

ЛАБОРАТОРНА РОБОТА № 7

Тема: Асинхронні запити на сервер та їх обробка

Мета роботи: ознайомитися з асинхронними запитами на сервер та їх обробкою

Завдання 1: Система керування користувачами

Git repository: <https://git.ztu.edu.ua/ipz/2023-2027/ipz-23-1/semenchuk-oleksiy/backend/-/tree/lab6>

Висновок: У ході виконання роботи було розглянуто основи роботи з базою даних MySQL за допомогою бібліотеки PDO у мові PHP. Було вивчено підключення до бази даних, виконання запитів, обробку результатів, а також використання підготовлених запитів для забезпечення безпеки та оптимізації роботи з даними.

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.

Результат:

```
[/var/www/labs] -> http localhost/task1/cache_page.php
HTTP/1.1 404 Not Found
Connection: Keep-Alive
Content-Length: 0
Content-Type: text/html; charset=UTF-8
Date: Thu, 10 Apr 2025 07:52:29 GMT
Keep-Alive: timeout=5, max=100
Server: Apache/2.4.58 (Ubuntu)
```

Рис. 1 Код 404

					ДУ«Житомирська політехніка».25.121.27.000 – Лр7			
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Семенчук О.А.			Звіт з лабораторної роботи		Літ.	Арк.
Перевір.		Дмитренко І.А.						1
Керівник							ФІКТ Гр. ІПЗ-23-1[2]	
Н. контр.								
Зав. каф.								
							18	

```
[/var/www/labs] -> http localhost/task1/cache_page.php id=5
HTTP/1.1 200 OK
Connection: Keep-Alive
Content-Length: 14
Content-Type: text/html; charset=UTF-8
Date: Thu, 10 Apr 2025 07:52:58 GMT
Keep-Alive: timeout=5, max=100
Server: Apache/2.4.58 (Ubuntu)

Cache created!
```

Рис. 2 Код 200

```
[/var/www/labs] -> http localhost/task1/cache_page.php id=5
HTTP/1.1 200 OK
Connection: Keep-Alive
Content-Encoding: gzip
Content-Length: 131
Content-Type: text/html; charset=UTF-8
Date: Thu, 10 Apr 2025 07:53:19 GMT
Keep-Alive: timeout=5, max=100
Server: Apache/2.4.58 (Ubuntu)
Vary: Accept-Encoding

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Page</title>
</head>
<body>
  <h1>User id 5</h1>
</body>
</html>
```

Рис. 3 Повторний код 200

Лістинг файлу cache_page.php:

```
<?php

ini_set('display_errors', 1);
ini_set('display_startup_errors', 1);
error_reporting(E_ALL);

$cache_file_path = "cache.html";

$data = json_decode(file_get_contents("php://input"));
```

		Семенчук О.А.			ДУ «Житомирська політехніка».25.121.27.000 – Лр7	Арк.
		Дмитренко І. А.				2
Змн.	Арк.	№ докум.	Підпис	Дата		

```

$id = $data->id ?? null;

if (!isset($id)  $id < 1  $id > 10)
{
    if (file_exists($cache_file_path))
    {
        echo "Cache deleted!";
        unlink($cache_file_path);
    }

    http_response_code(404);
    exit;
}

if (file_exists($cache_file_path))
{
    echo file_get_contents($cache_file_path);
    http_response_code(200);
    exit;
}

ob_start();

?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Page</title>
</head>
<body>
    <h1>User id <?=$id?></h1>
</body>
</html>

<?php

$content = ob_get_contents();
ob_clean();

if (http_response_code() == 200)
{
    echo "Cache created!";
    file_put_contents($cache_file_path, $content);
}

?>

```

		Семенчук О.А.			ДУ «Житомирська політехніка».25.121.27.000 – Лр7	Арк.
		Дмитренко І. А.				
Змн.	Арк.	№ докум.	Підпис	Дата		3

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.

```
[/var/www/labs] -> http localhost/task2/rate_limit.php
HTTP/1.1 200 OK
Connection: Keep-Alive
Content-Encoding: gzip
Content-Length: 126
Content-Type: text/html; charset=UTF-8
Date: Thu, 10 Apr 2025 07:54:52 GMT
Keep-Alive: timeout=5, max=100
Server: Apache/2.4.58 (Ubuntu)
Vary: Accept-Encoding

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title></title>
</head>
<body>
  <h1>Some page</h1>
</body>
</html>

[/var/www/labs] -> http localhost/task2/rate_limit.php
HTTP/1.1 429 Too Many Requests
Connection: Keep-Alive
Content-Length: 0
Content-Type: text/html; charset=UTF-8
Date: Thu, 10 Apr 2025 07:54:53 GMT
Keep-Alive: timeout=5, max=100
Server: Apache/2.4.58 (Ubuntu)
```

Рис. 4 Результат

Лістинг файлу `rate_limit.php`:

```
<?php
```

```
require_once "Logger.php";
```

		Семенчук О.А.			ДУ «Житомирська політехніка».25.121.27.000 – Пр7	Арк.
		Дмитренко І. А.				4
Змн.	Арк.	№ докум.	Підпис	Дата		

```

ini_set('display_errors', 1);
ini_set('display_startup_errors', 1);
error_reporting(E_ALL);

$IpAddress = $_SERVER["REMOTE_ADDR"];
$requestTime = $_SERVER["REQUEST_TIME"];

if (!Logger::ValidateRateLimit($IpAddress, $requestTime))
{
    http_response_code(429);
    exit;
}

Logger::AddLog($IpAddress, $requestTime);

?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title></title>
</head>
<body>
    <h1>Some page</h1>
</body>
</html>

```

Лістинг файлу Logger.php:

```

<?php

class Logger
{
    public static $FilePath = "requests.log";
    public static $TimeFormat = "m.d.Y H:i:s";
    public static $Separator = "|";
    public static $RateLimit = 5;
    public static $RateTimeLimitMinutes = 1;
}

```

		Семенчук О.А.			ДУ «Житомирська політехніка».25.121.27.000 – Лр7	Арк.
		Дмитренко І. А.				
Змн.	Арк.	№ докум.	Підпис	Дата		5

```

public static $LogKeepTimeMinutes = 1;
public static $DoLogCleanup = true;

public static function GetLogs(): mixed
{
    $lines = file(self::$FilePath, FILE_IGNORE_NEW_LINES |
FILE_SKIP_EMPTY_LINES);

    $Logs = array();

    for ($i = 0; $i < count($lines); $i++)
    {
        $data = explode(self::$Separator, $lines[$i]);

        $IpAddress = $data[0];
        $RequestTime = $data[1];

        $Logs[] = ["IpAddress" => $IpAddress, "RequestTime" =>
$RequestTime];
    }

    return $Logs;
}

public static function ValidateRateLimit(string $IpAddress):
bool
{
    self::CleanLogs();

    $Logs = self::GetLogs();

    $IpLogs = array();

    for ($i = 0; $i < count($Logs); $i++)
    {
        if ($Logs[$i]["IpAddress"] == $IpAddress)
        {
            $IpLogs[] = $Logs[$i];
        }
    }
}

```

		Семенчук О.А.			ДУ «Житомирська політехніка».25.121.27.000 – Лр7	Арк.
		Дмитренко І. А.				
Змн.	Арк.	№ докум.	Підпис	Дата		6

```

    }
}

$IpLogsWithinTime = array();

for ($i = 0; $i < count($IpLogs); $i++)
{
    $Difference = time() - $IpLogs[$i]["RequestTime"];

    $SecondsSince = $Difference / 60;

    if ($Difference < (60 * self::$RateTimeLimitMinutes))
    {
        $IpLogsWithinTime[] = $IpLogs[$i];
    }
}

return count($IpLogsWithinTime) < self::$RateLimit;
}

public static function AddLog(string $IpAddress, string $RequestTime): void
{
    $file = fopen(self::$FilePath, "a+");
    fwrite($file, $IpAddress . Logger::$Separator . $RequestTime . "\n");
    fclose($file);
}

public static function CleanLogs(): void
{
    if (!self::$DoLogCleanup) return;

    $Logs = self::GetLogs();

    $FilteredLogs = array();

    for ($i = 0; $i < count($Logs); $i++)

```

		Семенчук О.А.			ДУ «Житомирська політехніка».25.121.27.000 – Лр7	Арк.
		Дмитренко І. А.				7
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    {
        $Difference = time() - (int)$Logs[$i]["RequestTime"];

        if (($Difference / 60) > self::$LogKeepTimeMinutes)
        {
            continue;
        }

        $FilteredLogs[] = $Logs[$i];
    }

    $file = fopen(self::$FilePath, "w");

    for ($i = 0; $i < count($FilteredLogs); $i++)
    {
        fwrite($file, $FilteredLogs[$i]["IpAddress"] .
self::$Separator . $FilteredLogs[$i]["RequestTime"] . "\n");
    }

    fclose($file);
}
}

```

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`.

		Семенчук О.А.			ДУ «Житомирська політехніка».25.121.27.000 – Лр7	Арк.
		Дмитренко І. А.				8
Змн.	Арк.	№ докум.	Підпис	Дата		

Результат:

```
[/var/www/labs] -> http localhost/task3/
HTTP/1.1 200 OK
Connection: Keep-Alive
Content-Encoding: gzip
Content-Length: 129
Content-Type: text/html; charset=UTF-8
Date: Thu, 10 Apr 2025 07:57:28 GMT
Keep-Alive: timeout=5, max=100
Server: Apache/2.4.58 (Ubuntu)
Vary: Accept-Encoding

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title></title>
</head>

<body>
  <h1>Some page</h1>
</body>
</html>
```

Рис. 5 Результат без помилки

```
[/var/www/labs] -> http localhost/task3/ error=true
HTTP/1.0 500 Internal Server Error
Connection: close
Content-Length: 100
Content-Type: text/html; charset=UTF-8
Date: Thu, 10 Apr 2025 07:57:08 GMT
Server: Apache/2.4.58 (Ubuntu)

We are currently experiencing an issue.
We expect everything to be resolved by: 04.10.2025 12:57 PM
```

Рис. 6 Помилка

Лістинг файлу index.php

```
<?php
```

```
$EstimatedResolutionTime = time() + (60 * 60 * 5);
```

```
$DateFormat = "m.d.Y h:i A";
```

```
register_shutdown_function(function() use($DateFormat,
$EstimatedResolutionTime)
```

		Семенчук О.А.			ДУ «Житомирська політехніка».25.121.27.000 – Лр7	Арк.
		Дмитренко І. А.				9
Змн.	Арк.	№ докум.	Підпис	Дата		

```

{
    if (!error_get_last()) return;

    ob_clean();
    echo "We are currently experiencing an issue.\n";
    echo "We expect everything to be resolved by: " .
date($DateFormat, $EstimatedResolutionTime) . "\n";
    http_response_code(500);
});

ob_start();

?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title></title>
</head>

<?php
    $data = json_decode(file_get_contents("php://input"));

    if (isset($data->error) && $data->error === "true")
    {
        throw new Error("Exception is thrown!");
    }
?>

<body>
    <h1>Some page</h1>
</body>
</html>

<?php

```

		Семенчук О.А.			ДУ «Житомирська політехніка».25.121.27.000 – Лр7	Арк.
		Дмитренко І. А.				10
Змн.	Арк.	№ докум.	Підпис	Дата		

```

$page = ob_get_contents();
ob_clean();

echo $page;
http_response_code(200);

?>

```

Завдання 4. Гнучка система редиректів:

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

Умови:

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

```

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

```

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.

Результат:

```

[/var/www/labs] -> http localhost/task4/old-page
HTTP/1.1 301 Moved Permanently
Connection: Keep-Alive
Content-Length: 0
Content-Type: text/html; charset=UTF-8
Date: Thu, 10 Apr 2025 08:00:12 GMT
Keep-Alive: timeout=5, max=100
Location: /task4/new-page/
Server: Apache/2.4.58 (Ubuntu)

```

Рис. 7 old-page -> new-page

		Семенчук О.А.			ДУ «Житомирська політехніка».25.121.27.000 – Лр7	Арк.
		Дмитренко І. А.				11
Змн.	Арк.	№ докум.	Підпис	Дата		

```
[/var/www/labs] -> http localhost/task4/deprecated
HTTP/1.0 404 Not Found
Connection: close
Content-Length: 15
Content-Type: text/html; charset=UTF-8
Date: Thu, 10 Apr 2025 08:00:44 GMT
Server: Apache/2.4.58 (Ubuntu)

404 - Not Found
```

Рис. 8 deprecated -> 404

Лістинг файлу redirect_manager.php:

```
<?php

ini_set('display_errors', '1');
ini_set('display_startup_errors', '1');
error_reporting(E_ALL);

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

$uri = $_SERVER["REQUEST_URI"];

$uri_parts = explode("/", trim($uri, "/"));

$folder = $uri_parts[0] ?? "";
$path = isset($uri_parts[1]) ? "/" . $uri_parts[1] : "";

if (isset($redirects[$path]))
{
    $new_path = $redirects[$path];

    $new_uri = "/" . $folder . $new_path . "/";

    header("Location: " . $new_uri, true, 301);
    exit;
}
```

		Семенчук О.А.			ДУ «Житомирська політехніка».25.121.27.000 – Лр7	Арк.
		Дмитренко І. А.				12
Змн.	Арк.	№ докум.	Підпис	Дата		

```
header("HTTP/1.0 404 Not Found");
echo "404 - Not Found";
exit;
```

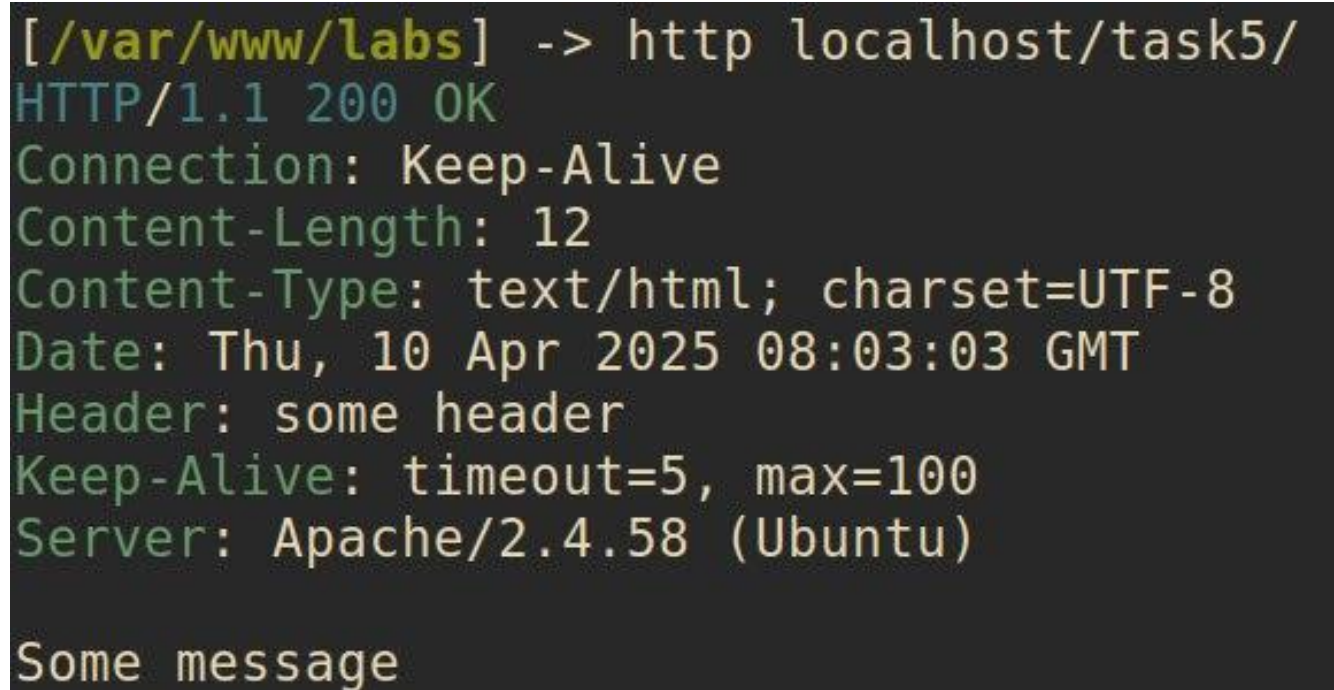
Завдання 5. Динамічне підключення заголовків без видимого виводу:

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.

Результат:



```
[/var/www/labs] -> http localhost/task5/
HTTP/1.1 200 OK
Connection: Keep-Alive
Content-Length: 12
Content-Type: text/html; charset=UTF-8
Date: Thu, 10 Apr 2025 08:03:03 GMT
Header: some header
Keep-Alive: timeout=5, max=100
Server: Apache/2.4.58 (Ubuntu)

Some message
```

Рис. 9 Результат

Лістиг файлу index.php:

```
<?php

ini_set('display_errors', '1');
ini_set('display_startup_errors', '1');
error_reporting(E_ALL);

require_once "Response.php";
```

		Семенчук О.А.			ДУ «Житомирська політехніка».25.121.27.000 – Лр7	Арк.
		Дмитренко І. А.				13
Змн.	Арк.	№ докум.	Підпис	Дата		

```

$response = new Response();

$response->setCode(200);
$response->addHeader("Header: some header");

$response->send("Some message");

```

Лістинг файлу Response.php:

```

<?php

class Response
{
    public function __construct()
    {
        ob_start();
    }

    public function setCode(int $code=200): void
    {
        http_response_code($code);
    }

    public function addHeader(string $header): void
    {
        header($header);
    }

    public function send(string $content): void
    {
        echo $content;
        $contents = ob_get_contents();

        ob_end_clean();
        echo $contents;
    }
}

```

		Семенчук О.А.			ДУ «Житомирська політехніка».25.121.27.000 – Лр7	Арк.
		Дмитренко І. А.				14
Змн.	Арк.	№ докум.	Підпис	Дата		

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:

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

Результат:

```
[/var/www/labs] -> http localhost/task6/traffic_logger.php
HTTP/1.1 200 OK
Connection: Keep-Alive
Content-Length: 0
Content-Type: text/html; charset=UTF-8
Date: Thu, 10 Apr 2025 08:05:21 GMT
Keep-Alive: timeout=5, max=100
Server: Apache/2.4.58 (Ubuntu)
```

Рис. 10 Результат traffic_logger.php

```
[/var/www/labs] -> http localhost/task6/stats.php
HTTP/1.1 200 OK
Connection: Keep-Alive
Content-Encoding: gzip
Content-Length: 165
Content-Type: text/html; charset=UTF-8
Date: Thu, 10 Apr 2025 08:05:43 GMT
Keep-Alive: timeout=5, max=100
Server: Apache/2.4.58 (Ubuntu)
Vary: Accept-Encoding

There were 0 requests with status code 404 in the past 24 ours.<br>
Warning! The percentage of 404 errors has exceeded 10% of the total requests.<br>
Please check the server logs and take the necessary actions.
```

Рис. 11 Результат traffic_stats.php

		Семенчук О.А.			ДУ «Житомирська політехніка».25.121.27.000 – Лр7	Арк.
		Дмитренко І. А.				15
Змн.	Арк.	№ докум.	Підпис	Дата		

Лістинг файлу traffic_logger.php:

```
<?php

require_once "db.php";

function add_request($requested_at, $requested_url, $status_code,
$ip_address)
{
    global $pdo;

    $prepare = $pdo->prepare("insert into requests (requested_at,
requested_url, status_code, ip_address)" .
        "values (:requested_at, :requested_url, :status_code,
:ip_address)");

    $prepare->execute([
        "requested_at" => $requested_at,
        "requested_url" => $requested_url,
        "status_code" => $status_code,
        "ip_address" => $ip_address
    ]);
}

$datetime_format = "Y-m-d H:i";

$ip_address = $_SERVER["REMOTE_ADDR"];
$requested_time = $_SERVER["REQUEST_TIME"];
$requested_url = $_SERVER["REQUEST_URI"];

$status_code = http_response_code();

$data = json_decode(file_get_contents("php://input"));

if (isset($data) && isset($data->code))
{
    $status_code = $data->code;
}
```

		Семенчук О.А.			ДУ «Житомирська політехніка».25.121.27.000 – Лр7	Арк.
		Дмитренко І. А.				
Змн.	Арк.	№ докум.	Підпис	Дата		16


```
$formatted_requested_time = date($datetime_format, $requested_time);

add_request($formatted_requested_time, $requested_url, $status_code,
$ip_address);
```

Лістинг файлу stats.php:

```
<?php

require_once "db.php";

function get_requests()
{
    global $pdo;

    $query = $pdo->query("select * from requests");

    return $query->fetchAll(PDO::FETCH_OBJ);
}

function get_not_found_within_seconds($seconds=null): mixed
{
    global $pdo;

    $query = $pdo->query("select * from requests where status_code =
404");

    $requests = $query->fetchAll(PDO::FETCH_OBJ);

    if ($seconds == null) return $requests;

    $requests = array_filter($requests, function($request) use
($seconds)
    {
        $unix_timestamp = strtotime($request->requested_at);

        $difference = time() - $unix_timestamp;
```

		Семенчук О.А.			ДУ «Житомирська політехніка».25.121.27.000 – Лр7	Арк.
		Дмитренко І. А.				17
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        if ($difference < $seconds) return true;

        return false;
    });

    return $requests;
}

$day_unix_timestamp = 60 * 60 * 24;

$recent_not_found_requests_count =
count(get_not_found_within_seconds($day_unix_timestamp));

echo "There were " . $recent_not_found_requests_count . " requests
with status code 404 in the past 24 ours." . "<br>\n";

$total_count = count(get_requests());
$total_count_not_found = count(get_not_found_within_seconds());

if ($total_count > 0 && $total_count_not_found / $total_count > 0.1)
{
    echo "Warning! The percentage of 404 errors has exceeded 10% of
the total requests.<br>\n" .
        "Please check the server logs and take the necessary
actions.";
}

```

Висновок: У результаті виконання роботи було розглянуто принципи здійснення асинхронних запитів до сервера та способи їх обробки. Отримані знання дозволяють ефективно працювати з асинхронною взаємодією між клієнтом і сервером, що є важливою складовою сучасних веб-застосунків.

		Семенчук О.А.			ДУ «Житомирська політехніка».25.121.27.000 – Лр7	Арк.
		Дмитренко І. А.				18
Змн.	Арк.	№ докум.	Підпис	Дата		