



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 №2**

по дисциплине «Разработка серверных частей интернет-ресурсов»

**Студент группы ИКБО-30-20**

**Земар Аким Саидович**

---

(подпись студента)

**Руководитель практической работы**

---

(подпись руководителя)

Работа представлена

«\_\_\_» \_\_\_\_\_ 2022 г.

Допущен к работе

«\_\_\_» \_\_\_\_\_ 2022 г.

Москва 2022

## Оглавление

Цель работы .....	3
Ход работы .....	3
Вывод .....	11
Ответы на вопросы к практической работе .....	12
Список использованной литературы .....	15

## Цель работы

Используя серверную конфигурацию, разработанную в прошлой практической работе выполнить следующие упражнения. Предполагается создать 3 независимых сервиса, устойчивых к минимальному набору самых простых ошибок. Предполагается создание 1 общего проекта с разделенными сервисами, разделением проекта на файлы для разделения функционала и переиспользования файлов. Каждый сервис должен состоять как минимум из 2 файлов.

## Ход работы

1. Был создан общий проект с разделенными сервисами, разделением проекта на файлы для разделения функционала. Его структура представлена на рисунке 1.

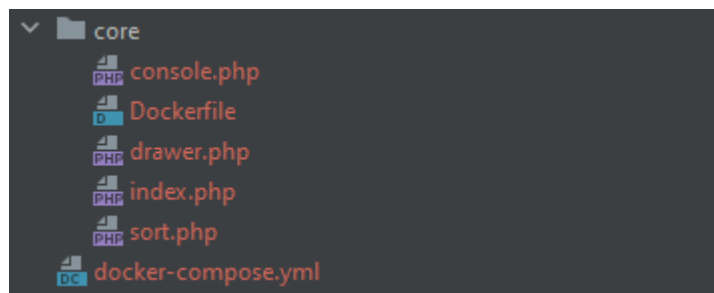


Рисунок 1 - Файловая система проекта

Ниже на рисунках 2 – 7 представлен код программы.

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Console</title>
6   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-EVILKN" crossorigin="anonymous">
7
8 </head>
9 <body>
10 <?php
11   function run_cmd($cmd)
12   {
13     $lines = array();
14     exec($cmd, $output, $lines);
15     echo "<pre>> " . $cmd . "</pre>";
16     echo "<pre>" . implode(separator: "\n", $lines) . "</pre>";
17   }
18
19   if (isset($_GET['cmd'])) {
20     $cmd = $_GET['cmd'];
21     run_cmd($cmd);
22   } else {
23     echo "
24     <div class='d-flex flex-column min-vh-100 justify-content-center align-items-center'>
25       <form action='/console.php' method='GET'>
26         <div class='mb-3'>
27           <label for='cmdInput' class='form-label'>Command</label>
28           <input name='cmd' type='text' class='form-control' id='cmdInput' aria-describedby='cmdHelp'>
29           <div id='cmdHelp' class='form-text'>Input command</div>
30         </div>
31         <button type='submit' class='btn btn-primary'>Execute</button>
32       </form>
33     </div>
34     ";
35   }
36 <?>
37 </body>
38 </html>
39

```

Рисунок 2 – Файл console.php



```

1  <html lang="ru">
2  <head>
3      <title>Main Page</title>
4      <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/css/bootstrap.min.c
5  </head>
6  <body>
7      <div class="d-flex flex-column min-vh-100 justify-content-center align-items-cent
8          <h3>Select function</h3>
9          <div class="row row-cols-auto">
10             <div class="col">
11                 <a href="/console.php" class="btn btn-primary">Console</a>
12             </div>
13             <div class="col">
14                 <a href="/sort.php" class="btn btn-primary">Sort</a>
15             </div>
16             <div class="col">
17                 <a href="/drawer.php" class="btn btn-primary">Drawer</a>
18             </div>
19         </div>
20     </div>
21     <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/js/bootstrap.bundl
22 </body>
23 </html>

```

Рисунок 4 – Файл index.php

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>Quick Sort</title>
6      <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-EVSTQN3/azprG1Anm3QDgp3L/9adGaKb6GE2Ej9lG6633BzR7eXp16V36gKzOwrKqC4>
7  </head>
8  <body>
9      <?php
10     function quick_sort($array)
11     {
12         if (count($array) < 2) {
13             return $array;
14         }
15         $target = $array[0];
16         $less = array();
17         $equal = array($target);
18         $greater = array();
19         for ($i = 1; $i < count($array); $i++) {
20             $elem = $array[$i];
21             if ($elem > $target) {
22                 $greater[] = $elem;
23             } elseif ($elem < $target) {
24                 $less[] = $elem;
25             } else {
26                 $equal[] = $elem;
27             }
28         }
29         $less = quick_sort($less);
30         $greater = quick_sort($greater);
31         return array_merge($less, $equal, $greater);
32     }
33
34     function print_array($array): void
35     {
36         echo "<pre>[";
37         echo implode( separator: ", ", $array);
38         echo "]\</pre>";
39     }
40

```

Рисунок 5 – Файл sort.php (1 часть)

```

if (isset($_GET['array'])) {
    $array = explode( separator: " ", $$_GET["array"]);
    echo "<p>Original array:</p>";
    print_array($array);
    $array = quick_sort($array);
    echo "<p>Sorted array:</p>";
    print_array($array);
}
else {
    echo "
    <div class='d-flex flex-column min-vh-100 justify-content-center align-items-center'>
        <form action='/sort.php' method='GET'>
            <div class='mb-3'>
                <label for='arrayInput' class='form-label'>Array</label>
                <input name='array' type='text' class='form-control' id='arrayInput' aria-describedby='arrayHelp'>
                <div id='arrayHelp' class='form-text'>Input array</div>
            </div>
            <button type='submit' class='btn btn-primary'>Sort array</button>
        </form>
    </div>
    ";
}
?>
</body>
</html>

```

Рисунок 6 – Файл sort.php (2 часть)

```

1    version: '3.9'
2
3    >> services:
4    >   apache:
5       build: ./core
6       volumes:
7         - ./core:/var/www/html/
8       ports:
9         - "8080:80"

```

Рисунок 7 – Файл docker-compose.yml

2. Веб-сервис Drawer для рисования svg объектов. Ему передается параметр с целым числом, первые два бита которого отвечают за тип фигуры,



двадцать четыре – за цвет, три – за ширину и три – за высоту. Результаты работы представлены на рисунках 2-5.

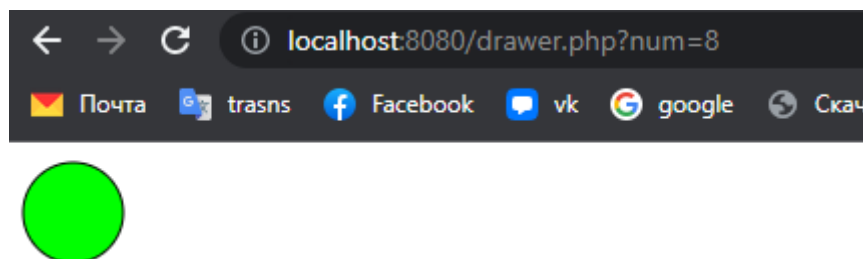


Рисунок 8 – Отображение круга

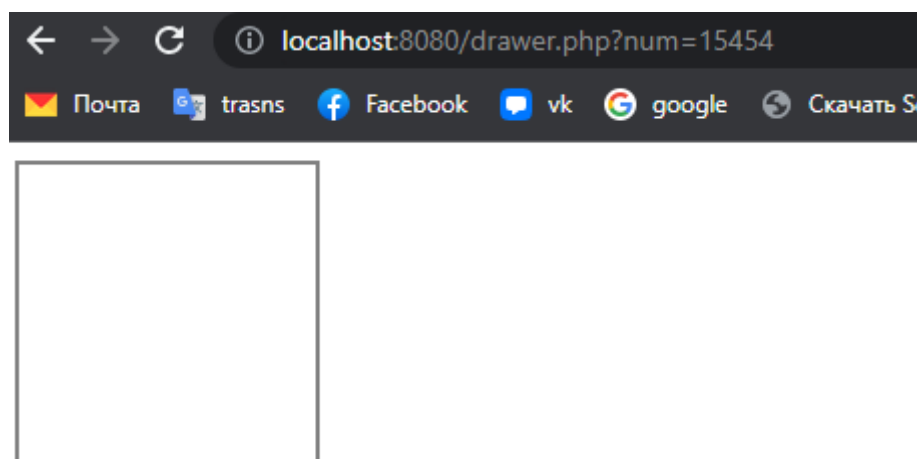


Рисунок 9 – Отображение квадрата

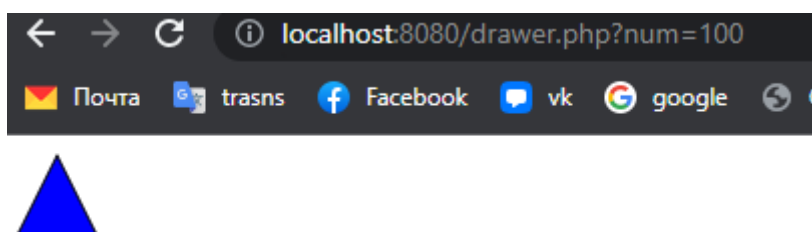
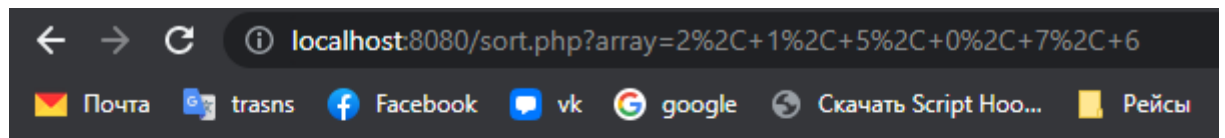


Рисунок 10 – Отображение треугольника

3. Веб-сервис для сортировки массива из целых чисел, принимающий в качестве параметра агау – строку, где числа записаны через запятую. Результаты работы представлены на рисунке 6.



Original array:

[2, 1, 5, 0, 7, 6]

Sorted array:

[0, 1, 2, 5, 6, 7]

Рисунок 11 – Результат сортировки массива

4. Веб-сервис для генерации информационно-административной веб-страницы, принимающий как параметр команду Unix.

Результаты работы представлены на рисунках 7-10.

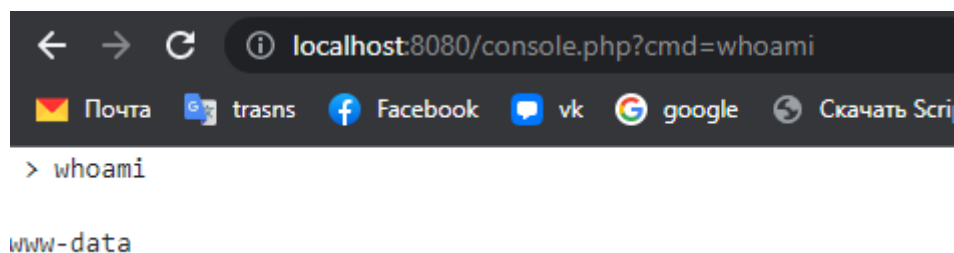
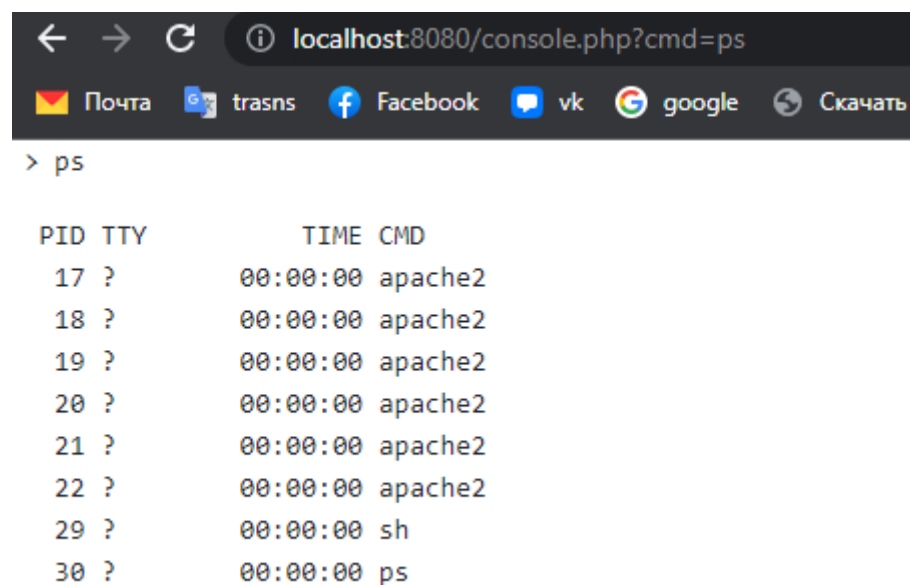
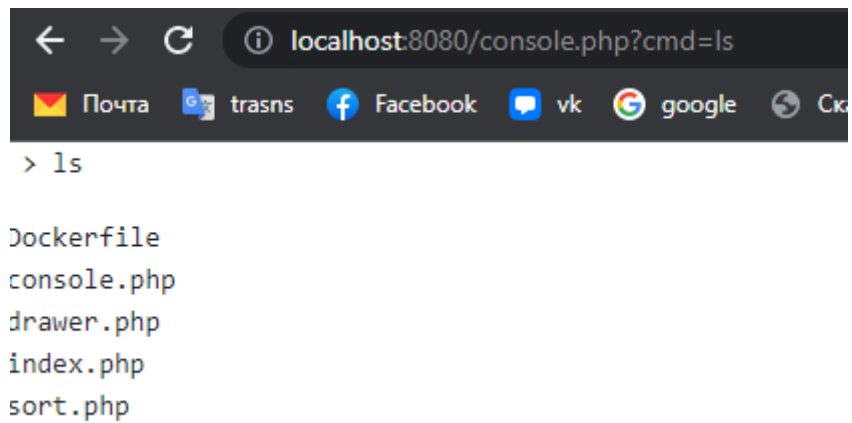


Рисунок 12 - Вывод команды whoami



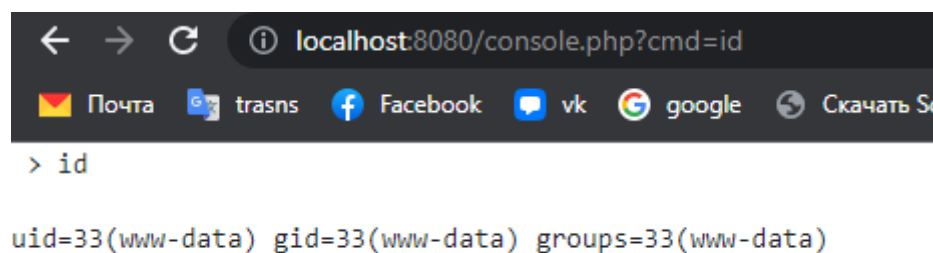
PID	TTY	TIME	CMD
17	?	00:00:00	apache2
18	?	00:00:00	apache2
19	?	00:00:00	apache2
20	?	00:00:00	apache2
21	?	00:00:00	apache2
22	?	00:00:00	apache2
29	?	00:00:00	sh
30	?	00:00:00	ps

Рисунок 13 - Вывод команды ps



```
localhost:8080/console.php?cmd=ls
Почта trasns Facebook vk google
> ls
Dockerfile
console.php
drawer.php
index.php
sort.php
```

Рисунок 14 – Вывод команды ls



```
localhost:8080/console.php?cmd=id
Почта trasns Facebook vk google Скачать So
> id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
```

Рисунок 15 – Вывод команды id

Ссылка на github: [https://github.com/MycroftBrooks/PHP\\_mirea](https://github.com/MycroftBrooks/PHP_mirea)

## Вывод

В работе была продолжена настройка серверной конфигурации, разработанная в прошлой практической работе. В результате было изучено использование разделения проекта на сервисы и файлы. Также были использованы на практике различные возможности языка php для обработки входящих параметров.

## Ответы на вопросы к практической работе

1. Конфигурационный файл `php.ini` - это конфигурационный файл, содержащий настройки интерпретатора PHP. Чтобы интерпретатор PHP находил этот файл его нужно переименовать в `php`.

2. Как написать простой скрипт на `php` – ниже указаны основные инструменты для написания простого скрипта.

3. Основные правила, связанные с переменными в `php`:

Переменная начинается с знака `$`, за которым следует имя переменной

Имя переменной должно начинаться с буквы или символа

подчеркивания

Имя переменной не может начинаться с числа

Имя переменной может содержать только буквенно-цифровые символы и знаки подчеркивания (`a-z`, `0-9` и `_`)

Имена переменных чувствительны к регистру (`$Age` и `$aGe` являются двумя различными переменными)

4. Основные типы данных в `php` - в PHP есть десять базовых типов данных: `bool`, `int`, `float`, `string`, `array`, `object`, `callable`, `mixed`

5. Какие существуют функции для работы с переменными в `php` вне зависимости от типа данных - анонимные функции, также известные как замыкания (`closures`), позволяют создавать функции, не имеющие определённых имён. Они наиболее полезны в качестве значений `callable`-параметров, но также могут иметь и множество других применений.

6. Предопределенные переменные в `php` - PHP предоставляет всем скриптам большое количество предопределённых переменных. Эти переменные содержат всё, от внешних данных до переменных среды окружения, от текста сообщений об ошибках до последних полученных заголовков.

7. Переменные переменных в php - переменная переменной берет значение переменной и рассматривает его как имя переменной.

8. Выражения в php – ниже представлены возможные выражения на php.

9. Арифметические операторы в php -  $+\$a$  – Идентичность;  $-\$a$  – Отрицание;  $\$a + \$b$  – Сложение;  $\$a - \$b$  – Вычитание;  $\$a * \$b$  – Умножение;  $\$a / \$b$  – Деление;  $\$a \% \$b$  - Деление по модулю;  $\$a ** \$b$  – Возведение в степень;

10. Битовые операторы -  $\$a \& \$b$  - Побитовое 'и';  $\$a | \$b$  - Побитовое 'или';  $\$a \wedge \$b$  - Исключающее или;  $\sim \$a$  – Отрицание;  $\$a \ll \$b$  - Сдвиг влево;  $\$a \gg \$b$  - Сдвиг вправо.

11. Оператор присваивания в php - базовый оператор присваивания обозначается как "=". Операторы сравнения в php. Результатом выполнения оператора присваивания является само присвоенное значение.

12. Операторы сравнения -  $\$a == \$b$  – равно,  $\$a === \$b$  – тождественно равно,  $\$a != \$b$ ,  $\$a <> \$b$  – не равно,  $\$a !== \$b$  – тождественно не равно,  $\$a < \$b$  – меньше,  $\$a > \$b$  больше.

13. Логические операторы в php -  $\$a \text{ and } \$b$  – И;  $\$a \text{ or } \$b$  – Или;  $\$a \text{ xor } \$b$  - Исключающее или;  $! \$a$  – Отрицание;  $\$a \&\& \$b$  – И;  $\$a || \$b$  – Или

14. Условная конструкция в php- условные конструкции позволяют направлять работу программы в зависимости от условия по одному из возможных путей. И одной из таких конструкций в языке PHP является конструкция `if..else` . Тернарная операция состоит из трех операндов и имеет следующее определение: [первый операнд - условие] ? [второй операнд] : [третий операнд].

15. Циклы в php - циклы позволяют повторять определенное (и даже неопределенное - когда работа цикла зависит от условия) количество раз

различные операторы. Данные операторы называются телом цикла. Проход цикла называется итерацией. PHP поддерживает три вида циклов: Цикл с предусловием (while); Цикл с постусловием (do-while); Цикл со счетчиком (for); Специальный цикл перебора массивов (foreach).

16. Конструкции switch и match в php - оператор switch похож на ряд операторов IF с одинаковым условием. Во многих случаях вам может понадобиться сравнивать одну и ту же переменную (или выражение) с множеством различных значений и выполнять различные участки кода в зависимости от того, какое значение принимает эта переменная (или выражение). Выражение match предназначено для ветвления потока исполнения на основании проверки совпадения значения с заданным условием. Аналогично оператору switch, выражение match принимает на вход выражение, которое сравнивается с множеством альтернатив. Но, в отличие от switch, оно обрабатывает значение в стиле, больше похожем на тернарный оператор. Также, в отличие от switch, используется строгое сравнение (===), а не слабое (==). Include и require в php.

17. Include и require в php - В PHP есть две функции, которые используются для помещения содержимого файла, содержащего исходный код PHP, в другой файл PHP. Это функции Include() и Require(). Обе функции одинаковы, но они имеют одно различие. Разница в том, что функция include() выдает предупреждение, но скрипт продолжит выполнение, а функция require() выдает предупреждение и фатальную ошибку, т.е. скрипт не будет продолжать выполнение. Эти две функции используются для помещения данных файла в другой файл PHP перед его выполнением сервером.

18. Функции в php - Имена функций следуют тем же правилам, что и другие метки в PHP. Корректное имя функции начинается с буквы или знака подчеркивания, за которым следует любое количество букв, цифр или знаков подчеркивания.

## **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).