
logTraffic();
?>

```

		Лютянський Дю			ДУ «Житомирська політехніка».23.121.07.000 – Лр6	Арк.
		Дмитренко І.А				3
Змн.	Арк.	№ докум.	Підпис	Дата		