|  |  |  |
| --- | --- | --- |
|  |  |  |
| МИНОБРНАУКИ РОССИИ | | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | |

Институт Информационных технологий

Кафедра Инструментального и прикладного программного обеспечения

**ПРАКТИЧЕСКАЯ РАБОТА №6**

по дисциплине «Разработка серверных частей интернет-ресурсов»

**Студент группы** ИКБО-20-19 Анваржонов Ж Т

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись студента)

**Руководитель практической работы** преподаватель Волков М.Ю.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись руководителя)

Работа представлена «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_ 2021 г.

Допущен к работе «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2021 г.

Москва 2021

**Оглавление**

[Цель работы 3](#_Toc85888177)

[Ход работы 3](#_Toc85888178)

[Вывод 9](#_Toc85888179)

[Ответы на вопросы к практической работе 9](#_Toc85888180)

[Список использованной литературы 17](#_Toc85888181)

# Цель работы

Используя серверную конфигурацию из предыдущих работ, научиться работать различными расширениями и сторонними модулями PHP.

# Задание

1. Сгенерировать набор фикстур с помощью выбранного стороннего модуля. Каждая фикстура должна содержать не менее 5 полей. Всего фикстур должно быть не менее 50.
2. На основе этих фикстур построить минимум 3 графика различного типа с помощью сторонних модулей PHP.
3. График превратить в картинку и добавить на него ваш личный полупрозрачный водяной знак с помощью библиотеки GD(строго рекомендуется) или другой библиотеки для работы с изображениями в PHP.
4. Отобразить полученные графики на странице статистики.

# Ход работы

1. На основе серверной конфигурации, разработанной в первой работе, были добавлены настройки и код для работы с созданием фикстур и графиков (Листинг 1 – 11).

Листинг 1 – изменения в сервисе web в docker-compose.yml

 web:

    build:

      context: .

      dockerfile: php.Dockerfile

    volumes:

      - ./src/php:/var/www/html/php

      - ./src/static/uploads:/var/www/uploads/

    restart: always

Листинг 2 – php.Dockerfile

FROM php:7.4-apache

RUN php -r "readfile('http://getcomposer.org/installer');" | php -- --install-dir=/usr/local/bin/ --filename=composer

RUN alias composer='php /usr/local/bin/composer'

RUN apt-get update && \

    pecl uninstall redis && \

    pecl install -f redis && \

    docker-php-ext-enable redis && \

    docker-php-ext-install mysqli

RUN apt-get install -y zip

RUN composer require --prefer-dist nelmio/alice && \

    composer update

RUN apt-get update && \

    apt-get install -y libfreetype6-dev libjpeg62-turbo-dev libpng-dev && \

    docker-php-ext-configure gd --with-freetype=/usr/include/ --with-jpeg=/usr/include/ && \

    docker-php-ext-install gd

RUN composer require szymach/c-pchart

Листинг 3 – index.php в graphics

<?php

    include '../login.php';

?>

<!DOCTYPE html>

<html lang="ru">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Графики</title>

</head>

<body>

    <h1>Графики</h1>

    <div>

        <a href="/admin/index.php">административная панель</a>

    </div>

    <div>

        <img src="/admin/graphics/graphicHumadityAndTemperature.php"/>

    </div>

    <div>

        <img src="/admin/graphics/graphicilluminace.php"/>

        <img src="/admin/graphics/graphicSound.php"/>

    </div>

    <div>

        <img src="/admin/graphics/graphicVoltage.php"/>

    </div>

</body>

</html>

Листинг 4 – graphicVoltage.php

<?php

include '../login.php';

include './getFixtures.php';

require '../../../vendor/autoload.php';

use CpChart\Chart\Pie;

use CpChart\Data;

use CpChart\Image;

include './drawImage.php';

$points = [];

$labels = [];

$uniqueVoltage = [];

foreach ($fixtures->getObjects() as $fixture) {

    $voltage = floor($fixture->voltage);

    $index = array\_search($voltage, $uniqueVoltage);

    if ($index !== false) {

        $points[$index]++;

    } else {

        array\_push($points, 1);

        array\_push($labels, $voltage . " - " . $voltage . ".99");

        array\_push($uniqueVoltage, $voltage);

    }

}

$serie\_abcissa = "serie\_abcissa";

$serie\_voltage = "serie\_voltage";

// Create and populate data

$data = new Data();

$data->addPoints($points, $serie\_voltage);

// Define the absissa serie

$data->addPoints($labels, $serie\_abcissa);

$data->setAbscissa($serie\_abcissa);

// Create the image

$image = new Image(400, 230, $data);

// Add a border to the picture

$image->drawRectangle(0, 0, 399, 229, ["R" => 0, "G" => 0, "B" => 0]);

// Write the picture title

$image->setFontProperties(["FontName" => "Silkscreen.ttf", "FontSize" => 14]);

$image->drawText(200, 13, "Voltage", ["R" => 0, "G" => 0, "B" => 0, "Align" => TEXT\_ALIGN\_TOPMIDDLE]);

// Set the default font properties

$image->setFontProperties(["FontName" => "Forgotte.ttf", "FontSize" => 10, "R" => 80, "G" => 80, "B" => 80]);

// Create and draw the chart

$pieChart = new Pie($image, $data);

$pieChart->draw2DPie(200, 125, ["DrawLabels" => true, "Border" => true]);

drawImage($image);

Листинг 5 – graphicSound.php

<?php

include '../login.php';

include './getFixtures.php';

include './drawNormGraphic.php';

$norm = 61;

$points = [0, 0]; // 0 - <=$norm; 1 - >$norm

foreach ($fixtures->getObjects() as $fixture) {

    $points[$fixture->sound > $norm ? 1 : 0]++;

}

drawNormGraphic($norm, $points, "Sound", "Sound norm", 0, max($points) + 1);

Листинг 6 – graphicilluminace.php

<?php

include '../login.php';

include './getFixtures.php';

include './drawNormGraphic.php';

$norm = 361;

$points = [0, 0]; // 0 - <=$norm; 1 - >$norm

foreach ($fixtures->getObjects() as $fixture) {

    $points[$fixture->illuminace > $norm ? 1 : 0]++;

}

drawNormGraphic($norm, $points, "Illuminace", "Illuminance norm", 0, max($points) + 1);

Листинг 7 – graphicHumadityAndTemperature.php

<?php

include '../login.php';

include './getFixtures.php';

require '../../../vendor/autoload.php';

use CpChart\Data;

use CpChart\Image;

include './drawImage.php';

$data = new Data();

$serie\_temp = "temperature";

$serie\_humidity = "humidity";

$serie\_minutes = "minutes";

$minute = 0;

foreach ($fixtures->getObjects() as $fixture) {

    $data->addPoints($fixture->temperature, $serie\_temp);

    $data->addPoints($fixture->humidity, $serie\_humidity);

    $data->addPoints($minute, $serie\_minutes);

    $minute++;

}

$data->setAbscissa($serie\_minutes);

$data->setAbscissaName("Time in minutes");

//SERIE\_SHAPE\_FILLEDTRIANGLE - triangle

//SERIE\_SHAPE\_FILLEDSQUARE - square

$data->setSerieShape($serie\_temp, SERIE\_SHAPE\_FILLEDTRIANGLE);

$data->setSerieShape($serie\_humidity, SERIE\_SHAPE\_FILLEDSQUARE);

$data->setSerieOnAxis($serie\_temp, 0);

$data->setSerieOnAxis($serie\_humidity, 1);

$data->setAxisName(0, "Temperatures");

$data->setAxisName(1, "Humadity");

$data->setAxisPosition(1, AXIS\_POSITION\_RIGHT);

/\* Create the Image object \*/

$image = new Image(900, 300, $data);

/\* Turn off Antialiasing \*/

$image->Antialias = false;

/\* Add a border to the picture \*/

$image->drawRectangle(0, 0, 899, 299, ["R" => 0, "G" => 0, "B" => 0]);

/\* Write the chart title \*/

$image->setFontProperties(["FontName" => "Forgotte.ttf", "FontSize" => 11]);

$image->drawText(200, 35, "Change humadity and temperature", ["FontSize" => 20, "Align" => TEXT\_ALIGN\_BOTTOMMIDDLE]);

/\* Set the default font \*/

$image->setFontProperties(["FontName" => "pf\_arma\_five.ttf", "FontSize" => 6]);

/\* Define the chart area \*/

$image->setGraphArea(60, 40, 850, 250);

/\* Draw the scale \*/

$scaleSettings = [

    "XMargin" => 10,

    "YMargin" => 10,

    "Floating" => true,

    "GridR" => 200,

    "GridG" => 200,

    "GridB" => 200,

    "DrawSubTicks" => true,

    "CycleBackground" => true

];

$image->drawScale($scaleSettings);

/\* Turn on Antialiasing \*/

$image->Antialias = true;

$image->setShadow(true, ["X" => 1, "Y" => 1, "R" => 0, "G" => 0, "B" => 0, "Alpha" => 10]);

/\* Draw the line chart \*/

$image->drawPlotChart();

$image->drawLineChart();

/\* Write the chart legend \*/

$image->drawLegend(680, 20, ["Style" => LEGEND\_NOBORDER, "Mode" => LEGEND\_HORIZONTAL]);

drawImage($image);

Листинг 8 – getFixtures.php

<?php

include '../login.php';

require '../../../vendor/autoload.php';

use Nelmio\Alice\Loader\NativeLoader;

// данные для ящичка интернет-вещей

class Dump {

    public $illuminace;

    public $voltage;

    public $temperature;

    public $sound;

    public $humidity;

}

$loader = new Nelmio\Alice\Loader\NativeLoader();

$fixtures = $loader->loadFile('./fixtures.yml');

?>

Листинг 9 – fixtures.yml

Dump:

    dump{0..59}:

      illuminace: '<randomFloat(1, 360.5, 361.56)>'

      voltage: '<randomFloat(2, 28.01, 31.99)>'

      temperature: '<numberBetween(10, 25)>'

      sound: '<randomFloat(1, 60.5, 61.08)>'

      humidity: '<randomFloat(2, 30.01, 30.99)>'

Листинг 10 – drawNormGraphic.php

<?php

include '../login.php';

require '../../../vendor/autoload.php';

use CpChart\Data;

use CpChart\Image;

include './drawImage.php';

function drawNormGraphic($norm, $points, $axisName, $title, $min, $max)

{

    $data = new Data();

    $serie\_norm = "serie\_norm";

    $serie\_illuminace = "serie\_illuminace";

    $data->addPoints($points, $serie\_illuminace);

    $data->setAxisName(0, $axisName);

    $data->addPoints(["<=" . $norm, ">" . $norm], $serie\_norm);

    $data->setAbscissa($serie\_norm);

    /\* Create the Image object \*/

    $image = new Image(500, 500, $data);

    $image->setFontProperties(["FontName" => "pf\_arma\_five.ttf", "FontSize" => 6]);

    /\* Draw the chart scale \*/

    $image->setGraphArea(30, 80, 480, 480);

    $image->drawText(250, 55, $title, array("FontSize" => 20, "Align" => TEXT\_ALIGN\_BOTTOMMIDDLE));

    /\* Turn on shadow computing \*/

    $image->setShadow(true, ["X" => 1, "Y" => 1, "R" => 0, "G" => 0, "B" => 0, "Alpha" => 10]);

    /\* Draw the chart \*/

    $image->drawScale([

        "Mode" => SCALE\_MODE\_MANUAL,

        "ManualScale" => [

            0 => [

                "Min" => $min,

                "Max" => $max,

            ],

        ],

    ]);

    $image->drawBarChart([

        "DisplayValues" => true,

        "Rounded" => true,

        "Surrounding" => 30,

    ]);

    drawImage($image);

}

Листинг 11 – drawImage.yml

<?php

    function drawImage($image) {

        $filename1 = "example.createFunctionSerie.scatter.png";

        $filename\_whater\_mark = "whater\_mark.png";

        $image->render($filename1);

        $stamp = imagecreatefrompng($filename\_whater\_mark);

        $im = imagecreatefrompng($filename1);

        $marge\_right = 10;

        $marge\_bottom = 10;

        $sx = imagesx($stamp);

        $sy = imagesy($stamp);

        imagecopy($im, $stamp, imagesx($im) - $sx - $marge\_right, imagesy($im) - $sy - $marge\_bottom, 0, 0, imagesx($stamp), imagesy($stamp));

        header('Content-type: image/png');

        imagepng($im);

        imagedestroy($im);

        unlink($filename1);

    }

1. Программа была запущена и протестирована (Рисунок 1 - 3).

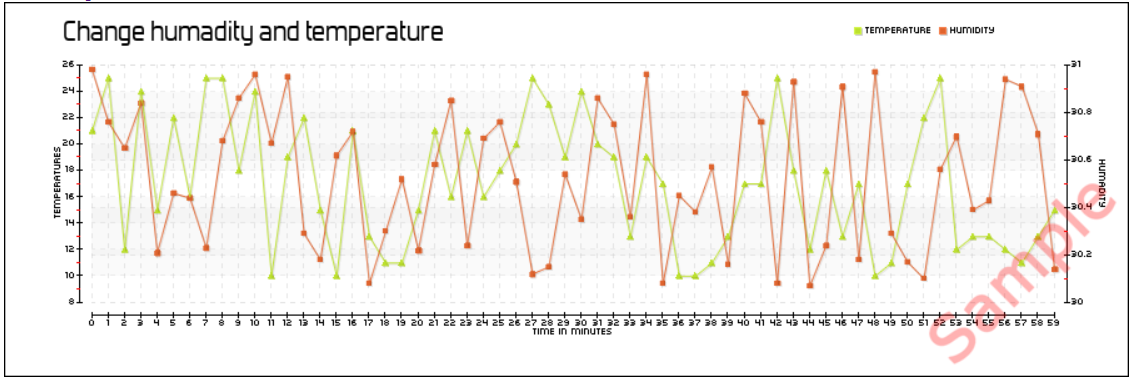


Рисунок 1 – линейная диаграмма

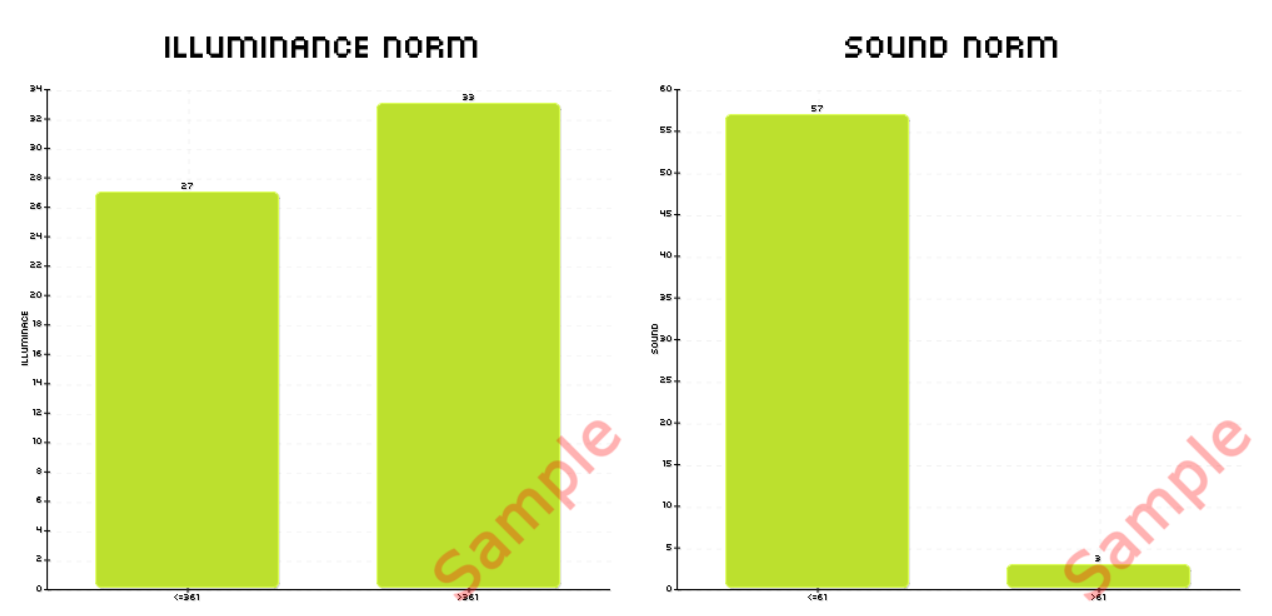


Рисунок 2 – столбчатая диаграмма

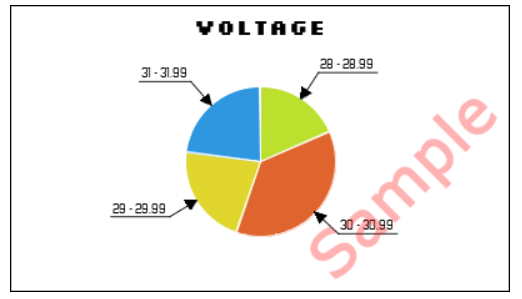


Рисунок 3 – круговая диаграмма

# Вывод

В результате работы была изучена работа с различными расширениями и сторонними модулями PHP.

# Ответы на вопросы к практической работе

# 

# Список использованной литературы

1. Документация Docker — Текст : электронный // Docker docs : [Электронный ресурс]. — URL: https://docs.docker.com/ (дата обращения: 04.09.2021).
2. Руководство по php — Текст : электронный // php : [Электронный ресурс]. — URL: https://www.php.net/ (дата обращения: 22.09.2021).
3. Docker Hub — Текст : электронный // Docker Hub : [Электронный ресурс]. — URL: https://hub.docker.com/ (дата обращения: 04.09.2021).
4. Справочник по php — Текст : электронный // Docker Hub : [Электронный ресурс]. — URL: https://php.ru/manual/ (дата обращения: 22.09.2021).