

Міністерство освіти і науки України  
НТУУ «Київський політехнічний інститут»  
Фізико-технічний інститут

**Системи та засоби інтерактивної аналітики**

Лабораторна робота №5

Збирання і попередня обробка інформації

Варіант №6

Виконав:  
Студент 4 курсу ФТІ  
групи ФБ-01

Сахній Назар

Перевірів:

---

## Лабораторна робота №5

### Збирання і попередня обробка інформації

**Тема:** Збирання і попередня обробка інформації.

**Завдання:** Навчитися **автоматизовано** отримувати інформацію з сторінки веб-сайту, попередньо її обробляти і записувати у базу даних для подальшої роботи.

№ варіанта	Завдання
6	46.125.249.79

### Робоче завдання

**0. Попередньо необхідно було виконати деякі налаштування робочого середовища.** Для реалізації автоматизованої обробки логів із веб-сайту було створено базу даних “autoprocessing”, а також необхідно було під’єднати бібліотеку GD із метою подальшого її використання для створення простого зображення із візуалізацією результатів.

### 1. Автоматизоване опрацювання інформації.

## Info page

### 1. Отримання близько 2 Мб даних, їх опрацювання та завантаження у БД:

Table 'collection' created successfully

Successfully inserted 8037 records into the 'collection' table

Основний фрагмент коду, призначений для виконання наведеного вище на фото завдання:

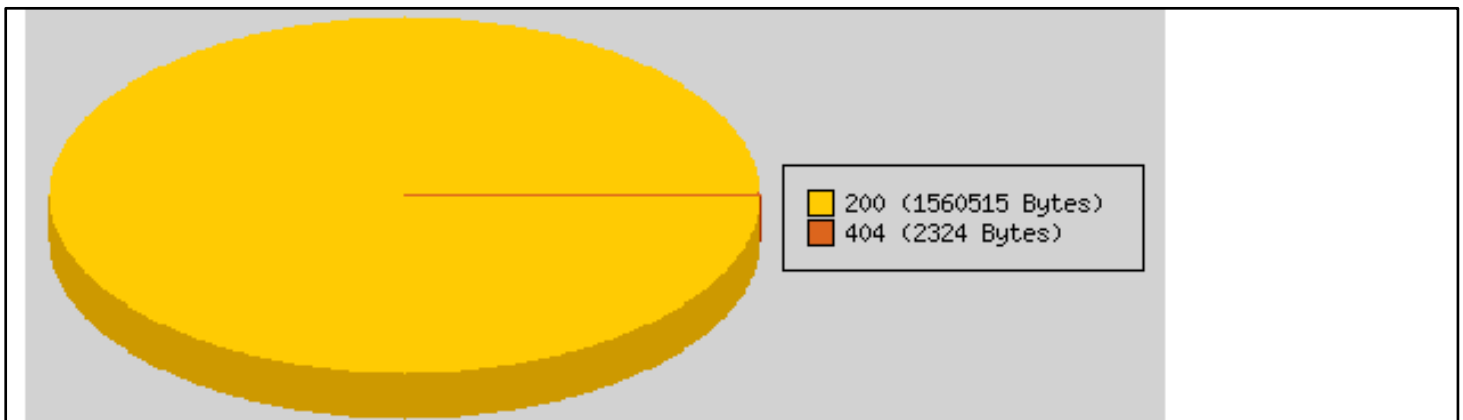
```
foreach ($arr as $k => $line) {
    $line = str_replace("'", "\'", $line); // Escape apostrophes
    preg_match('/^([\d\.]+) \S+ \S+ \[.*\] "(.*)".*$/ ', $line, $matches); // Extract IP addresses and log data

    if (count($matches) == 3) {
        $ip = $matches[1];
        $log = $matches[2];
        $sqlstr = "INSERT INTO collection (ID, IP, Log) VALUES ('" . $id . "', '" . $ip . "', '" . $log . "')";
        if ($mysqli->query($sqlstr)) {
            $successCount++;
        }
        $id++;
    }
}
```

## 2. Нижче наведена більше детальна інформація про IP-адресу '46.125.249.79':

[status] => success  
[country] => Austria  
[countryCode] => AT  
[region] => 4  
[regionName] => Upper Austria  
[city] => Linz  
[zip] => 4020  
[lat] => 48.3074  
[lon] => 14.285  
[timezone] => Europe/Vienna  
[isp] => T-Mobile Austria  
[org] => T-Mobile Austria GmbH  
[as] => AS8412 T-Mobile Austria GmbH  
[query] => 46.125.249.79

## 3. Продемонструємо діаграму розподілу всього скачаного від IP-адреси '46.125.249.79':



Фрагмент коду, призначений для відображеного вище на фото міститься в обох скриптах:

```
$RA = "46.125.249.79"; // IP

$ch = curl_init("http://ip-api.com/json/" . $RA);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false);
curl_setopt($ch, CURLOPT_HEADER, false);
$res = curl_exec($ch);

if ($res === false) {
    echo "cURL Error: " . curl_error($ch) . "<br>";
} else {
    $res = json_decode($res, true);

    echo "<b>2. Нижче наведена більше детальна інформація про IP-адресу '$RA':</b><br>";
    // Print each key-value pair on a new line
    foreach ($res as $key => $value) {
        echo "[$key] => $value<br>";
    }
    echo "<br>";
}
curl_close($ch);
```

## 2. Программний код, який був використаний.

- **main.php** ↓

```
<?php
// Remove the restrictions
set_time_limit(0); // No time limits (realistically 600 seconds)
ini_set("memory_limit", "128M");

echo "<html>\n";
echo "<body>\n";

echo "<h1>Info page</h1>\n";

// Fetch the content from the URL
$homepage = file_get_contents(
    "http://www.almhuetten-raith.at/apache-log/access.log",
    false,
    null,
    0,
    2000000
);

// Divide the content into lines
$arr = explode("\n", $homepage);

// Drop the last line
array_pop($arr);

// Drop the first line
array_shift($arr);

// Connect to the database
$servername = "localhost"; // Server address
$username = "root"; // Database server username
$password = ""; // Database server password
$dbname = "autoprocessing"; // Database name

// Create a database connection
$mysqli = new mysqli($servername, $username, $password, $dbname);

// Check the connection
if ($mysqli->connect_error) {
    die("Connection failed: " . $mysqli->connect_error);
}

// SQL query to create the 'collection' table
$sqlCreateTable = "CREATE TABLE IF NOT EXISTS collection (
                    ID INT PRIMARY KEY,
                    IP VARCHAR(15),
                    Log TEXT
                );";

// Execute the query
if ($mysqli->query($sqlCreateTable) === TRUE) {
    echo "<b>1. Отримання близько 2 Мб даних, їх опрацювання та завантаження у БД:</b><br>";
    echo "Table 'collection' created successfully<br>";
} else {
    echo "Error creating table: " . $mysqli->error . "<br>";
}

// Counter for ID
$id = 0;

// Counter for successful inserts
$successCount = 0;
```

```

foreach ($arr as $k => $line) {
    $line = str_replace("'", "\'", $line); // Escape apostrophes
    preg_match('/^([\d\.]+) \S+ \S+ \[.*\] "(.*)".*$/ ', $line, $matches); // Extract IP
    addresses and log data

    if (count($matches) == 3) {
        $ip = $matches[1];
        $log = $matches[2];
        $sqlstr = "INSERT INTO collection (ID, IP, Log) VALUES ('" . $id . "', '" . $ip .
        "' , '" . $log . "')";
        if ($mysqli->query($sqlstr)) {
            $successCount++;
        }
        $id++;
    }
}

// Output the success message
echo "Successfully inserted " . $successCount . " records into the 'collection'
table<br><br>";

// Close the database connection
$mysqli->close();

$RA = "46.125.249.79"; // IP

$ch = curl_init("http://ip-api.com/json/" . $RA);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false);
curl_setopt($ch, CURLOPT_HEADER, false);
$res = curl_exec($ch);

if ($res === false) {
    echo "cURL Error: " . curl_error($ch) . "<br>";
} else {
    $res = json_decode($res, true);

    echo "<b>2. Нижче наведена більше детальна інформація про IP-адресу '$RA':</b><br>";
    // Print each key-value pair on a new line
    foreach ($res as $key => $value) {
        echo "[$key] => $value<br>";
    }
    echo "<br>";
}
curl_close($ch);

echo "<b>3. Продемонструємо діаграму розподілу всього скачаного від IP-адреси
'$RA':</b><br>";
echo "<img src='./diagram.php'>";

echo "</body>\n";
echo "</html>\n";
?>

```

- **diagram.php** ↓

```

<?php
set_time_limit(0); //no limits (really 600sec)
ini_set("memory_limit", "128M");
error_reporting(E_ERROR);

// $im - идентификатор изображения
// $VALUES - массив со значениями
// $LEGEND - массив с подписями
function Diagramm($im, $VALUES, $LEGEND)
{

```

```

global $COLORS, $SHADOWS;

$black = ImageColorAllocate($im, 0, 0, 0);

// Получим размеры изображения
$W = ImageSX($im);
$H = ImageSY($im);

// Вывод легенды #####

// Посчитаем количество пунктов, от этого зависит высота легенды
$legend_count = count($LEGEND);

// Посчитаем максимальную длину пункта, от этого зависит ширина легенды
$max_length = 0;
foreach ($LEGEND as $v) {
    if ($max_length < strlen($v)) {
        $max_length = strlen($v);
    }
}

// Номер шрифта, которым мы будем выводить легенду
$FONT = 2;
$font_w = ImageFontWidth($FONT);
$font_h = ImageFontHeight($FONT);

// Вывод прямоугольника - границы легенды -----

$l_width = $font_w * $max_length + $font_h + 10 + 5 + 10;
$l_height = $font_h * $legend_count + 10 + 10;

// Получим координаты верхнего левого угла прямоугольника - границы легенды
$l_x1 = $W - 10 - $l_width;
$l_y1 = ($H - $l_height) / 2;

// Выводя прямоугольника - границы легенды
ImageRectangle(
    $im,
    $l_x1,
    $l_y1,
    $l_x1 + $l_width,
    $l_y1 + $l_height,
    $black
);

// Вывод текст легенды и цветных квадратиков
$text_x = $l_x1 + 10 + 5 + $font_h;
$square_x = $l_x1 + 10;
$y = $l_y1 + 10;

$i = 0;
foreach ($LEGEND as $v) {
    $dy = $y + $i * $font_h;
    ImageString($im, $FONT, $text_x, $dy, $v, $black);
    ImageFilledRectangle(
        $im,
        $square_x + 1,
        $dy + 1,
        $square_x + $font_h - 1,
        $dy + $font_h - 1,
        $COLORS[$i]
    );
    ImageRectangle(
        $im,
        $square_x + 1,
        $dy + 1,
        $square_x + $font_h - 1,
        $dy + $font_h - 1,
        $black
    );
}

```

```

        $i++;
    }

    // Вывод круговой диаграммы -----

    $total = array_sum($VALUES);
    $anglesum = $angle = [0];
    $i = 1;

    // Расчет углов
    while ($i < count($VALUES)) {
        $part = $VALUES[$i - 1] / $total;
        $angle[$i] = floor($part * 360);
        $anglesum[$i] = array_sum($angle);
        $i++;
    }
    $anglesum[] = $anglesum[0];

    // Расчет диаметра
    $diametr = $l_xl - 10 - 10;

    // Расчет координат центра эллипса
    $circle_x = $diametr / 2 + 10;
    $circle_y = $H / 2 - 10;

    // Поправка диаметра, если эллипс не помещается по высоте
    if ($diametr > $H * 2 - 10 - 10) {
        $diametr = $H * 2 - 20 - 20 - 40;
    }

    // Вывод тени
    for ($j = 20; $j > 0; $j--) {
        for ($i = 0; $i < count($anglesum) - 1; $i++) {
            ImageFilledArc(
                $im,
                $circle_x,
                $circle_y + $j,
                $diametr,
                $diametr / 2,
                $anglesum[$i],
                $anglesum[$i + 1],
                $SHADOWS[$i],
                IMG_ARC_PIE
            );
        };
    }

    // Вывод круговой диаграммы
    for ($i = 0; $i < count($anglesum) - 1; $i++) {
        ImageFilledArc(
            $im,
            $circle_x,
            $circle_y,
            $diametr,
            $diametr / 2,
            $anglesum[$i],
            $anglesum[$i + 1],
            $COLORS[$i],
            IMG_ARC_PIE
        );
    }
}

// Создадим изображения
header("Content-Type: image/png");
$im = ImageCreate(500, 500);

// Зададим цвет фона. Немного желтоватый, для того, чтобы было
// видно границы изображения на белом фоне.
// $bgcolor=ImageColorAllocate($im,255,255,200); //чорный

```

```

$bgcolor = ImageColorAllocate($im, 210, 210, 210); //світлосірий

// Connect to the database
$servername = "localhost"; // Server address
$username = "root"; // Database server username
$password = ""; // Database server password
$dbname = "autoprocessing"; // Database name

// Create connection
$mysqli = new mysqli($servername, $username, $password, $dbname);

$sqlstr = "SELECT
            SUBSTRING_INDEX(SUBSTRING_INDEX(Log, 'HTTP/1.1\" ' , -1), ' ' , 1) AS
HTTPStatusCode,
            SUM(SUBSTRING_INDEX(SUBSTRING_INDEX(SUBSTRING_INDEX(Log, 'HTTP/1.1\" ' , -
1), ' ' , 2), ' ' , -1)) AS DataSize
        FROM
            collection
        WHERE
            IP = '46.125.249.79'
        GROUP BY
            HTTPStatusCode
        ORDER BY
            DataSize DESC;";
//echo $sqlstr."<br>"; //for test
$result = $mysqli->query($sqlstr);

$VALUES = [];
$LEGEND = [];

for ($i = 0; $i < 2; $i++) { //firtst 3 rows
    $row = mysqli_fetch_array($result, MYSQLI_NUM);

    $VALUES[] = $row[1];
    $LEGEND[] = $row[0] . " " . "(" . $row[1] . " Bytes) ";
}

/* close connection */
$mysqli->close();

// Зададим цвета элементов
$COLORS[0] = imagecolorallocate($im, 255, 203, 3);
$COLORS[1] = imagecolorallocate($im, 220, 101, 29);
$COLORS[2] = imagecolorallocate($im, 189, 24, 51);
$COLORS[3] = imagecolorallocate($im, 214, 0, 127);
$COLORS[4] = imagecolorallocate($im, 98, 1, 96);

// Зададим цвета теней элементов
$SHADOWS[0] = imagecolorallocate($im, 205, 153, 0);
$SHADOWS[1] = imagecolorallocate($im, 170, 51, 0);
$SHADOWS[2] = imagecolorallocate($im, 139, 0, 1);
$SHADOWS[3] = imagecolorallocate($im, 164, 0, 77);
$SHADOWS[4] = imagecolorallocate($im, 48, 0, 46);

// Вызов функции рисования диаграммы
Diagramm($im, $VALUES, $LEGEND);

// Генерация изображения
ImagePNG($im);
ImagePNG($im, "./pie.png"); //save to file
imagedestroy($im);
?>

```



## Контрольні питання

1. Як можна отримувати дані з бази даних за допомогою SQL-запиту і відображати їх на сторінці за допомогою **PHP** та **GD**, або у тих мовах і засобах, які ви використали?

Щоб отримати дані з бази даних за допомогою SQL-запиту та відображати їх на сторінці за допомогою PHP та бібліотеки GD (яка використовується для маніпулювання зображеннями), необхідно виконати наступні кроки (те саме, що й в самій лаб. роботі):

- Виконати SQL-запит до бази даних
- Створити зображення, використовуючи GD

## Висновки

Під час виконання лабораторної роботи було проведено обробку лог-файлів за допомогою PHP та MySQL. Для вилучення та обробки інформації з файлу було використано функції PHP, такі як “file\_get\_contents” для отримання вмісту та “explode” для розбиття на рядки. Дані були занесені до бази даних MySQL, а використання SQL-запитів дозволило ефективно обробити та вибрати необхідні дані.

Створено таблицю “**collection**” для зберігання інформації з лог-файлу, при цьому враховано унікальність IP-адрес та виведено повідомлення про те, чи дані вже існують у таблиці.

Також, використовуючи бібліотеку GD, було створено просте зображення для візуалізації результатів. Усі кроки були коректно оброблені, враховуючи можливі помилки та винятки.

У цій роботі було ефективно поєднано можливості PHP для обробки даних та роботи з базою даних із здатністю GD для генерації зображення, що дозволило успішно виконати завдання та візуалізувати результати.