



МИНОБРНАУКИ РОССИИ

*Федеральное государственное бюджетное образовательное учреждение
высшего образования*

«МИРЭА – Российский технологический университет»

РТУ МИРЭА

Институт Информационных технологий (ИТ)

**Кафедра инструментального и прикладного программного обеспечения
(ИиППО)**

**ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ № 2
по дисциплине**

«Проектирование и разработка серверных частей интернет-ресурсов»

Выполнил студент группы ИКБО-20-23

Комисарик М.А.

Принял Ассистент кафедры ИиППО

Благирев М.М.

Практическая работа выполнена

«__»_____202__ г.

(подпись студента)

«Зачтено»

«__»_____202__ г.

(подпись руководителя)

Москва 2025

ХОД РАБОТЫ

```
Dockerfile x
1 FROM php:apache
2 EXPOSE 80
3 RUN apt-get update && apt-get install -y \
4     libfreetype-dev \
5     libjpeg62-turbo-dev \
6     libpng-dev \
7     libonig-dev \
8     libgmp-dev \
9     && docker-php-ext-configure gd --with-freetype --with-jpeg \
10    && docker-php-ext-install -j$(nproc) gd gmp \
11    && docker-php-ext-enable gd
```

Рисунок 1 - Dockerfile

```
docker-compose.yml x
1 services:
2   web_server:
3     build:
4       context: .
5       dockerfile: Dockerfile
6     volumes:
7       - ./html:/var/www/html/
8     ports:
9       - 8000:80
10    restart: always
```

Рисунок 2 – docker-compose.yml

```
1 <?php
2 function getNextField(&$num, $mask)
3 {
4     $out = gmp_and($num, $mask);
5     $num = gmp_div_q($num, gmp_pow(num: 2, exponent: 8));
6     return gmp_intval($out);
7 }
8
9 $num = gmp_init($_GET["num"]);
10 $mask = gmp_init(num: 0b11111111);
11 $fig = getNextField(&$num, $mask);
12 $r = getNextField(&$num, $mask);
13 $g = getNextField(&$num, $mask);
14 $b = getNextField(&$num, $mask);
15 $x = getNextField(&$num, $mask);
16 $y = getNextField(&$num, $mask);
17 $v1 = getNextField(&$num, $mask);
18 $v2 = getNextField(&$num, $mask);
19
20 $width = 256;
21 $height = 256;
22
23 $image = imagecreatetruecolor($width, $height);
24
25 $backgroundColor = imagecolorallocate($image, red: 255, green: 255, blue: 255);
26 $figureColor = imagecolorallocate($image, $r, $g, $b);
27
28 imagefill($image, x: 0, y: 0, $backgroundColor);
29
```

Рисунок 3 – drawer.php, часть 1

```

30     if ($fig == 0)
31     {
32         imagefilledellipse($image, $x, $y, $v1, $v1, $figureColor);
33     }
34     else if ($fig > 0)
35     {
36         if ($fig == 1)
37         {
38             imagefilledellipse($image, $x, $y, width: 5, height: 5, $figureColor);
39         }
40         else if ($fig == 2)
41         {
42             imageline($image, $x, $y, $v1, $v2, $figureColor);
43         }
44         else
45         {
46             $arr = [$x, $y, $v1, $v2];
47             for ($i = 2; $i < $fig; $i++)
48             {
49                 $nextX = getNextField(&$num, $mask);
50                 $nextY = getNextField(&$num, $mask);
51                 array_push(&$arr, $nextX, $nextY);
52             }
53             imagefilledpolygon($image, $arr, $figureColor);
54         }
55     }
56
57     header(header: 'Content-Type: image/png');
58
59     imagepng($image);
60
61     imagedestroy($image);

```

Рисунок 4 – drawer.php, часть 2

Число, указываемое в качестве параметра `num` содержит в себе несколько полей, каждое из которых длиной 1 байт:

- первое поле – тип фигуры (0 – круг, 1 – точка, 2 – прямая, 3 и более – многоугольник с соответствующим количеством углов),
- 2-4 поля – цвет фигуры (r, g, b),
- 5, 6 поля – x, y координаты круга или первая координата фигуры,
- 7 поле – x координата второй вершины фигуры или диаметр круга,
- 8 поле – y координата второй вершины фигура,
- 9 и далее – x и y координаты последующих вершин соответственно.

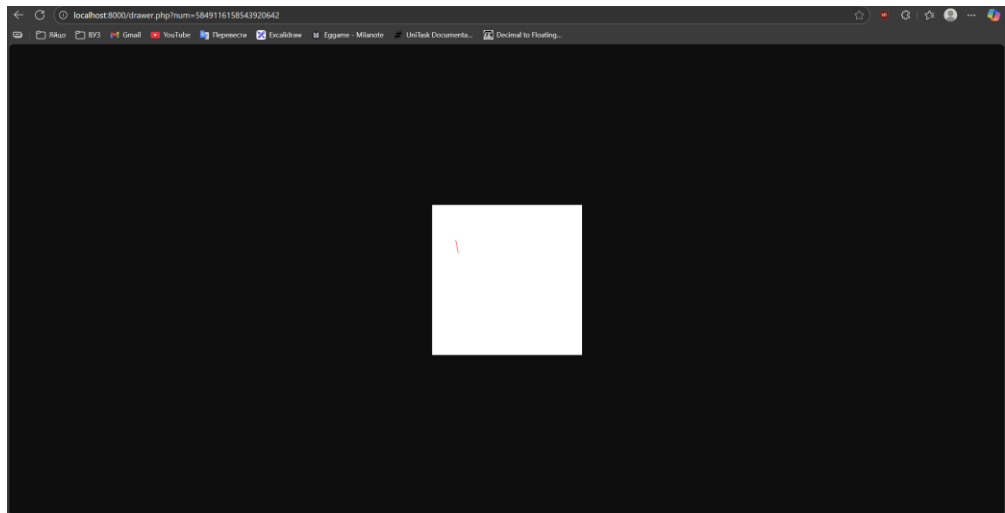


Рисунок 5 – Работа drawer.php, часть 1

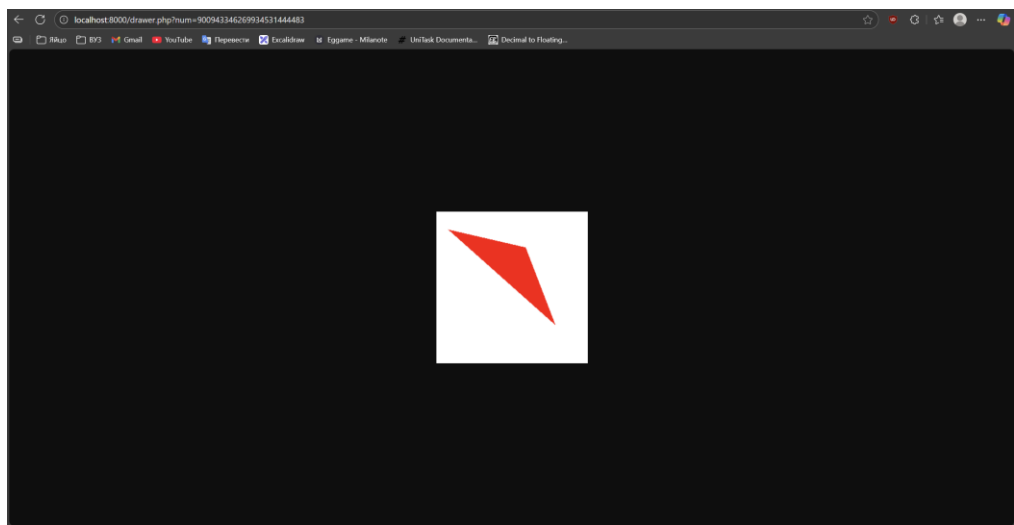


Рисунок 6 – Работа drawer.php, часть 2

```

1  <?php
2
3  3 usages  Krezon *
4  function mergeSort(array $arr): array
5  {
6      $length = count($arr);
7      if ($length <= 1) return $arr;
8
9      $mid = (int)($length / 2);
10     $left = array_slice($arr, offset: 0, $mid);
11     $right = array_slice($arr, $mid);
12
13     $left = mergeSort($left);
14     $right = mergeSort($right);
15
16     return merge($left, $right);
17 }

```

Рисунок 7 – функция merge файла sort.php

```

18 function merge(array $left, array $right): array
19 {
20     $result = [];
21     $i = 0;
22     $j = 0;
23
24     while ($i < count($left) && $j < count($right))
25     {
26         if ($left[$i] <= $right[$j])
27         {
28             $result[] = $left[$i];
29             $i++;
30         } else
31         {
32             $result[] = $right[$j];
33             $j++;
34         }
35     }
36
37     while ($i < count($left))
38     {
39         $result[] = $left[$i];
40         $i++;
41     }
42
43     while ($j < count($right))
44     {
45         $result[] = $right[$j];
46         $j++;
47     }
48
49     return $result;
50 }

```

Рисунок 8 – функция mergeSort файла sort.php

```

52 echo "<p>Initial values: " . $_GET["nums"] . "</p>";
53 $nums = explode( separator: ",", $GET["nums"]);
54
55 $nums = mergeSort($nums);
56 echo "<p>Sorted values: ";
57 foreach ($nums as $num)
58 {
59     echo "$num ";
60 }
61 echo "</p>";

```

Рисунок 9 – sort.php, часть 3

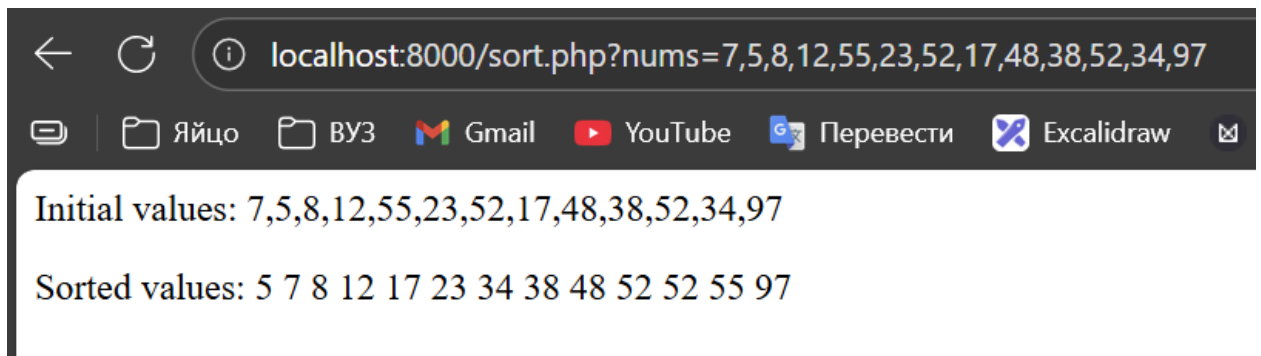


Рисунок 10 – Работа sort.php

```
1 <?php
2 4 usages new *
3 function executeCommand($command): string
4 {
5     $output = '';
6     @exec( command: $command . " 2>&1", &: $output);
7     return implode( separator: "\n", $output);
8 }
9 ?>
10 <!DOCTYPE html>
11 <html lang="ru">
12 > <head...>
13 </head>
14 <body>
15 <pre>
16 Пользователь PHP: <?php echo executeCommand( command: 'whoami'); ?>
17
18 Идентификатор пользователя: <?php echo executeCommand( command: 'id'); ?>
19
20 </pre>
21
22 <h3>Дисковое пространство</h3>
23 <pre>
24 <?php echo executeCommand( command: 'df -h'); ?>
25 </pre>
26
27 <h3>Содержимое текущей директории</h3>
28 <pre>
29 <?php echo executeCommand( command: 'ls -la'); ?>
30 </pre>
31 </body>
32 </html>
```

Рисунок 11 – admin.php

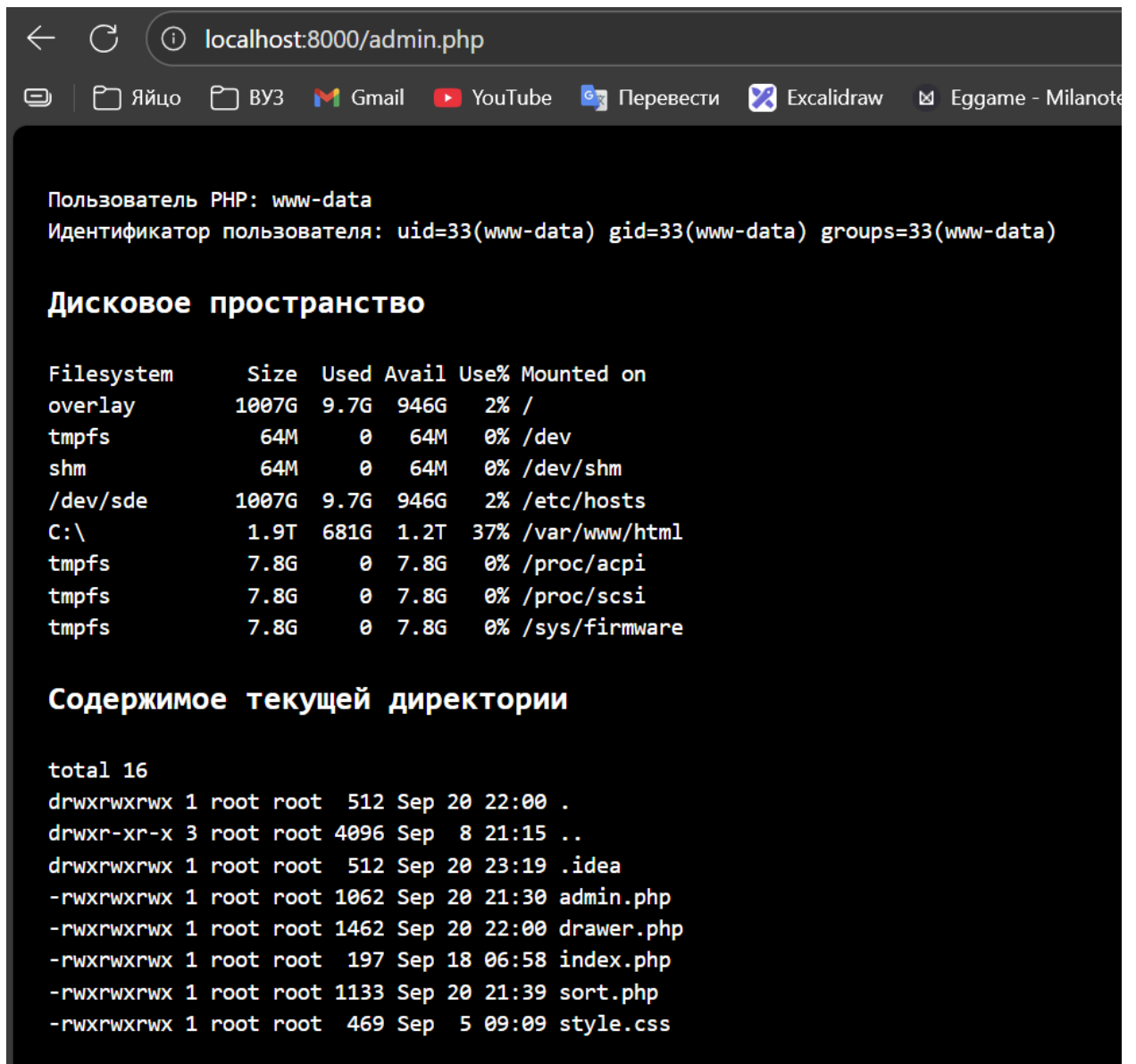


Рисунок 12 – Работа admin.php