

ЛАБОРАТОРНА РОБОТА № 7

Встановлення HTTP-статус кодів та функції керування буферизацією виводу

Мета: Навчитися встановлювати HTTP-статус кодів та вивчити 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.

					ДУ «Житомирська політехніка».25.121.08.000 – Лр7			
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Дзінзілевич Д.О.			Звіт з лабораторної роботи		Літ.	Арк.
Перевір.		Ковтун В.В.						1
Керівник								18
Н. контр.							ФІКТ Гр. ІПЗ-23-2[1]	
Зав. каф.								

Лістинг коду:

cache_page.php

```
<title>CheckPageStatus</title>
<link rel="stylesheet" href="/templates/styles/style.css">
<?php
require __DIR__ . '/../t6/traffic_logger.php';
function get_http_status($url) {
    $headers = get_headers($url, 1);
    return intval(substr($headers[0], 9, 3));
}

function fetch_page_content($url) {
    ob_start();
    $content = file_get_contents($url);
    if ($content === false) {
        ob_end_clean();
        return false;
    }
    echo $content;
    $content = ob_get_contents();
    ob_end_clean();
    return $content;
}

function cache_page($url, $cache_file = 'cache.html') {
    $status = get_http_status($url);

    if ($status === 404) {
        if (file_exists($cache_file)) {
            unlink($cache_file);
        }
        echo "Сторінка не знайдена (404)";
        return;
    }

    if (file_exists($cache_file)) {
        echo file_get_contents($cache_file);
        return;
    }

    if ($status === 200) {
        $content = fetch_page_content($url);

        if ($content !== false) {
            file_put_contents($cache_file, $content);
            echo $content;
        } else {
            echo "Помилка при отриманні вмісту сторінки";
        }
    } else {
        echo "Невідомий статус: $status";
    }
}

$page_url = 'https://example.com/page-to-cache';
cache_page($page_url);
```

		Дзінзілевич Д.О.			ДУ «Житомирська політехніка».25.121.08.000 – Лр7	Арк.
		Ковтун В.В.				2
Змн.	Арк.	№ докум.	Підпис	Дата		

Результат виконання програми:

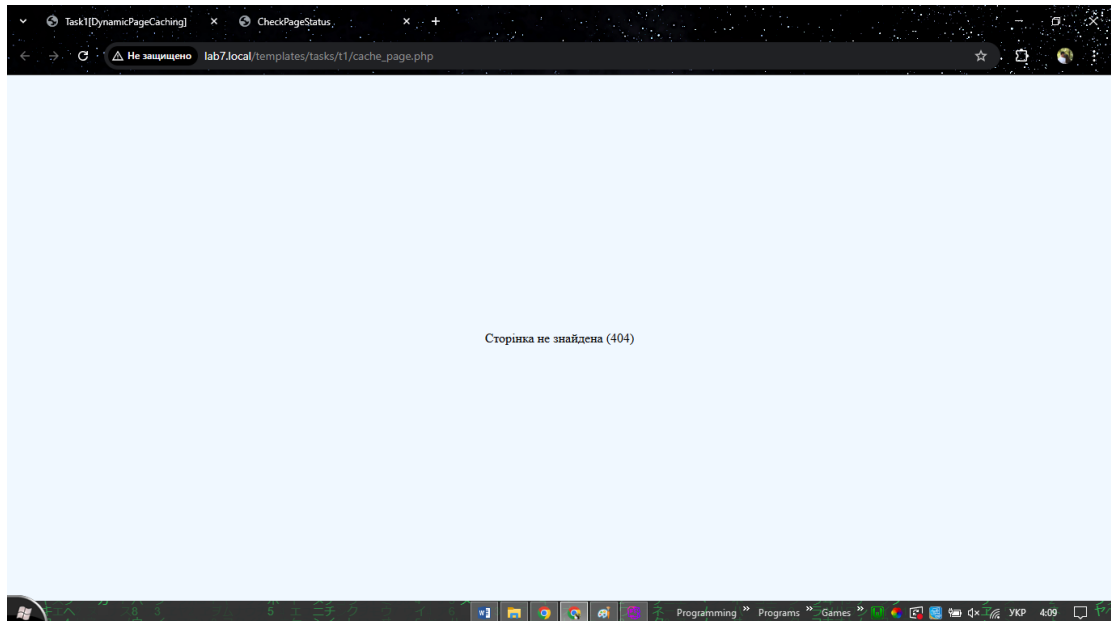


Рис.1.1 Поведінка сторінки при статусі 404

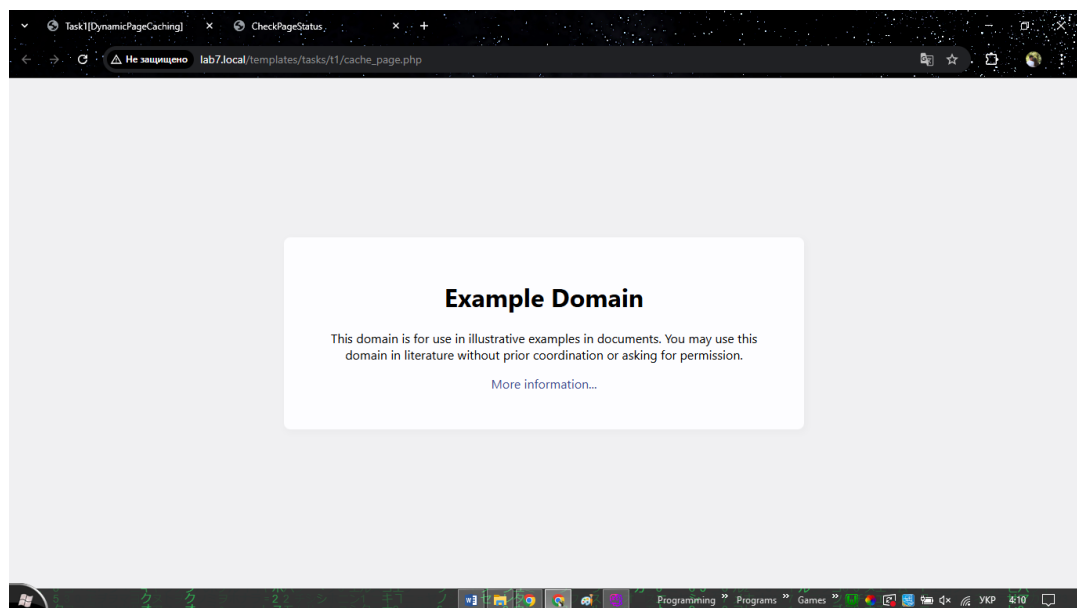


Рис.1.2 Поведінка сторінки при статусі 200

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:

1. Логувати IP-адресу користувача в файл **requests.log**.

		Дзінзілевич Д.О.			ДУ «Житомирська політехніка».25.121.08.000 – Пр7	Арк.
		Ковтун В.В.				3
Змн.	Арк.	№ докум.	Підпис	Дата		

2. Якщо користувач зробив більше **5 запитів за хвилину**, встановлювати статус **429 Too Many Requests** і рекомендувати спробувати пізніше.
3. 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.

Лістинг коду:

rate_limit.php

```
<title>Log_IP_Address</title>
<link rel="stylesheet" href="/templates/styles/style.css">
<?php
require __DIR__ . '/../t6/traffic_logger.php';
$limit = 5;
$logFile = 'requests.log';
$currentTime = time();

$clientIP = $_SERVER['REMOTE_ADDR'] ?? 'невідомий';

ob_start();

function cleanOldEntries($logFile, $currentTime) {
    if (!file_exists($logFile)) {
        return;
    }

    $lines = file($logFile, FILE_IGNORE_NEW_LINES | FILE_SKIP_EMPTY_LINES);
    if ($lines === false) {
        return;
    }

    $updatedLines = [];
    foreach ($lines as $line) {
        list($ip, $timestamp) = explode('|', $line, 2);
        if ($currentTime - intval($timestamp) <= 60) {
            $updatedLines[] = $line;
        }
    }

    file_put_contents($logFile, implode(PHP_EOL, $updatedLines) . PHP_EOL);
}

function countRecentRequests($logFile, $clientIP, $currentTime) {
    if (!file_exists($logFile)) {
        return 0;
    }

    $count = 0;
    $lines = file($logFile, FILE_IGNORE_NEW_LINES | FILE_SKIP_EMPTY_LINES);
    if ($lines === false) {
        return 0;
    }
}
```

		Дзінзілевич Д.О.			ДУ «Житомирська політехніка».25.121.08.000 – Лр7	Арк.
		Ковтун В.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		4

```

    }

    foreach ($lines as $line) {
        list($ip, $timestamp) = explode('|', $line, 2);
        if ($ip === $clientIP && $currentTime - intval($timestamp) <= 60) {
            $count++;
        }
    }

    return $count;
}

cleanOldEntries($logFile, $currentTime);

file_put_contents($logFile, "$clientIP|$currentTime" . PHP_EOL, FILE_APPEND);

$requestCount = countRecentRequests($logFile, $clientIP, $currentTime);

if ($requestCount > $limit) {
    ob_end_clean();

    header('HTTP/1.1 429 Too Many Requests');
    header('Retry-After: 60');

    echo "Error 429: Забагато запитів. Будь ласка, спробуйте пізніше.";
    exit;
}

header('HTTP/1.1 200 OK');
echo "Ласкаво просимо! Це ваш запит №" . $requestCount . " за останню хвилину.";
ob_end_flush();

```

Результат виконання програми:

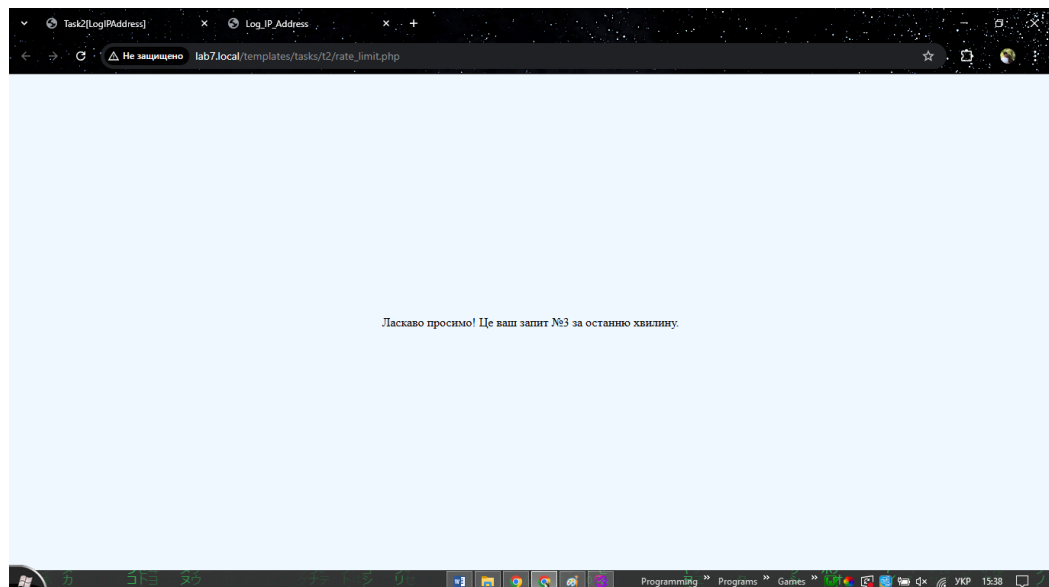
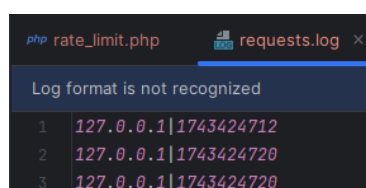


Рис.2.1 Підрахунок запитів за останню хвилину



		Дзінзілевич Д.О.			ДУ «Житомирська політехніка».25.121.08.000 – Лр7	Арк.
		Ковтун В.В.				5
Змн.	Арк.	№ докум.	Підпис	Дата		

Рис.2.2 Запис даних у відповідний файл

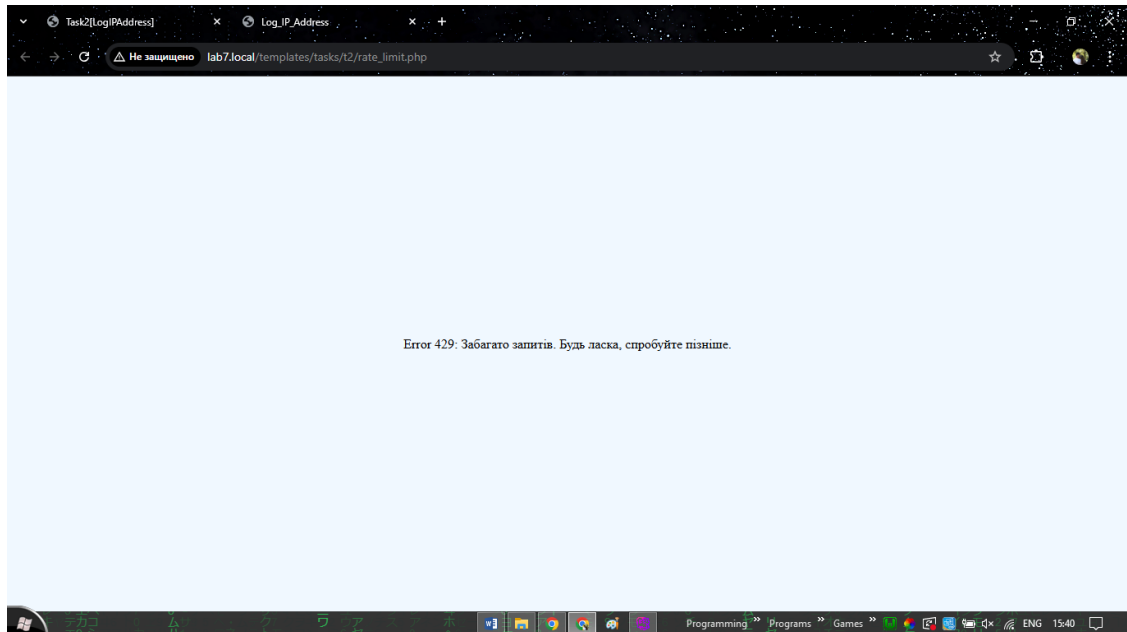


Рис.2.3 Поведінка сайту при перевищенні ліміту запитів

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:

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

Підказка:

- **error_get_last()** допоможе отримати останню помилку.
- Використовуйте **ob_start()**, щоб уникнути передчасного виводу.

		Дзінзілевич Д.О.			ДУ «Житомирська політехніка».25.121.08.000 – Пр7	Арк.
		Ковтун В.В.				6
Змн.	Арк.	№ докум.	Підпис	Дата		

Лістинг коду:

error_hendler.php

```
<link rel="stylesheet" href="/templates/styles/style.css">
<?php
require DIR . '/../t6/traffic logger.php';
ob_start();
function handleFatalError() {
    $error = error_get_last();

    if ($error !== null && $error['type'] === E_ERROR) {
        ob_clean();

        header($_SERVER['SERVER_PROTOCOL'] . ' 500 Internal Server Error', true,
500);
        header('Content-Type: text/html; charset=utf-8');
        date_default_timezone_set('Europe/Kyiv');
        $timeFixed = date('H:i', time() + 3600);
        echo '<!DOCTYPE html>
<html lang="uk">
<head>
<title>500_ServerError</title>
<style>
body { font-family: Arial, sans-serif; text-align: center; }
h1 { color: #d9534f; }
.info { background: #f9f9f9; padding: 20px; margin: 20px auto;
max-width: 500px; }
</style>
</head>
<body>
<h1>500 Помилка сервера</h1>
<div class="info">
<p>Вибачте, сталася критична помилка.</p>
<p>Спробуйте оновити сторінку або повернутися пізніше.</p>
<p><strong>Проблема буде вирішена до ' . $timeFixed .
'</strong></p>
</div>
</body>
</html>';
        exit();
    }
}
register_shutdown_function('handleFatalError');
//=====
echo '<title>Welcome!</title>';
echo "<h1>Ласкаво просимо на сайт!</h1>";
echo "<p>Це головна сторінка.</p>";
//someFunction();
$message = "Усе працює коректно!";
echo "<p>$message</p>";
//=====
header($_SERVER['SERVER_PROTOCOL'] . ' 200 OK', true, 200);
ob_end_flush();
```

		Дзінзілевич Д.О.			ДУ «Житомирська політехніка».25.121.08.000 – Лр7	Арк.
		Ковтун В.В.				7
Змн.	Арк.	№ докум.	Підпис	Дата		

Результат виконання програми:

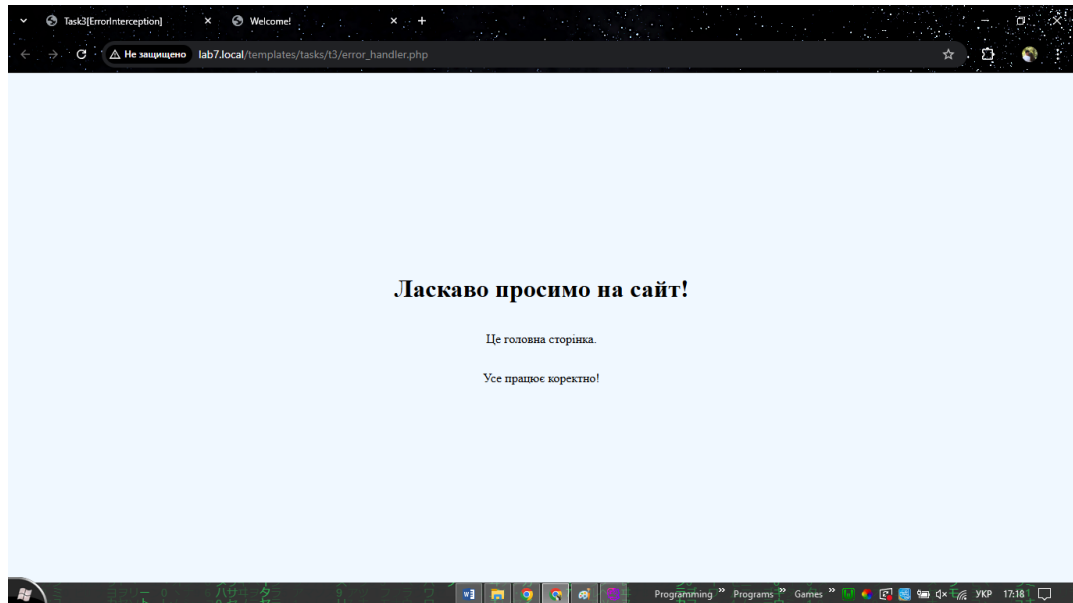


Рис.3.1 Коректна робота сторінки

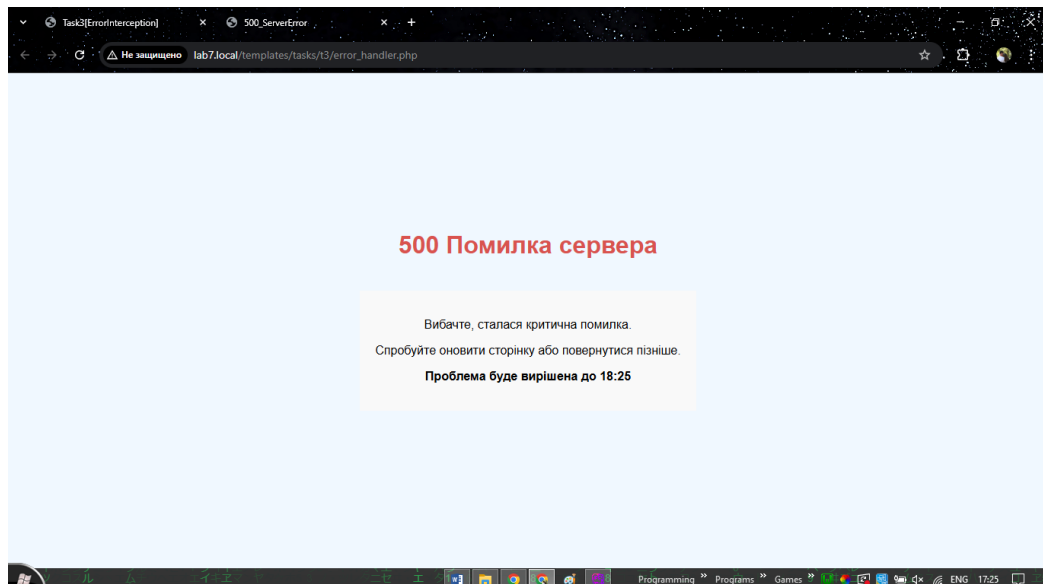


Рис.3.2 Поведінка сторінки при виникненні помилки

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"  
}
```

		Дзінзілевич Д.О.			ДУ «Житомирська політехніка».25.121.08.000 – Лр7	Арк.
		Ковтун В.В.				8
Змн.	Арк.	№ докум.	Підпис	Дата		

2. Якщо користувач заходить на **/old-page**, його потрібно перенаправити на **/new-page** зі статусом **301 Moved Permanently**.
3. Якщо сторінка **/deprecated**, видавати **404 Not Found** та відповідне повідомлення.
4. 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.

Лістинг коду:

redirect_manager.php

```
<link rel="stylesheet" href="/templates/styles/style.css">
<?php
ob_start();

$redirects = json_decode(file_get_contents(__DIR__ . "/redirects.json"), true);

$request_uri = str_replace("/templates/tasks/t4", "",
parse_url($_SERVER['REQUEST_URI'], PHP_URL_PATH));

if (isset($redirects[$request_uri])) {
    $destination = $redirects[$request_uri];

    if ($destination === "/404") {
        http_response_code(404);
        echo "<title>Error404!</title>";
        echo "404 Not Found - Сторінку не знайдено.";
    } else {
        $query_string = $_SERVER['QUERY_STRING'] ? '?' . $_SERVER['QUERY_STRING']
: '';
        header("Location: /templates/tasks/t4" . $destination . $query_string,
true, 301);
        exit();
    }
}

ob_end_flush();
```

redirect.json

```
{
    "/old-page": "/new-page",
    "/deprecated": "/404"
}
```

new-page.php

```
<?php
require_once 'redirect_manager.php';
?>
<!DOCTYPE html>
<html>
```

		Дзінзілевич Д.О.			ДУ «Житомирська політехніка».25.121.08.000 – Лр7	Арк.
		Ковтун В.В.				9
Змн.	Арк.	№ докум.	Підпис	Дата		

```

<head>
  <title>New Page</title>
</head>
<body>
<h1>Ви були перенаправлені</h1>
<p>Параметри: <?php print_r($_GET); ?></p>
<a href="/templates/tasks/t4/">Повернутися</a>
</body>
</html>

```

index.php

```

<?php
require_once 'redirect_manager.php';
require __DIR__ . '/../t6/traffic_logger.php';
?>
<!DOCTYPE html>
<html>
<head>
  <title>Redirect Test</title>
</head>
<body>
<h1>Тестування системи перенаправлень</h1>
<h3>Тестові посилання:</h3>
<ul>
  <li><a href="/templates/tasks/t4/old-page">/old-page (301 на /new-
page)</a></li>
  <li><a href="/templates/tasks/t4/deprecated">/deprecated (404
помилка)</a></li>
</ul>
</body>
</html>

```

Результат виконання програми:

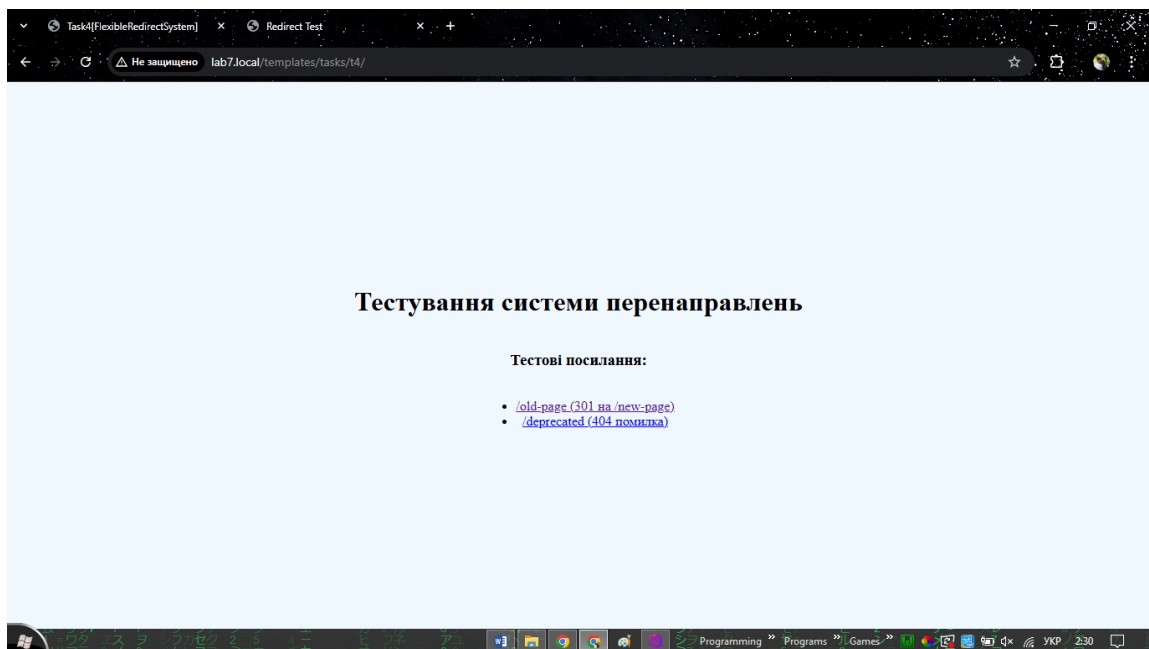


Рис.4.1 Вигляд основної сторінки

		Дзінзілевич Д.О.			ДУ «Житомирська політехніка».25.121.08.000 – Лр7	Арк.
		Ковтун В.В.				10
Змн.	Арк.	№ докум.	Підпис	Дата		

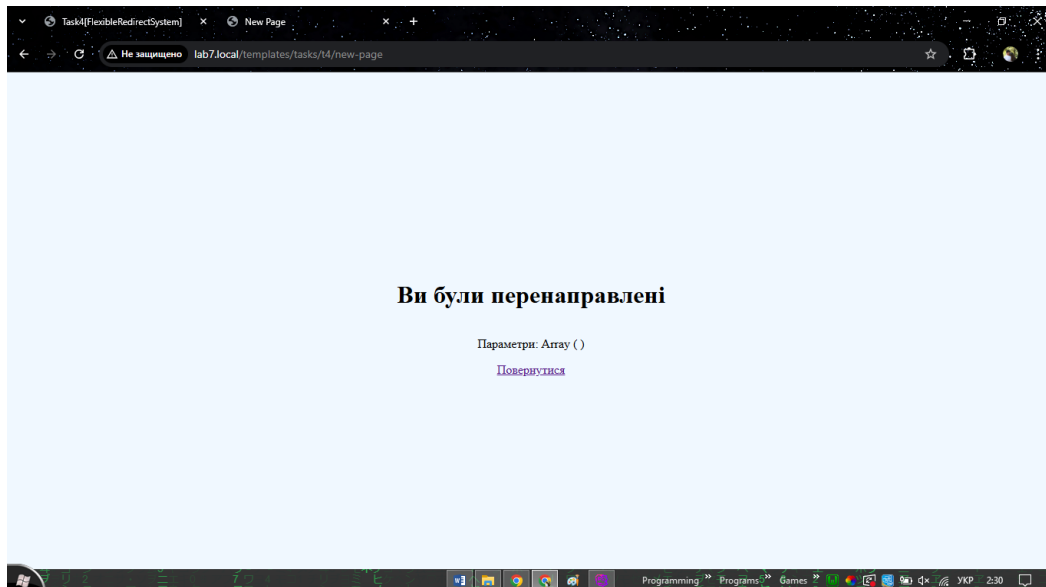


Рис.4.2 Перенаправлення користувача при переході на /old-page

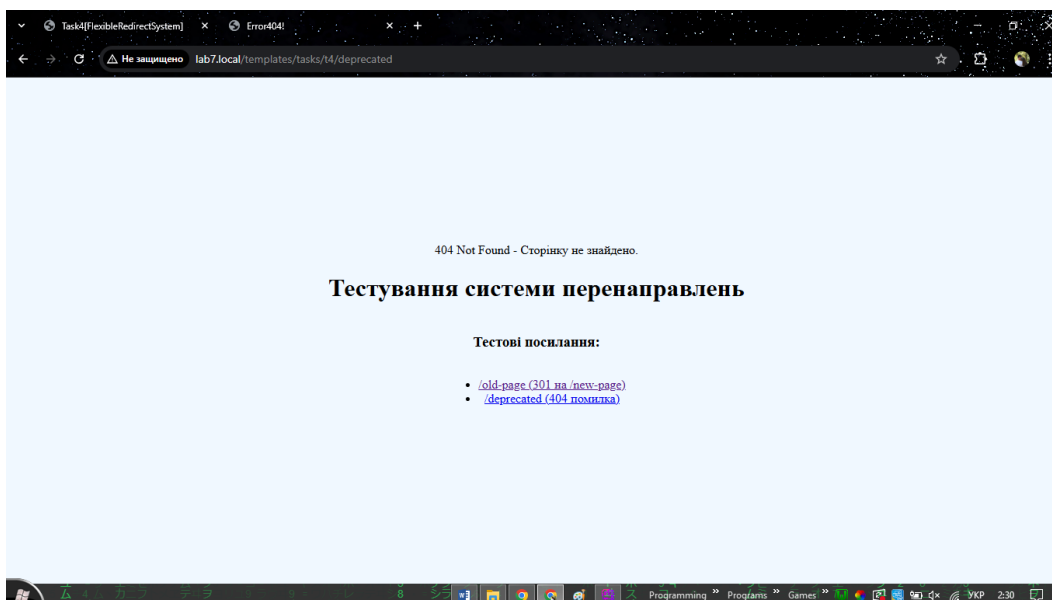


Рис.4.3 Виведення помилки при переході на /deprecated

Завдання 5: Динамічне підключення заголовків без видимого виводу.

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

Функціонал класу:

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

		Дзінзілевич Д.О.			ДУ «Житомирська політехніка».25.121.08.000 – Лр7	Арк.
		Ковтун В.В.				11
Змн.	Арк.	№ докум.	Підпис	Дата		

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.

Лістинг коду:

Response.php

```
<?php  
namespace t5;  
class Response {  
    private $headers = [];  
    private $statusCode = 200;  
    /**  
     * Встановлює HTTP-статус відповіді  
     * @param int $code HTTP-код статусу  
     */  
    public function setStatus($code) {  
        $this->statusCode = (int)$code;  
        http_response_code($this->statusCode);  
    }  
    /**  
     * Додає HTTP-заголовок до відповіді  
     * @param string $header Рядок заголовка  
     */  
    public function addHeader($header) {  
        $this->headers[] = $header;  
    }  
    /**  
     * Надсилає відповідь з встановленими заголовками та вмістом  
     * @param string $content Тіло відповіді  
     */  
    public function send($content) {  
        if (ob_get_length() > 0) {  
            ob_clean();  
        }  
  
        if (!headers_sent()) {  
            foreach ($this->headers as $header) {  
                header($header);  
            }  
        }  
  
        echo $content;  
  
        if (function_exists('fastcgi_finish_request')) {  
            fastcgi_finish_request();  
        }  
    }  
}
```

		Дзінзілевич Д.О.			ДУ «Житомирська політехніка».25.121.08.000 – Лр7	Арк.
		Ковтун В.В.				12
Змн.	Арк.	№ докум.	Підпис	Дата		

test_responce.php

```
<?php
require_once 'Response.php';
use t5\Response;
require __DIR__ . '/../t6/traffic_logger.php';
$response = new Response();
$response->setStatus(200);
$response->addHeader("Content-Type: text/plain");
$response->send("Тестова відповідь");
```

Результат виконання програми:

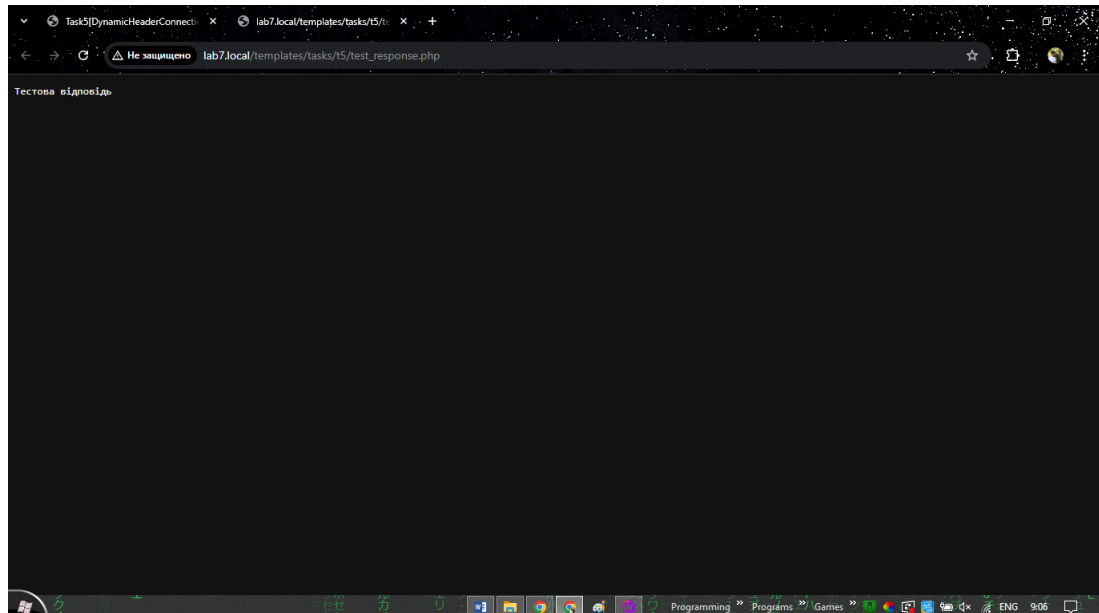


Рис.5.1 Виведення HTTP-заголовку

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.

		Дзінзілевич Д.О.			ДУ «Житомирська політехніка».25.121.08.000 – Лр7	Арк.
		Ковтун В.В.				13
Змн.	Арк.	№ докум.	Підпис	Дата		

Підказка:

- Використовуйте MySQL (або SQLite) для збереження даних.
- Функція `http_response_code()` допоможе отримати поточний статус.

База даних:

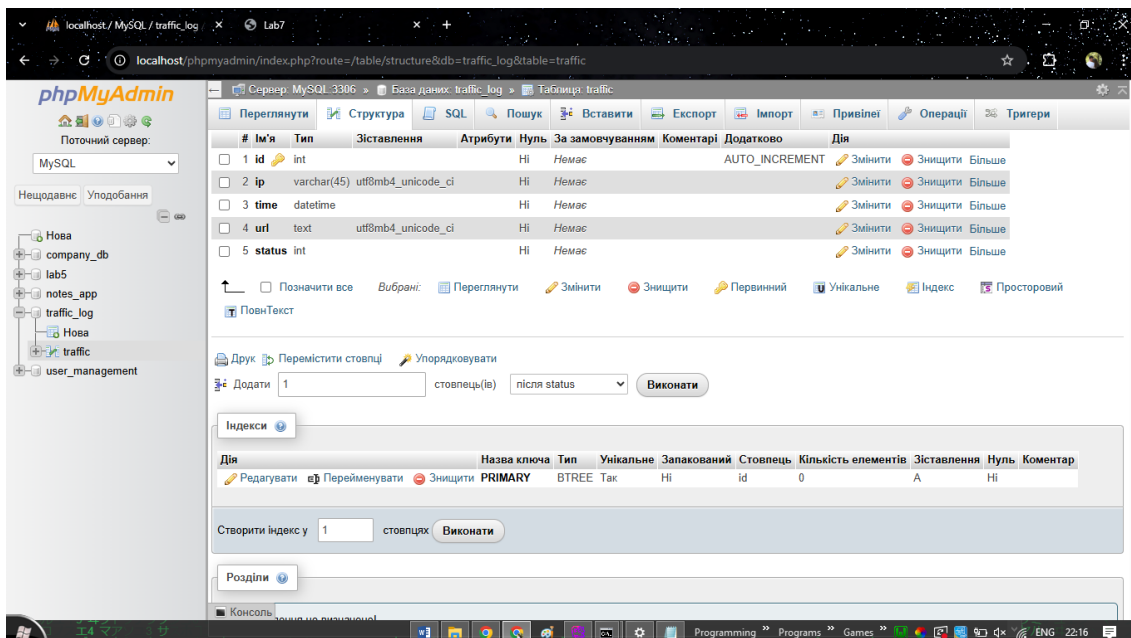


Рис.6.1 Вигляд використовуваної таблиці в БД

Лістинг коду:

index.php (Головна сторінка проекту)

```
<?php
if (!isset($_GET['task'])) {
    include('templates/parts/header.php');
    include('templates/main.php');
} else {
    $task = $_GET['task'];
    $path = 'templates/tasks/' . $task . '.php';

    if (file_exists($path)) {
        include($path);
    } else {
        http_response_code(404);
        header("HTTP/1.1 404 Not Found");

        include('templates/error404.php');
    }
}
require __DIR__ . '/templates/tasks/t6/traffic_logger.php';
include('templates/parts/footer.php');
```

db_connect.php

```
<?php
$dsn = "mysql:host=localhost;dbname=traffic_log;charset=utf8mb4";
$username = "root";
$password = "";
```

		Дзінзілевич Д.О.			ДУ «Житомирська політехніка».25.121.08.000 – Лр7	Арк.
		Ковтун В.В.				14
Змн.	Арк.	№ докум.	Підпис	Дата		

```
try {
    $pdo = new PDO($dsn, $username, $password, [
        PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION,
        PDO::ATTR_DEFAULT_FETCH_MODE => PDO::FETCH_ASSOC,
    ]);
} catch (PDOException $e) {
    die("Помилка підключення: " . $e->getMessage());
}
```

stats.php

```
<title>Stats</title>
<link rel="stylesheet" href="/templates/styles/style.css">
<?php
use PHPMailer\PHPMailer\PHPMailer;
use PHPMailer\PHPMailer\Exception;

require $_SERVER['DOCUMENT_ROOT'] . '/vendor/autoload.php';
require 'db_connect.php';

global $pdo;

$stmt = $pdo->prepare("SELECT COUNT(*) AS count_404 FROM traffic WHERE status =
404 AND time >= NOW() - INTERVAL 1 DAY");
$stmt->execute();
$count_404 = $stmt->fetchColumn();

$stmt = $pdo->prepare("SELECT COUNT(*) AS total_requests FROM traffic WHERE time
>= NOW() - INTERVAL 1 DAY");
$stmt->execute();
$total_requests = $stmt->fetchColumn();

$percentage = ($total_requests > 0) ? ($count_404 / $total_requests) * 100 : 0;

echo "Кількість 404 за добу: $count_404<br>";
echo "Відсоток 404: " . round($percentage, 2) . "%<br>";

if ($percentage > 10) {
    $admin_email = "ipz232 ddo@student.ztu.edu.ua";

    try {
        $mail = new PHPMailer(true);
        $mail->isSMTP();
        $mail->Host = 'smtp.gmail.com';
        $mail->SMTPAuth = true;
        $mail->Username = 'tomridzevel@gmail.com';
        $mail->Password = 'rcgb xqsd dtgn gpdh';
        $mail->SMTPSecure = PHPMailer::ENCRYPTION_STARTTLS;
        $mail->Port = 587;

        $mail->setFrom('tomridzevel@gmail.com', 'Monitoring System');
        $mail->addAddress($admin_email);

        $mail->CharSet = 'UTF-8';
        $mail->Encoding = 'base64';

        $mail->Subject = 'Попередження: високий рівень 404';
        $mail->Body = "Відсоток 404 помилок за останню добу: " .
round($percentage, 2) . "%";

        $mail->send();
        echo "Повідомлення адміністратору надіслано.";
    } catch (Exception $e) {
        echo "Помилка відправки листа: {$mail->ErrorInfo}";
    }
}
```

		Дзінзілевич Д.О.			ДУ «Житомирська політехніка».25.121.08.000 – Лр7	Арк.
		Ковтун В.В.				15
Змн.	Арк.	№ докум.	Підпис	Дата		

```
}  
}
```

traffic_logger.php3

```
<?php  
global $pdo;  
require 'db_connect.php';  
  
$ip = $_SERVER['REMOTE_ADDR'];  
$time = date('Y-m-d H:i:s');  
$url = $_SERVER['REQUEST_URI'];  
$status = http_response_code();  
error_log("Запит: $url | Статус: $status");  
  
$stmt = $pdo->prepare("INSERT INTO traffic (ip, time, url, status) VALUES (?, ?,  
?, ?)");  
if (!$stmt->execute([$ip, $time, $url, $status])) {  
    print_r($stmt->errorInfo());  
}
```

Результат виконання програми:

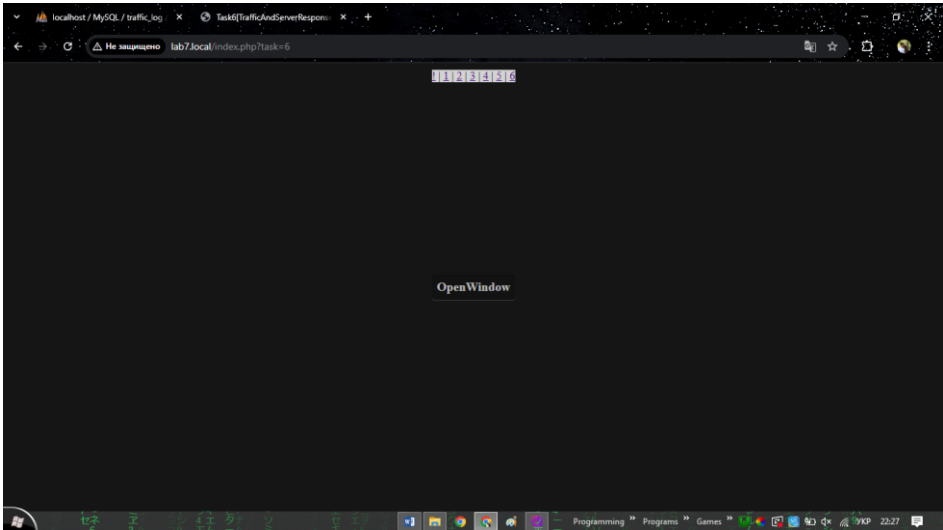


Рис.6.2 Відкриваємо декілька сторінок проекту

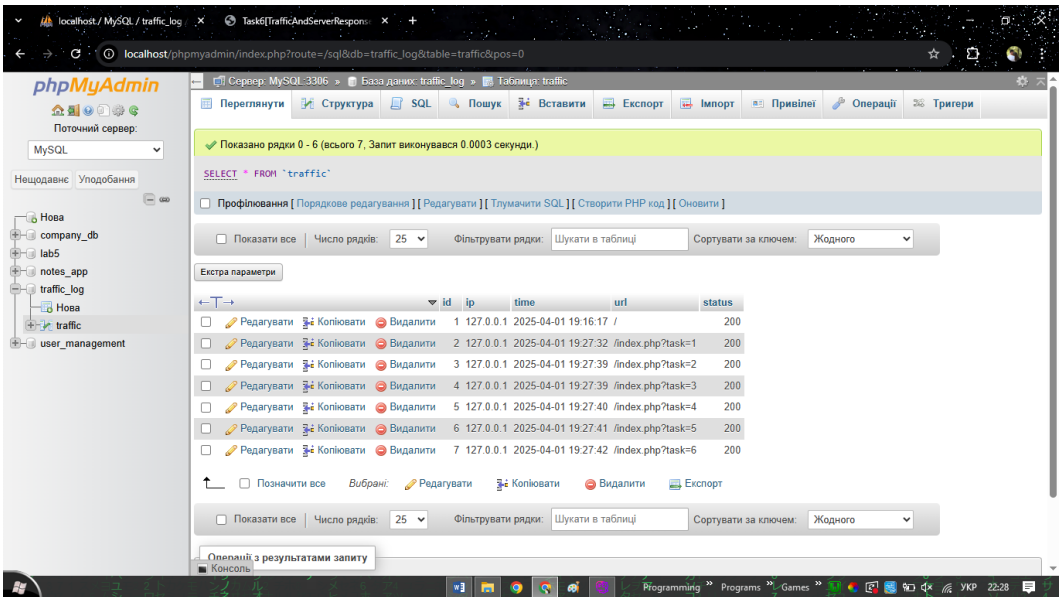


Рис.6.3 Перевіряємо БД

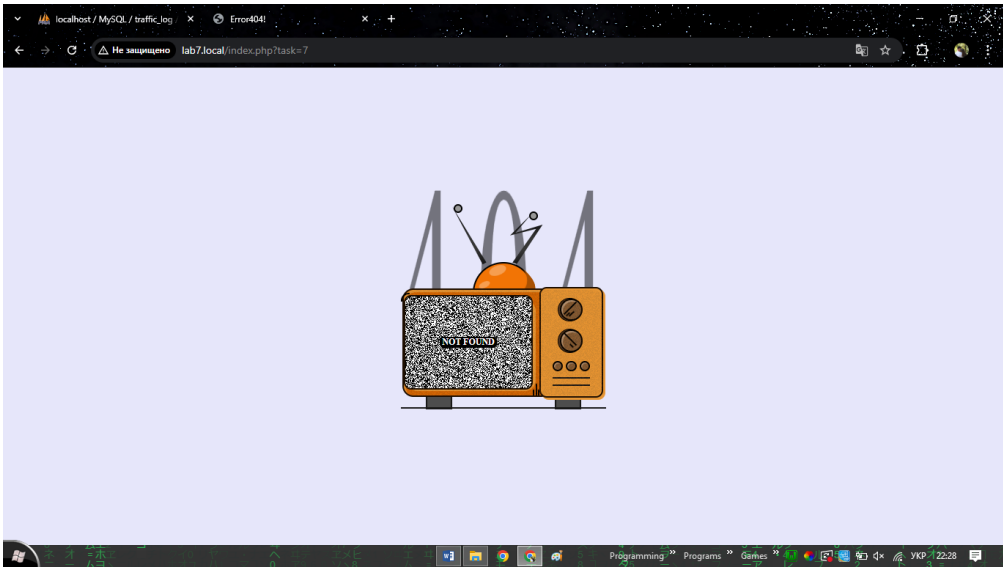


Рис.6.4 Відкриваємо сторінку з помилкою (якої не існує)

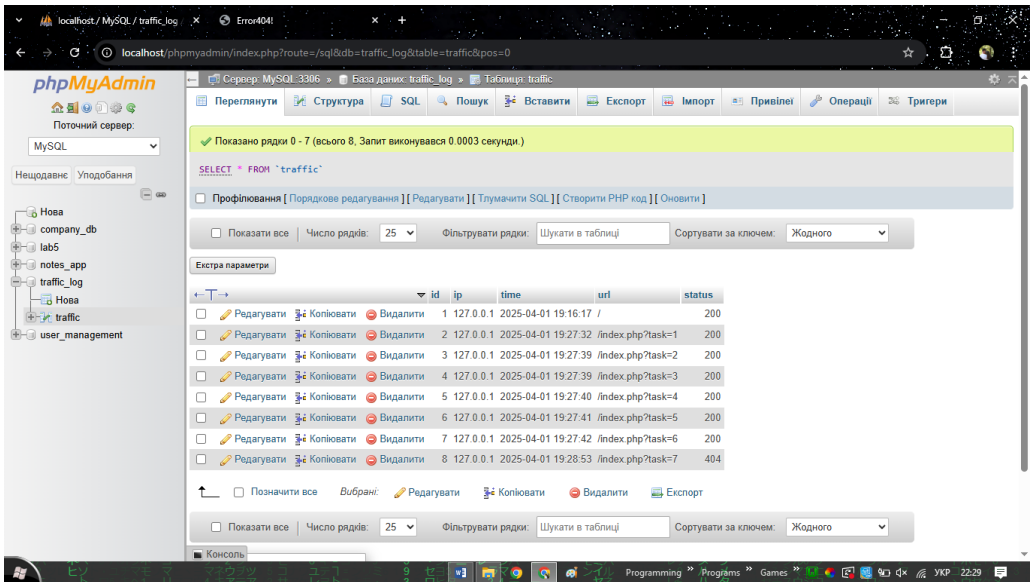


Рис.6.5 Перевіряємо БД

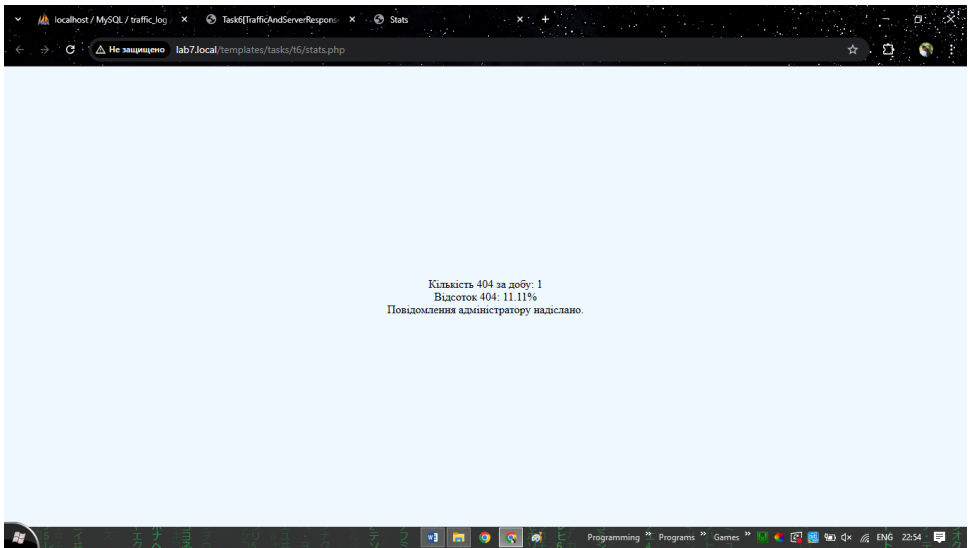


Рис.6.6 Переглядаємо статистику

		Дзінзілевич Д.О.			ДУ «Житомирська політехніка».25.121.08.000 – Лр7	Арк.
		Ковтун В.В.				17
Змн.	Арк.	№ докум.	Підпис	Дата		

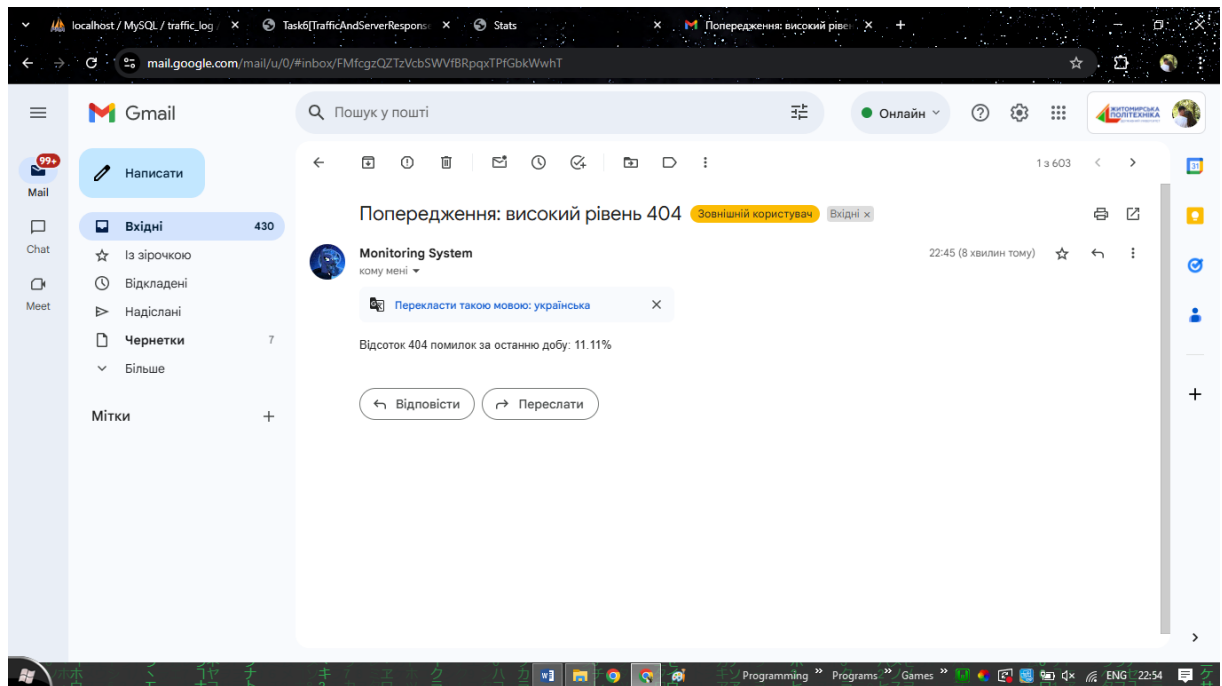


Рис.6.7 Повідомлення про відсоток помилок на пошті

Висновки: Навчилися встановлювати HTTP-статус кодів та вивчили функції керування буферизацією виводу.

		Дзінзілевич Д.О.			ДУ «Житомирська політехніка».25.121.08.000 – Пр7	Арк.
		Ковтун В.В.				18
Змн.	Арк.	№ докум.	Підпис	Дата		