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

**Институт информационных технологий**

КАФЕДРА ИНСТРУМЕНТАЛЬНОГО И ПРИКЛАДНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ (ИиППО)

Практическая работа 2

по дисциплине «Разработка серверных частей интернет-ресурсов»

Выполнил студент группы ИКБО-01-20 *Клочек А.М.*

Принял преподаватель *Волков М.Ю.*

Практические работы выполнены «\_\_\_»\_\_\_\_\_\_\_2022г. *подпись студента*

«Зачтено» «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2022г. *подпись преподавателя*

Москва 2022

# ОГЛАВЛЕНИЕ

[Цель работы 3](#_Toc113206000)

[Теоретическое введение 3](#_Toc113206001)

[Задание на практическую работу 3](#_Toc113206002)

[Выполнение практической работы 4](#_Toc113206003)

[Вывод 6](#_Toc113206004)

[Ссылка на удалённый репозиторий проекта 6](#_Toc113206005)

[Ответы на вопросы: 6](#_Toc113206006)

[Список использованной литературы 8](#_Toc113206007)

# 

**Цель работы:**

Получить навыки создания сервера. Познакомиться с языком PHP.

**Задание на практическую работу:**

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.

**Упражнение 1**

Предлагается создать веб-сервис Drawer для рисования svg объектов. Ему передается один параметр - целое число, представляющее закодированная фигура для рисования. Пример с заданием кода фигуры для рисования.

**Упражнение 1**

Кодирование фигуры состоит из нескольких параметров: форма, цвет, размеры ограничивающего прямоугольника примитива. Должен передаваться всего 1 целочисленный параметр, который будет содержать всю информацию о фигуре, которую требуется нарисовать. Для реализации данной задачи требуется использовать язык программирования PHP, битовые операции и разделение проекта на файлы.

**Упражнение 2**

Реализовать одну из сортировок на языке программирования PHP по варианту:

Вариант 1 - Быстрая сортировка

Вариант 2 - Сортировка вставками

Вариант 3 - Сортировка слиянием

Вариант 4 - Сортировка выбором

*Вариант 5 - Сортировка Шелла*

Массив передается скрипту как параметр строка: состоящий из значений элементов массива, разделенных запятыми. Итогом является веб-страница, содержащая отсортированный массив. Для реализации данной задачи требуется использовать язык программирования PHP, и разделение проекта на файлы.

**Упражнение 3**

Реализовать информационно-административную веб-страницу о сервере с помощью таких команд Unix как: ls, ps, whoami, id и так далее.

**Выполнение практической работы:**

Каждый сервис состоит из 2х частей: frontend-часть задает вид страницы, проверяет входящие данные и запускает backend-часть – имплементацию упражнения. Главная страница содержит view-компоненты для взаимодействия с сервисами, она запускает каждый из них.

Сервис рисования использует URL-параметер кодирующий фигуру, цвет и ее размеры следующим образом: используется восьми битное число, в котором самые левые 2 бита кодируют форму (00 – круг, 01 – квадрат, 10 - прямоугольник), следующие 2 бита кодируют цвет (00 – красный, 01 – зеленый, 10 - синий), остальные 4 бита кодируют длинну (2 бита) и высоту (2 бита) соответствено.

Сервис сортировки использует сортировку Шелла согласно варианту (20) и принимает в качестве параметра строку с числами, разделенными запятыми.

Информационный сервис содержит несколько кнопок, вызывающие соответствующие их названиям команды операционной системы, которые выполняются и затем страница отображает результат.

Файлы проекта расположены в удаленном репозитории на платформе GitHub.com, расположенном по ссылке: https://github.com/Agser228/RSCHIR/tree/PZ-2

# Результаты выполнения работы представлены на рисунках 1-13.

# 

Рисунок 1 – Dockerfile

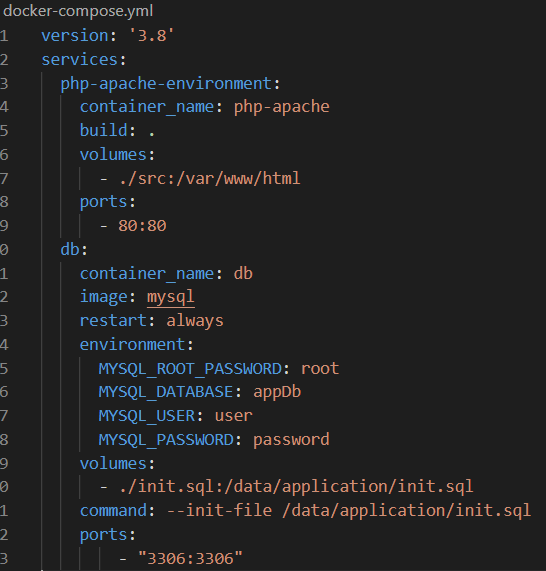


Рисунок 2 – Docker-compose

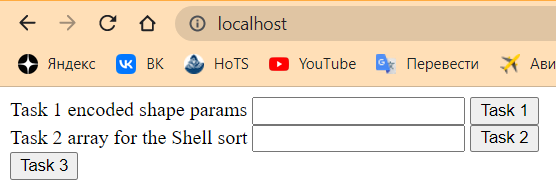


Рисунок 3 – Главная страница

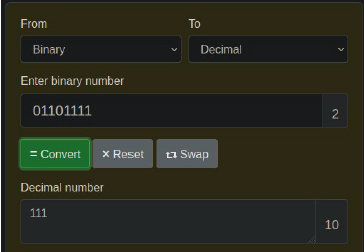


Рисунок 4 – Преобразование бинарного числа в десятичное

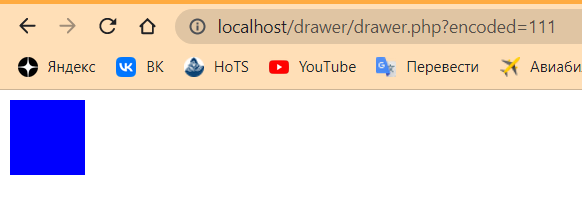


Рисунок 5 – Сервис рисования

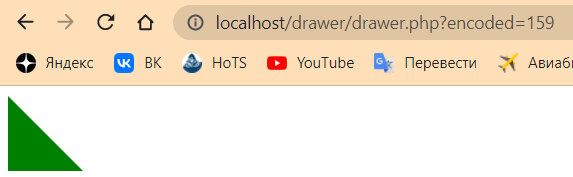


Рисунок 6 – Сервис рисования

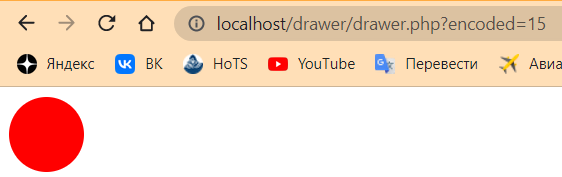


Рисунок 7 – Сервис рисования

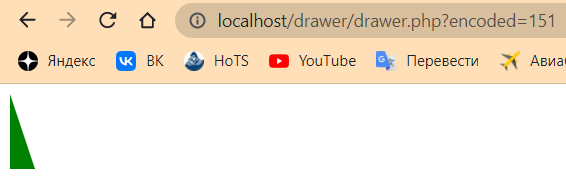


Рисунок 8 – Сервис рисования

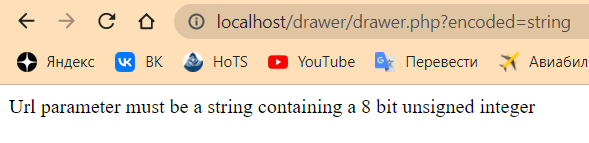


Рисунок 9 – Сервис сортировки

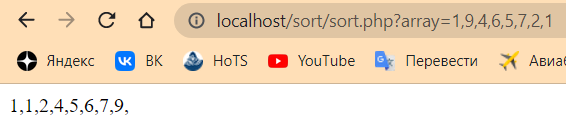


Рисунок 10 – Сервис сортировки

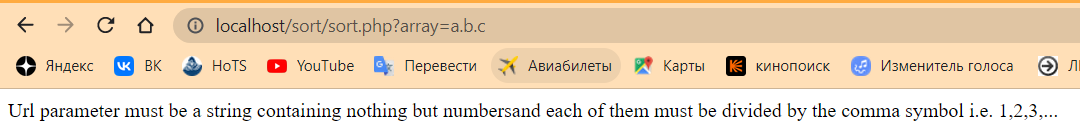


Рисунок 11 – Сервис сортировки

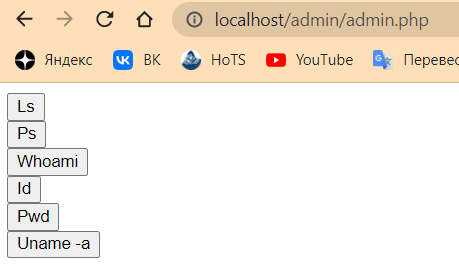


Рисунок 12 – Информационный сервис

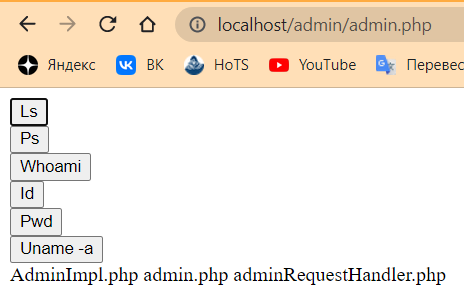


Рисунок 13 – Информационный сервис

**Вывод:**

# В ходе выполнения данной работы были приобретены навыки и опыт проектирования, разработки, конфигурирования и развертки веб-серверного ПО, а также его контейнеризация средствами Docker. Было также создано несколько полезных веб-сервисов, произведено знакомство с языком программирования PHP и закреплены базовые навыки работы с ним.

# Ссылка на удалённый репозиторий проекта

Ссылка на репозиторий: https://github.com/Agser228/RSCHIR/tree/PZ-2

# Ответы на вопросы:

1) Сервер – это сторона, которая принимает запросы от клиента и предпринимает соответствующие действия с данными необходимые по запросу.

2) База данных — совокупность данных, хранимых в соответствии со схемой данных, манипулирование которыми выполняют в соответствии с правилами средств моделирования данных.

3) API (программный интерфейс приложения, интерфейс прикладного программирования) — описание способов, которыми одна компьютерная программа может взаимодействовать с другой программой.

4) Сервис - одна ограниченная функция сервера.

5) «Клиент — сервер» — архитектура, в которой задания или сетевая нагрузка распределены между поставщиками услуг, называемыми серверами, и заказчиками услуг, называемыми клиентами.

6) Веб-серверы, Почтовые серверы, Файл-серверы, Прокси-серверы, Почтовые серверы, Веб-серверы, Серверы приложений, Серверы баз данных.

7) Вертикальная масштабируемость – увеличение производительности серверной системы. Горизонтальная масштабируемость – разбиение на более мелкие структурные компоненты и разнесение их (распараллеливание).

8) Протокол передачи данных — набор определённых инструкций интерфейса логического уровня, которые определяют способы обмена информации между различными программами или их компонентами.

9) Тонкий клиент – частный случай клиент-серверной архитектуры, в которой большая часть логики выполняется на сервере. Толстый же клиент, наоборот, когда логика выполняется на стороне клиента.

10) Паттерн MVC: общие тезисы. MVC – это способ организации кода, при котором выделяются компоненты, модель данных приложения, представление и контроллер. Эти компоненты независимы, модификации одного из них, не влияют на друга, что прощает процесс разработки приложения.

11) Паттерн MVC: Model-View-Presenter. Производный от MVC паттерн проектирования. Отличие от MVC в том, что элемент Presenter берёт на себя обработку событий пользовательского интерфейса, в то время как в MVC за это отвечает представление.

12) Паттерн MVC: Model-View-View Model. ModelView – модель представления, обёртка над моделью, преобразованная для прямого использования представлению, содержит команды, которые использует представление для работы с моделью.

13) MVC: Модель представляет собой логику работы с данными и описание фундаментальных данных, необходимых для работы приложения. Представление (View) отвечает за отображение данных модели пользователю, реагируя на изменения модели. Представление (View) отвечает за отображение данных модели пользователю, реагируя на изменения модели.

14) Docker — программное обеспечение для автоматизации развёртывания и управления приложениями в средах с поддержкой контейнеризации

15) Dockerfile – это сценарий, который состоит из последовательности команд и аргументов, необходимых для создания образа.

16) Docker Compose — это инструментальное средство, входящее в состав Docker. Оно предназначено для решения задач, связанных с развёртыванием проектов.

17) LAMP — Linux, Apache, MariaDB/MySQL, PHP.

# Список использованной литературы

1. Официальный репозиторий образов Docker, URL: <https://hub.docker.com/> (дата последнего обращения: 02.09.2022)
2. Docker самый простой и понятный туториал. Изучаем докер, так, если бы он был игровой приставкой, URL: <https://badcode.ru/docker-tutorial-dlia-novichkov-rassmatrivaiem-docker-tak-iesli-by-on-byl-ighrovoi-pristavkoi/amp/> (дата последнего обращения: 02.09.2022)
3. Шпаргалка с командами Docker: сайт. – <URL:https://habr.com/ru/company/flant/blog/336654/> (дата обращения: 02.01.2022). – Текст: электронный.
4. Официальная документация языка PHP, URL: <https://www.php.net/manual/ru/index.php> (дата последнего обращения: 02.09.2022)
5. Документация PHP: сайт. – URL: <https://www.php.net/docs.php> (дата обращения: 02.01.2022).
6. Medium blog. Статья «Apache Airflow in 10 minutes» [Электронный ресурс]. – URL: (дата последнего обращения: 02.09.2022)