

## Лабораторная работа №3

студента группы ИТ-212

Устинова Андрея Олеговича

### Скриптовый язык программирования PHP

Выполнение \_\_\_\_\_

Защита \_\_\_\_\_

Цель работы: ознакомиться с основными возможностями языка PHP, синтаксисом, основными операциями, получить практические навыки программирования.

#### Содержание работы

1. Установить программу Denwer (или любой другой локальный веб-сервер), включающую в себя web-сервер Apache, язык программирования PHP и СУБД MySQL.
2. Переместить сайт, разработанный в лабораторной работе №2, в директорию /WebServers/home/имя\_сайта/www/ и перезапустить Denwer. Изменить сайт, используя возможности языка программирования PHP.
3. Проанализировать сайт и вынести общие элементы web-страниц, такие как шапка (header), подвал (footer), меню и т.д. в отдельные файлы, подключив их к основным скриптам функциями require, require\_once, include, include\_once. При этом, такое содержимое шапки, как заголовок страницы (тег <title>), тег <meta> и др. на разных страницах должно отличаться, т.е. соответствовать назначению текущей страницы.
4. Средствами языка PHP произвести проверку данных, введенных в поля формы (на заполнение, на правильность email, на ввод только числового значения и т.д. при помощи регулярных выражений). В случае ошибок вывести их, выделить неправильно заполненные поля (бэграундом, цветом, картинкой) и заполнить поля теми же значениями, которые там были до отправки формы. Использовать метод POST.
5. Сделать страничку на PHP для подсчета размера, занимаемого на жестком диске любой папкой или файлом вашего сайта (можно разместить на about.php). Наименование папки/файла должно передаваться методом GET в виде относительного пути. *Предусмотреть ограничения на ввод, учитывая, что скрипту должна быть доступна только корневая папка сайта и вложенные.*
6. Реализовать одно из следующих заданий, разработав класс для обработки данных:
  - *отправка email'а через форму обратной связи с полями «Представьтесь», «Ваш E-mail», «Текст сообщения». Использовать несколько дополнительных заголовков и кодировку UTF-8. Данные полей должны проверяться на правильность.*
  - *RSS-поток новостей с возможностью просмотра в RSS-агрегаторе или браузере. Данные для потока читать из файла. Формат хранимых в файле данных разработать самостоятельно.*

- наложение текстов/изображений на другие изображения (водяные знаки), загружаемые с компьютера пользователя, с отображением ссылки на вновь полученный файл.
- рисование диаграмм по хранимым в произвольном файле данным. Вид диаграммы должен выбираться пользователем (график, гистограмма).
- организовать в файле локальную базу данных по определенной предметной области (например база данных аудиотреков, видеофильмов, компьютерных комплектующих и т.д.) с функциями сохранения, чтения и удаления. Список хранимых атрибутов (полей) определить самостоятельно.
- из заданного статичного html-файла со списком новостей (можно сохранить любую html-страницу с новостями) используя регулярные выражения в формате PCRE вывести все новости на web-страницу игнорируя ненужную служебную информацию, такую как строка меню, шапка заголовка, подвал, т.е. все что не является новостями.
- ведение статистики по посетителям сайта (ip-адрес, название и версия браузера, количество посещений, заходов на конкретную страницу, дата/время входа). Данные сохранять в \*.csv файл и представить на PHP-странице в удобном виде с возможностью группировки данных и выводом итоговых значений. Предусмотреть отображение в различных разрезах (по времени, по датам, по браузеру, по ip-адресу) в процентных соотношениях.

7. Страницы должны быть работоспособны в любом браузере.

## Ход работы

1. Установил программу Open Server.
  2. Переместил сайт, разработанный в лабораторной работе №2, в директорию /home/имя\_сайта/ и перезапустил Open Server. Изменил сайт, используя возможности языка программирования PHP.
  3. Проанализировал сайт и вынес общие элементы web-страниц, такие как шапка (header), подвал (footer) в отдельные файлы, подключив их к основным скриптам функциями require, require\_once, include, include\_once. При этом, такое содержимое шапки, как заголовок страницы (тег <title>), тег <meta> и др. на разных страницах должно отличаться, т.е. соответствовать назначению текущей страницы.
  4. Средствами языка PHP произвел проверку данных, введенных в поля формы (на заполнение, на правильность email при помощи регулярных выражений). В случае ошибок они выводятся, выделяются неправильно заполненные поля (обводка) и заполняются поля теми же значениями, которые там были до отправки формы. Использован метод POST.
- Проверка ввода почты и пароля с методом POST:

```
<?php
header('Access-Control-Allow-Origin: *');
header('Access-Control-Allow-Methods: GET, POST, OPTIONS');
header('Access-Control-Allow-Headers: Content-Type');
header('Access-Control-Allow-Credentials: true');
header('Content-Type: application/json');

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

$errors = [];
$email = '';
$password = '';

// Функция для проверки email
function validateEmail($email) {
    return filter_var($email, FILTER_VALIDATE_EMAIL) !== false;
}

// Функция для проверки пароля
function validatePassword($password) {
    return preg_match('/^[a-zA-Z0-9!@#%$^&*()_+{}:;<>.,?^\\-]{8,}$/', $password);
}

try {
    // Проверка предварительного запроса (OPTIONS)
    if ($_SERVER['REQUEST_METHOD'] === 'OPTIONS') {
        http_response_code(200);
        exit();
    }
}
```

```

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $postData = json_decode(file_get_contents('php://input'), true);
    $email = trim($postData['login-email']);
    $password = trim($postData['login-password']);

    // Проверка email
    if (empty($email) || !validateEmail($email)) {
        $errors['login-email'] = 'Введите корректный e-mail.';
    }

    // Проверка пароля
    if (empty($password) || !validatePassword($password)) {
        $errors['login-password'] = 'Пароль должен содержать не менее 8 символов.
Допускается ввод символов латинского алфавита, цифр и спецсимволов';
    }

    // Если есть ошибки
    if (!empty($errors)) {
        echo json_encode(['errors' => $errors]);
        exit();
    }

    // Пример успешного ответа
    echo json_encode(['success' => 'Успешный вход!']);
    exit();
}

} catch (Exception $e) {
    echo json_encode(['error' => $e->getMessage()]);
    exit();
}

```

- Обработка отправки формы ввода

```

document.querySelector(".login form").addEventListener("submit", (event) => {
    event.preventDefault();

    const formData = {
        "login-email": loginEmailInput.value,
        "login-password": loginPasswordInput.value,
    };

    fetch("validate_login.php", {
        method: "POST",
        headers: {
            "Content-Type": "application/json",
            Accept: "application/json",
        },
        body: JSON.stringify(formData),
    })
    .then((response) => response.json())

```

```

.then((data) => {
  if (data.errors) {
    displayErrors(data.errors);
  } else if (data.success) {
    modal.style.display = "none";
    console.log(data.success);
  }
})
.catch((error) => console.error("Error:", error));
});

```

**Войти** или зарегистрироваться

e-mail

Введите корректный e-mail.

Пароль

Пароль должен содержать не менее 8 символов. Допускается ввод символов латинского алфавита, цифр и спецсимволов

Впервые на сайте? Зарегистрируйтесь

**Войти**

Рисунок 1 Ошибки при вводе

5. Сделал страничку на PHP для подсчета размера, занимаемого на жестком диске любой папкой или файлом сайта (разместил на about.php). Наименование папки/файла передается методом GET в виде относительного пути. Предусмотрел ограничение на ввод, учитывая, что скрипту должна быть доступна только корневая папка сайта и вложенные.

```

<?php
$title = "Узнать размер файла";
require_once("header.php");

// Функции для расчета размера
function getDirectorySize($dir) {

```

```

        $size = 0;

        // Открываем каталог и проходим по каждому элементу
        foreach (new RecursiveIteratorIterator(new RecursiveDirectoryIterator($dir)) as
$file) {
            $size += $file->getSize();
        }

        return $size;
    }

    function getFileSize($file) {
        return filesize($file);
    }

    // Получаем параметр из GET-запроса
    $relativePath = isset($_GET['path']) ? $_GET['path'] : '';

    // Определяем корневую папку сайта (на один уровень выше папки pages)
    $rootDir = realpath(__DIR__ . '/../'); // Переход на уровень выше

    // Определяем полный путь
    $fullPath = realpath($rootDir . '/' . $relativePath);

    // Проверяем, что путь находится внутри корневой папки
    if ($fullPath && strpos($fullPath, $rootDir) === 0) {
        if (is_dir($fullPath)) {
            $size = getDirectorySize($fullPath);
            $type = "папки";
        } elseif (is_file($fullPath)) {
            $size = getFileSize($fullPath);
            $type = "файла";
        } else {
            $size = null;
            $type = "указанного пути";
        }
    } else {
        $size = null;
        $type = "недопустимого пути";
    }
?>

<div class="container">
    <p class="title">0 нас</p>
</div>

<div class="container">
    <div>
        <form method="GET">
            <label for="path">Введите относительный путь к файлу или папке:</label>

```

```

        <input type="text" id="path" name="path" placeholder="например,
img/logo.png" required>
        <button type="submit">Посчитать</button>
    </form>

    <?php if ($relativePath): ?>
        <h2>Результат:</h2>
        <?php if ($size !== null): ?>
            <p>Размер <?= htmlspecialchars($type) ?>: <?= number_format($size /
1024, 2) ?> KB</p>
        <?php else: ?>
            <p>Ошибка: <?= htmlspecialchars($type) ?></p>
        <?php endif; ?>
    <?php endif; ?>
</div>
</div>

<?php
    require_once("footer.php");
?>

```

## О нас

Введите относительный путь к файлу или папке:

## Результат:

Размер файла: 8.33 KB

Рисунок 2 Проверка размера файла или папки

6. Реализовал одно из заданий, разработав класс для обработки данных:

- *рисование диаграмм по хранимым в произвольном файле данным. Вид диаграммы должен выбираться пользователем (график, гистограмма).*

Функция для рисования диаграммы:

```

function generateSVG($chartType, $labels, $values, $xAxisLabel, $yAxisLabel) {
    $width = 800; // Ширина SVG
    $height = 500; // Высота SVG
    $barWidth = ($width - 100) / count($values); // Размер столбцов с учётом отступов
    $maxValue = max($values);
    $topPadding = 30; // Отступ сверху
    $leftPadding = 60; // Отступ слева для оси количества
    $bottomPadding = 40; // Отступ снизу для оси оценки
    $yStep = $maxValue / 5; // Шаг для отметок на оси Y (5 отметок)

    // Высота доступного пространства для диаграммы после учёта отступов
    $availableHeight = $height - $topPadding - $bottomPadding;
}

```

```

// Создание SVG
$svg = '<svg width="' . $width . '"' height="' . $height . '"
xmlns="http://www.w3.org/2000/svg">';
$svg .= '<rect width="100%" height="100%" fill="white" />';

// Линия оси Y (Количество)
$svg .= '<line x1="' . $leftPadding . '" y1="' . $topPadding . '" x2="' .
$leftPadding . '" y2="' . ($height - $bottomPadding) . '" stroke="black" stroke-
width="2" />';

// Линия оси X (Оценка)
$svg .= '<line x1="' . $leftPadding . '" y1="' . ($height - $bottomPadding) . '"
x2="' . ($width - 20) . '" y2="' . ($height - $bottomPadding) . '" stroke="black"
stroke-width="2" />';

// Добавление отметок на ось Y
for ($i = 0; $i <= 5; $i++) {
    $yValue = $yStep * $i;
    $yPosition = $height - $bottomPadding - ($yValue / $maxValue) *
$availableHeight;

    // Отмечаем отметки на оси Y
    $svg .= '<line x1="' . ($leftPadding - 5) . '" y1="' . $yPosition . '" x2="' .
$leftPadding . '" y2="' . $yPosition . '" stroke="black" stroke-width="1" />';

    // Добавляем текстовые метки для оси Y
    $svg .= '<text x="' . ($leftPadding - 10) . '" y="' . ($yPosition + 5) . '"
fill="black" text-anchor="end" font-size="12">' . round($yValue, 2) . '</text>';
}

// Отображение гистограммы
if ($chartType === 'bar') {
    foreach ($values as $index => $value) {
        // Применение отступа сверху и уменьшение высоты столбцов
        $barHeight = ($value / $maxValue) * $availableHeight;
        $x = $leftPadding + $index * $barWidth;
        $y = $height - $barHeight - $bottomPadding; // Столбцы рисуются снизу с
учётом отступа сверху

        // Рисуем столбец
        $svg .= '<rect x="' . $x . '" y="' . $y . '" width="' . ($barWidth - 10) .
'" height="' . $barHeight . '" fill="rgba(75, 192, 192, 0.7)" />';

        // Подпись над столбцом с отступом на 5 пикселей выше верхушки столбца
        $svg .= '<text x="' . ($x + ($barWidth - 10) / 2) . '" y="' . ($y - 5) . '"
fill="black" text-anchor="middle">' . htmlspecialchars($value) . '</text>';

        // Подпись под столбцом (оценка)
        $svg .= '<text x="' . ($x + ($barWidth - 10) / 2) . '" y="' . ($height -
$bottomPadding + 20) . '" fill="black" text-anchor="middle">' .
htmlspecialchars($labels[$index]) . '</text>';
    }
}

```



```

    }
    } elseif ($chartType === 'line') {
        $svg .= '<polyline fill="none" stroke="rgba(75, 192, 192, 1)" stroke-width="2"
points="';
        foreach ($values as $index => $value) {
            $x = $leftPadding + $index * ($width - $leftPadding - 20) / (count($values)
- 1);

            $y = $height - (($value / $maxValue) * $availableHeight) - $bottomPadding;
            $svg .= $x . ',' . $y . ' ';
        }
        $svg .= '" />';

        // Добавляем подписи под точками
        foreach ($values as $index => $value) {
            $x = $leftPadding + $index * ($width - $leftPadding - 20) / (count($values)
- 1);

            $svg .= '<text x="' . $x . '" y="' . ($height - $bottomPadding + 20) . '"
fill="black" text-anchor="middle">' . htmlspecialchars($labels[$index]) . '</text>';
        }
    }

    // Добавляем подписи для осей
    // Подпись оси X (Оценка)
    $svg .= '<text x="' . ($width / 2) . '" y="' . ($height - 5) . '" fill="black" text-
anchor="middle" font-size="14">' . htmlspecialchars($xAxisLabel) . '</text>';

    // Подпись оси Y (Количество) - вращаем текст на 90 градусов
    $svg .= '<text x="' . 20 . '" y="' . ($height / 2) . '" fill="black" text-
anchor="middle" font-size="14" transform="rotate(-90, 20, ' . ($height / 2) . ')">' .
htmlspecialchars($yAxisLabel) . '</text>';

    $svg .= '</svg>';

    return $svg;
}

```

Класс для обработки данных:

```

<?php
class ChartData {
    private $labels = [];
    private $values = [];
    private $chartType;
    private $xAxisLabel;
    private $yAxisLabel;

    public function loadFromFile($filePath) {
        if (!is_uploaded_file($filePath) || !file_exists($filePath)) {
            throw new Exception("Ошибка загрузки файла.");
        }

        $data = file($filePath, FILE_IGNORE_NEW_LINES | FILE_SKIP_EMPTY_LINES);
    }
}

```

```

        // Получаем метки для осей из первой строки
        if (isset($data[0])) {
            list($this->xAxisLabel, $this->yAxisLabel) = array_map('trim', explode(',', $data[0]));
        }

        foreach ($data as $index => $line) {
            if ($index === 0) continue; // Пропускаем первую строку (заголовок)

            $parts = array_map('trim', explode(',', $line));

            if (count($parts) >= 2) {
                $label = $parts[0];
                $value = (float)$parts[1];

                if (!empty($label) && is_numeric($value)) {
                    $this->labels[] = $label;
                    $this->values[] = $value;
                } else {
                    throw new Exception("Некорректные данные в строке: " .
                        htmlspecialchars($line));
                }
            }
        }
    }

    public function setChartType($chartType) {
        $this->chartType = htmlspecialchars($chartType, ENT_QUOTES, 'UTF-8');
    }

    public function getChartType() {
        return $this->chartType;
    }

    public function getLabels() {
        return $this->labels;
    }

    public function getValues() {
        return $this->values;
    }

    public function getXAxisLabel() {
        return $this->xAxisLabel;
    }

    public function getYAxisLabel() {
        return $this->yAxisLabel;
    }
}

```

# Отзывы

Загрузите файл с данными:  Не выбран ни один файл

Выберите тип диаграммы:

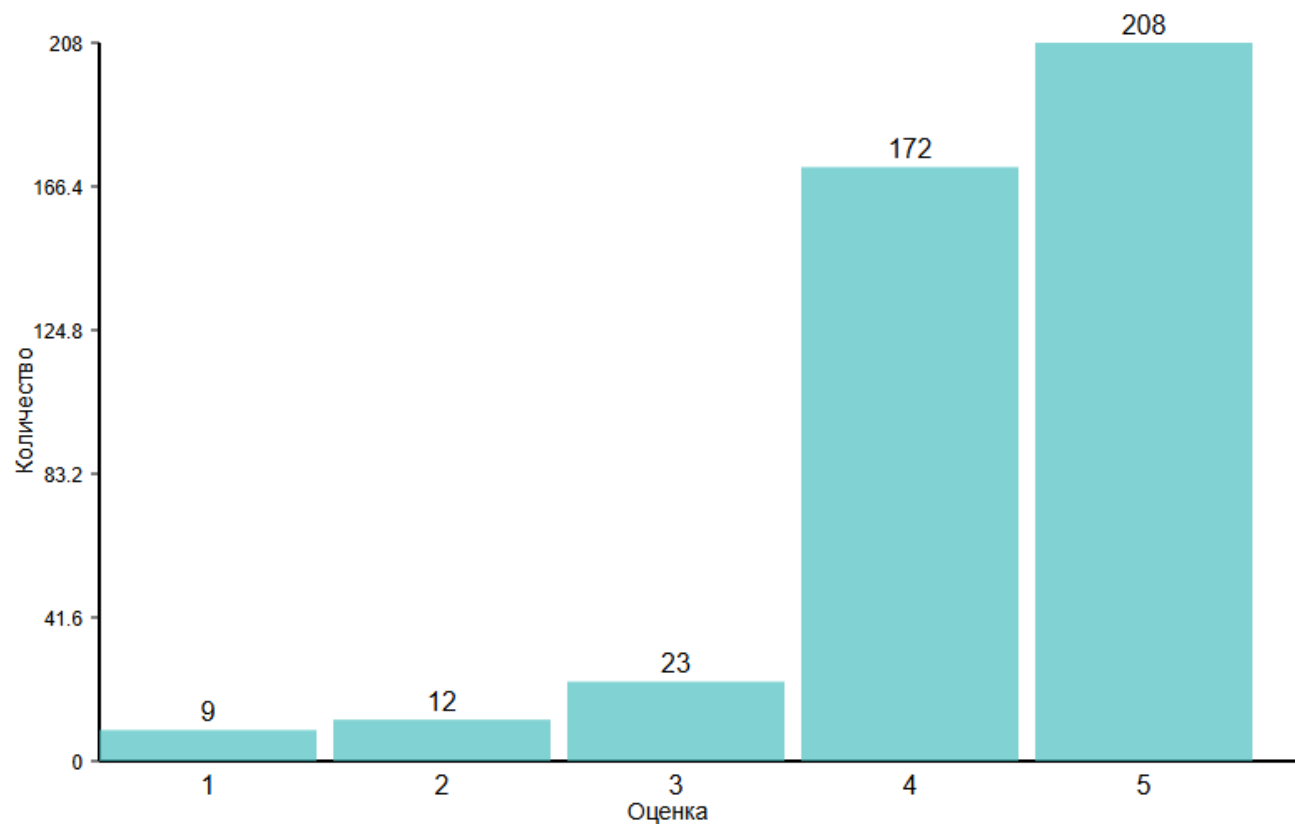


Рисунок 3 Гистограмма

# ОТЗЫВЫ

Загрузите файл с данными:  Не выбран ни один файл

Выберите тип диаграммы:  ▼

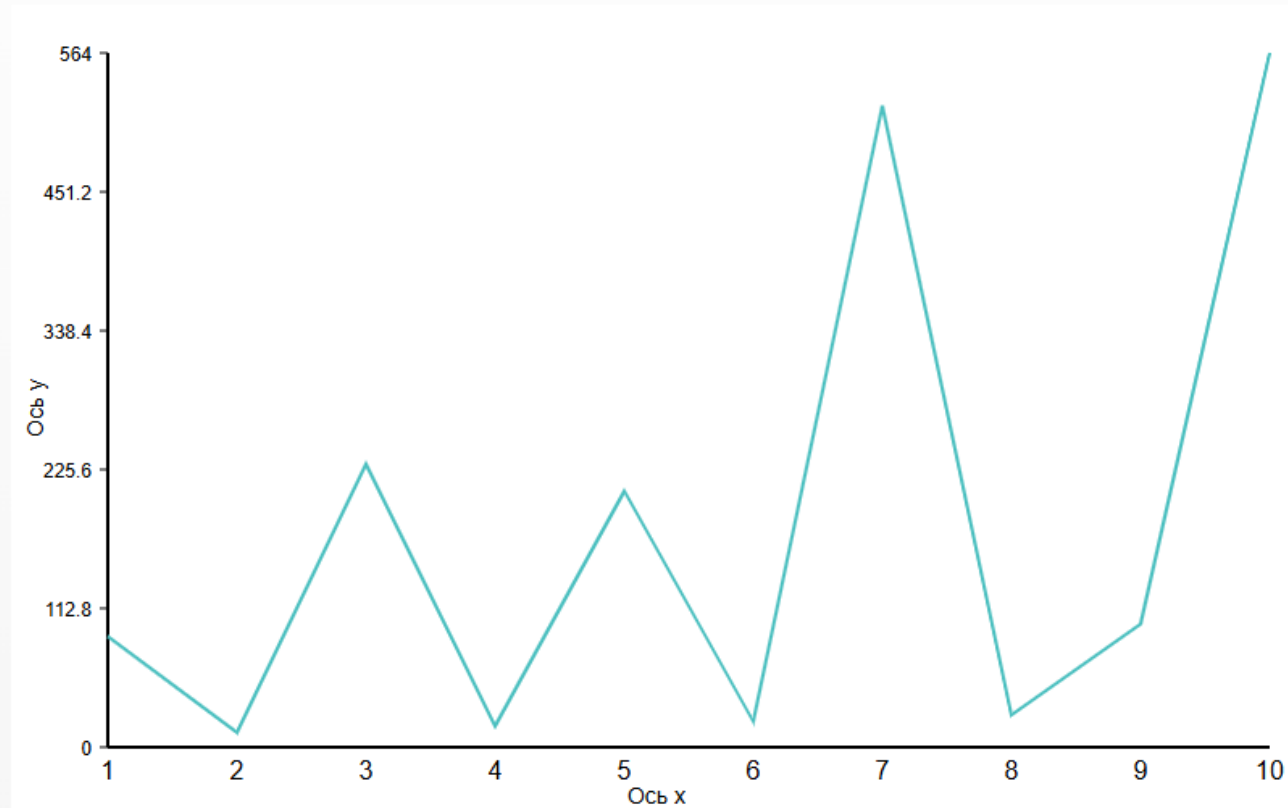


Рисунок 4 График