

## Лабораторна робота №7

**Тема:** Встановлення HTTP-статус кодів та функції 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.

#### Підказка:

- Використовуйте **ob\_start()** для збору вмісту сторінки.
- Використовуйте **ob\_get\_contents()** для отримання вмісту кешу.
- Використовуйте **file\_put\_contents()** для запису кешу.
- Використовуйте **file\_get\_contents()** для видачі кешу.

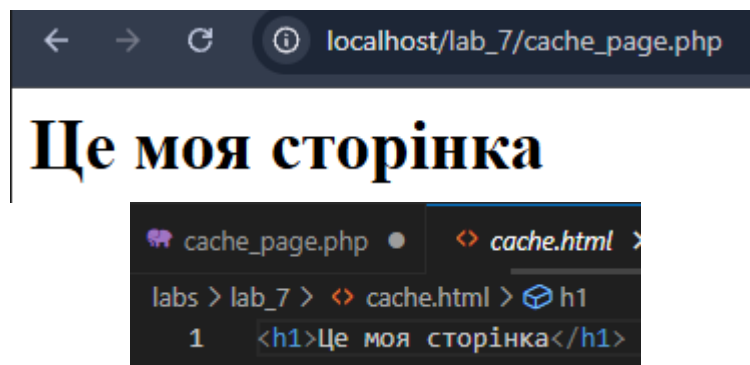


Рис. 1 та 2. Результат виконання першого завдання

Лістинг (cache\_page.php):

```
<?php
ob_start();

http_response_code(200);
echo "<h1>Це моя сторінка</h1>";
```

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.25.121.6.000 – Лр.7		
Змн.	Арк.	№ докум.	Підпис	Дата	Звіт з лабораторної роботи №7		
Розроб.		Гречко О. О.					
Перевір.		Ковтун В. В.					
Реценз.							
Н. Контр.							
Зав.каф.							
					Літ.	Арк.	Аркушів
						1	10
					ФІКТ, гр. ІПЗ-23-3		

```

$content = ob_get_contents(); //вміст сторінки з буфера

//в кеш, якщо статус 200
if (http_response_code() == 200) {
    file_put_contents('cache.html', $content);
} elseif (http_response_code() == 404) {
    // якщо статус 404, видаляємо
    if (file_exists('cache.html')) {
        unlink('cache.html');
    }
}

//вміст із кешу
if (file_exists('cache.html')) {
    echo file_get_contents('cache.html');
} else {
    echo $content;
}

ob_end_flush();
?>

```

## 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 OK** та звичайний контент.

### Підказка:

- Використовуйте **ob\_start()** для збору відповідей.
- Записуйте в лог: IP, час запиту.
- Видаляйте застарілі записи, щоб не накопичувати зайві дані.

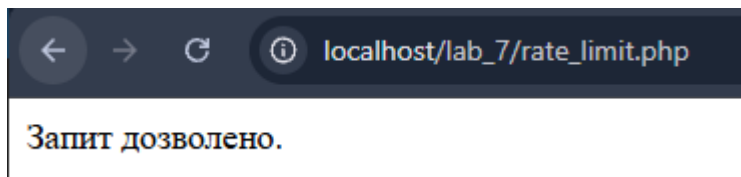


Рис. 3. Результат виконання другого завдання

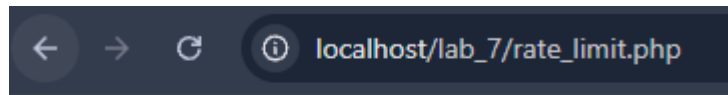
```

labs > lab_7 > ⌵ requests.log
1      :":":1", "time":1746656677}, "18": {"ip":":":1", "time":1746656677}, "19": {"ip":":":

```

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.25.121.6.000 – Лр.7	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		2

Рис. 4. Вивід в request.log



Занадто багато запитів. Спробуйте пізніше.

Рис. 5. При відправленні 5 запитів, наступні будуть скасовані

Лістинг (rate\_limit.php):

```
<?php
//IP та час запиту в лог
$ip = $_SERVER['REMOTE_ADDR'];
$timestamp = time();

$logFile = 'requests.log';

//читання логу
$requests = file_exists($logFile) ? json_decode(file_get_contents($logFile), true) :
[];

//застарілі записи
foreach ($requests as $key => $request) {
    if ($timestamp - $request['time'] > 60) {
        unset($requests[$key]);
    }
}

// Перевірка
$userRequests = array_filter($requests, function($request) use ($ip) {
    return $request['ip'] == $ip;
});

//обмежуємо доступ
if (count($userRequests) >= 5) {
    http_response_code(429);
    echo "Занадто багато запитів. Спробуйте пізніше.";
    exit;
}

//поточний запит
$requests[] = ['ip' => $ip, 'time' => $timestamp];

file_put_contents($logFile, json_encode($requests));

http_response_code(200);
echo "Запит дозволено.";
?>
```

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.25.121.6.000 – Лр.7	Арк.
						3
Змн.	Арк.	№ докум.	Підпис	Дата		

### Завдання 3. перехоплення помилок та перенаправлення:

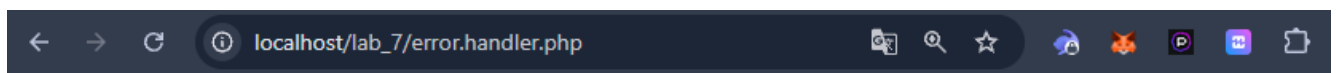
Створіть скрипт **error\_handler.php**, який перехоплює фатальні помилки та замість стандартного повідомлення показує кастомну сторінку з відповідним статус-кодом.

#### 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**.

#### Підказка:

- **error\_get\_last()** допоможе отримати останню помилку.
- Використовуйте **ob\_start()**, щоб уникнути передчасного виводу.



**Щось пішло не так. Ми працюємо над вирішенням проблеми. Спробуйте пізніше.**

Рис. 6. Результат виконання завдання

Лістинг (error.handler.php):

```
<?php
// rгf дозволяє зареєструвати функцію, яка буде виконана після завершення скрипта.
register_shutdown_function(function() {
    $error = error_get_last();
    if ($error && $error['type'] == E_ERROR) {
        ob_clean();
        http_response_code(500); //statu
        echo "<h1>Щось пішло не так. Ми працюємо над вирішенням проблеми. Спробуйте
пізніше.</h1>";
        exit;
    }
});

nonExistentFunction();//помилка
?>
```

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.25.121.6.000 – Лр.7	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		4

#### Завдання 4. Гнучка система редиректів:

Створіть скрипт **redirect\_manager.php**, який дозволяє керувати перенаправленнями через конфігураційний файл.

##### Умови:

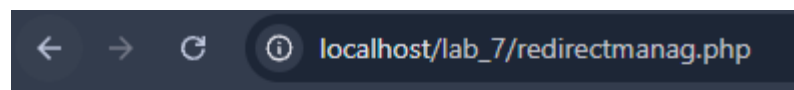
- Є файл **redirects.json** зі списком перенаправлень:

```
{
    "/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.

##### Підказка:

- json\_decode(file\_get\_contents("redirects.json"), true);** дозволить отримати дані.
- header("Location: \$new\_url", true, 301);** виконує перенаправлення.



# Ласкаво просимо

Це сторінка без перенаправлень.



Рис. 7 та 8. Результат виконання завдання

Лістинг (redirectmanag.php):

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.25.121.6.000 – Лр.7	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		5

```

<?php
ob_start();

$filename = 'redirects.json';

//перевірка існування конфігураційного файлу
if (!file_exists($filename)) {
    http_response_code(500);
    echo "<h1>Помилка сервера</h1><p>Файл перенаправлень не знайдено.</p>";
    exit;
}

$redirects = json_decode(file_get_contents($filename), true);

if (!is_array($redirects)) {
    http_response_code(500);
    echo "<h1>Помилка сервера</h1><p>Невірний формат redirects.json.</p>";
    exit;
}

$requestUri = parse_url($_SERVER['REQUEST_URI'], PHP_URL_PATH);

$scriptName = dirname($_SERVER['SCRIPT_NAME']);
$relativeUri = '/' . ltrim(str_replace($scriptName, '', $requestUri), '/');

//перевірка відповідності
if (array_key_exists($relativeUri, $redirects)) {
    $target = $redirects[$relativeUri];

    if ($target === "/404") {
        http_response_code(404);
        echo "<h1>Сторінка не знайдена</h1><p>Ця сторінка більше не існує.</p>";
    } else {
        header("Location: $target", true, 301);
        exit;
    }
} else {
    echo "<h1>Ласкаво просимо</h1><p>Це сторінка без перенаправлень.</p>";
}

//http://localhost/labs/lab_7/redirectmanag.php/deprecated

```

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.25.121.6.000 – Лр.7	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		6

### Завдання 5. Динамічне підключення заголовків без видимого виводу:

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

#### 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.

#### Приклад використання:

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

#### Підказка:

- Використовуйте **header()** всередині методу **setStatus()**.
- Використовуйте **ob\_start()**, **ob\_clean()** перед виводом контенту.

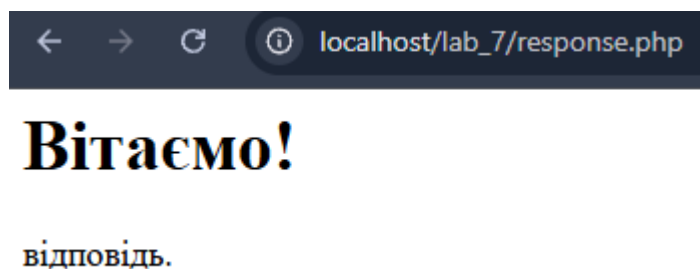


Рис. 9. Результат виконання завдання

Лістинг (response.php):

```
<?php  
class Response {  
    public function setStatus($code) { //встановлюєм статус  
        http_response_code($code);  
    }  
  
    public function addHeader($header) { //заголовки до відповіді  
        header($header);  
    }  
  
    public function send($content) { //очищення буфера
```

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.25.121.6.000 – Лр.7	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		7

```

        ob_clean();
        echo $content;
    }
}

$response = new Response();
$response->setStatus(200);
$response->addHeader("Content-Type: text/html");
$response->send("<h1>Вітаємо!</h1><p>Відповідь.</p>");
?>

```

## Завдання 6. Аналіз трафіку та відповідей сервера:

Розробіть **traffic\_logger.php**, який запише інформацію про всі запити в базу даних.

### Що зберігати в БД?

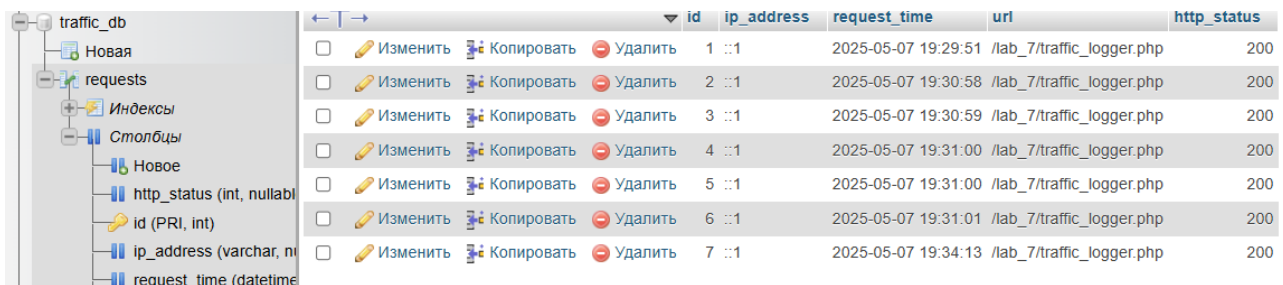
- IP-адресу
- Час запиту
- Запитаний URL
- Відповідний HTTP-статус

### 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.

### Підказка:

- Використовуйте **MySQL (або SQLite)** для збереження даних.
- Функція **http\_response\_code()** допоможе отримати поточний статус.



	id	ip_address	request_time	url	http_status
<input type="checkbox"/> Изменить <input type="checkbox"/> Копировать <input type="checkbox"/> Удалить	1	::1	2025-05-07 19:29:51	/lab_7/traffic_logger.php	200
<input type="checkbox"/> Изменить <input type="checkbox"/> Копировать <input type="checkbox"/> Удалить	2	::1	2025-05-07 19:30:58	/lab_7/traffic_logger.php	200
<input type="checkbox"/> Изменить <input type="checkbox"/> Копировать <input type="checkbox"/> Удалить	3	::1	2025-05-07 19:30:59	/lab_7/traffic_logger.php	200
<input type="checkbox"/> Изменить <input type="checkbox"/> Копировать <input type="checkbox"/> Удалить	4	::1	2025-05-07 19:31:00	/lab_7/traffic_logger.php	200
<input type="checkbox"/> Изменить <input type="checkbox"/> Копировать <input type="checkbox"/> Удалить	5	::1	2025-05-07 19:31:00	/lab_7/traffic_logger.php	200
<input type="checkbox"/> Изменить <input type="checkbox"/> Копировать <input type="checkbox"/> Удалить	6	::1	2025-05-07 19:31:01	/lab_7/traffic_logger.php	200
<input type="checkbox"/> Изменить <input type="checkbox"/> Копировать <input type="checkbox"/> Удалить	7	::1	2025-05-07 19:34:13	/lab_7/traffic_logger.php	200

Рис. 10. БД traffic\_db



# Статистика за останню добу

Загальна кількість запитів: 13

404 помилок: 0

Відсоток 404: 0%

Рис. 11. Статистика запитів

Лістинг (traffic\_logger.php):

```
<?php
ob_start();
$startTime = microtime(true);

$pdo = new PDO("mysql:host=localhost;dbname=traffic_db", "root", "");

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

register_shutdown_function(function() use ($pdo, $ip, $time, $url) {
    $status = http_response_code(); //HTTP статус

    $stmt = $pdo->prepare("INSERT INTO requests (ip_address, request_time, url,
http_status) VALUES (?, ?, ?, ?)");
    $stmt->execute([$ip, $time, $url, $status]);
});
?>
```

Лістинг (stats.php):

```
<?php
$pdo = new PDO("mysql:host=localhost;dbname=traffic_db", "root", "");

$since = date('Y-m-d H:i:s', strtotime('-1 day'));

$total = $pdo->query("SELECT COUNT(*) FROM requests WHERE request_time >= '$since')->fetchColumn();
```

```

$errors404 = $pdo->query("SELECT COUNT(*) FROM requests WHERE http_status = 404 AND
request_time >= '$since'")->fetchColumn();

// %
$percent = $total > 0 ? ($errors404 / $total) * 100 : 0;

echo "<h1>Статистика за останню добу</h1>";
echo "<p>Загальна кількість запитів: $total</p>";
echo "<p>404 помилок: $errors404</p>";
echo "<p>Відсоток 404: " . round($percent, 2) . "%</p>";

if ($percent > 10) {
    echo "<p style='color:red;'> Увага: більше 10% запитів – 404.</p>";
}
?>

```

Висновки: в ході виконання роботи я навчився працювати з HTTP запитам.

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.25.121.6.000 – Лр.7	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		10