



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 №1

по дисциплине «Разработка серверных частей интернет-ресурсов»

Студент группы ИКБО-24-20

Постнов Никита Сергеевич

(подпись студента)

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

(подпись руководителя)

Работа представлена « ____ » _____ 2022 г.

Допущен к работе « ____ » _____ 2022 г.

Москва 2022

Содержание

1	Цель работы	3
2	Ход работы	3
3	Вывод	4
	Ответы на вопросы к практической работе	4
	Ссылка на удаленный репозиторий проекта	7
	Список использованной литературы	7

1. Цель работы

Целью данной работы является создание собственной конфигурации серверного программного обеспечения.

2. Ход работы

1. Стек технологий

В данной практической работе используется WSL2, язык программирования PHP, веб-сервер Apache и СУБД MySQL.

2. Выполнение

Создадим docker-compose файл, в котором запустим два образа: сервер Apache с интерпретатором php, СУБД MySQL. Прокинем порт сервера наружу, чтобы отправлять запросы с помощью браузера на хост.

Листинг 1. Файл docker-compose

```
version: '3'
services:
  apache:
    build: ./apache
    volumes:
      - ./src:/var/www/html/
    ports:
      - 80:80
  my-sql:
    image: mysql:latest
    volumes:
      - ./db:/docker-entrypoint-initdb.d
    environment:
      - MYSQL_ROOT_PASSWORD=root
```

FROM php:apache

RUN docker-php-ext-install mysqli

The screenshot shows a web browser window with the address bar set to 'localhost'. The page displays a table titled 'Таблица пользователей данного продукта' (Table of users of this product). The table has three columns: 'Id', 'Name', and 'Surname'. It contains four rows of data. Below the table is a blue banner for 'PHP Version 8.1.10' with the PHP logo. Underneath the banner is a detailed system information table.

Id	Name	Surname
1	Alex	Rover
2	Bob	Marley
3	Kate	Yandson
4	Lilo	Black

PHP Version 8.1.10	
System	Linux f3fb7442a257 5.10.16.3-microsoft-standard-WSL2 #1 SMP Fri Apr 2 22:23:49 UTC 2021 x86_64
Build Date	Sep 1 2022 21:15:46
Build System	Linux 072284680fe0 5.10.0-13-cloud-amd64 #1 SMP Debian 5.10.106-1 (2022-03-17) x86_64 GNU/Linux
Configure Command	'./configure' '--build=x86_64-linux-gnu' '--with-config-file-path=/usr/local/etc/php' '--with-config-file-scan-dir=/usr/local/etc/php/conf.d' '--enable-option-checking=fatal' '--with-mhash' '--with-pic' '--enable-ftp' '--enable-mbstring' '--enable-mysqlnd' '--with-password-argon2' '--with-sodium=shared' '--with-pdo-sqlite=/usr' '--with-sqlite3=/usr' '--with-curl' '--with-iconv' '--with-openssl' '--with-readline' '--with-zlib' '--disable-phpdbg' '--with-pear' '--with-libdir=lib/x86_64-linux-gnu' '--disable-cgi' '--with-apxs2' 'build_alias=x86_64-linux-gnu'
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/usr/local/etc/php
Loaded Configuration File	(none)
Scan this dir for additional .ini files	/usr/local/etc/php/conf.d
Additional .ini files parsed	/usr/local/etc/php/conf.d/docker-php-ext-mysqli.ini, /usr/local/etc/php/conf.d/docker-php-ext-sodium.ini
PHP API	20210902
PHP Extension	20210902
Zend Extension	420210902
Zend Extension Build	API420210902,NTS
PHP Extension Build	API20210902,NTS

Рис. 2.1. Результат работы

3. Вывод

В ходе практической работы были получены навыки по созданию конфигурации сервера.

Ответы на вопросы к практической работе

1. Сервер – программный компонент вычислительной системы, выполняющий сервисные (обслуживающие) функции по запросу клиента, предоставляя ему доступ к определённым ресурсам или

услугам.

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) — описание способов взаимодействия одной компьютерной программы с другими.
4. Сервис - это разделенные программные компоненты, предоставляющие определенную функциональность и используемые в составе многих приложений.
5. «Клиент — сервер» — архитектура, в которой задания или сетевая нагрузка распределены между поставщиками услуг, называемыми серверами, и заказчиками услуг, называемыми клиентами.
6. Виды сервисов - Серверы приложений, веб-серверы, серверы баз данных, файл-серверы, прокси-серверы, файрволы, почтовые серверы.
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 правил или соглашений интерфейса логического уровня, который определяет обмен данными между различными программами.
9. Тонкий клиент — большая часть логики выполняется на сервере. Толстый клиент — логика выполняется на стороне клиента.
10. MVC: Model - данные, методы для работы с данными, изменения и обновления данных. View - Отображение данных, оформление и другие

аспекты презентации модели. Controllor - Реагирует на действия пользователя, интерпретирует данные, введенные пользователем, и информирует модель и производит необходимые манипуляции с моделью и видом.

11. MVP: Модель (англ. Model) — данные для отображения. Вид (англ. View) — реализует отображение данных (из Модели), обращается к Presenter за обновлениями, перенаправляет события от пользователя в Presenter. Представитель (англ. Presenter) — реализует взаимодействие между Моделью и Видом и содержит в себе всю логику представления данных о предметной области; при необходимости получает данные из хранилища и преобразует для отображения во View.
12. MVVM: Модель представляет собой логику работы с данными и описание фундаментальных данных, необходимых для работы приложения. Модель (англ. Model) (так же, как в классической MVC) представляет собой логику работы с данными и описание фундаментальных данных, необходимых для работы приложения. Модель (англ. Model) (так же, как в классической MVC) представляет собой логику работы с данными и описание фундаментальных данных, необходимых для работы приложения.
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 развёртывание приложений средствами контейнеризации.

15. Dockerfile – файл конфигурации сборки контейнера, в нём обязательно указывается базовый образ, а также может содержаться последовательность команд для создания контейнера.
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.

Ссылка на удаленный репозиторий проекта

https://github.com/1TSOP/RSCHIR_1PR

Список использованной литературы

1. Документация языка PHP [Электронный ресурс]. – URL:
<https://www.php.net/docs.php> (дата последнего обращения: 08.09.2022)
2. Документация docker [Электронный ресурс]. – URL:
<https://docs.docker.com/> (дата последнего обращения: 08.09.2022)