|  |  |  |
| --- | --- | --- |
|  |  |  |
| 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»

**Тема практической работы:**

**Студент группы** ИКБО-20-19 Анваржонов Ж Т

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись студента)

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

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись руководителя)

Работа представлена «\_15\_» сентября 2021 г.

Допущен к работе «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2021 г.

Москва 2021

**Создание конфигурации серверного программного обеспечения**

**Docker-compose.yml:**

version: '3.8'

services:

  php:

    build: ./php

    volumes:

      - ./php:/var/www/html/

    ports:

      - 8000:80

    depends\_on:

      - db

  db:

    image: mysql

    restart: always

    volumes:

      - ./database/init.sql:/docker-entrypoint-initdb.d/init.sql

    environment:

         - MYSQL\_ALLOW\_EMPTY\_PASSWORD=true

**DockerFile:**

FROM php:8.0-apache

RUN docker-php-ext-install mysqli && docker-php-ext-enable mysqli

RUN apt-get update && apt-get upgrade -y

**Проверка работоспособности конфигурации:**

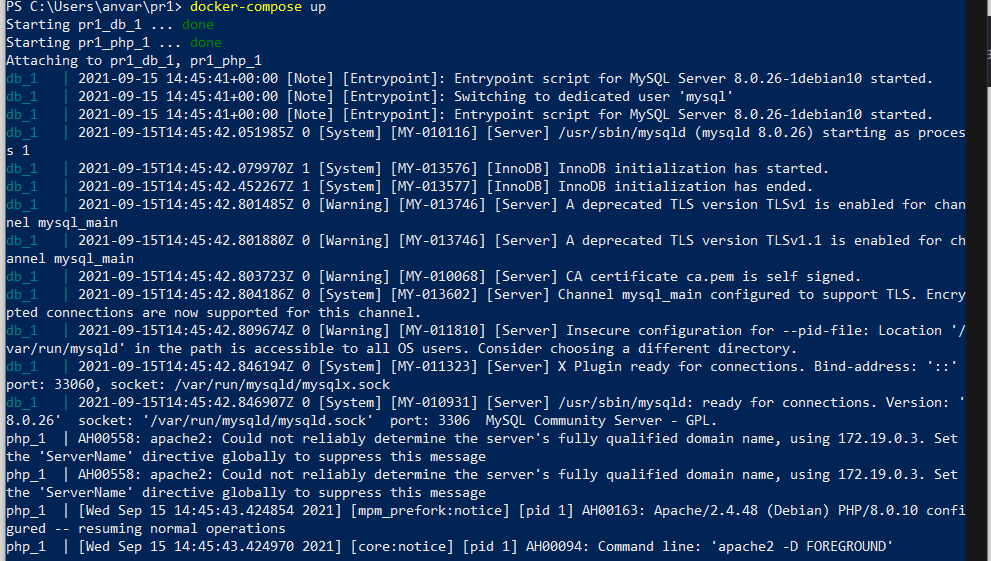
****

Рисунок 1. Выполнение docker-compose в командной строке

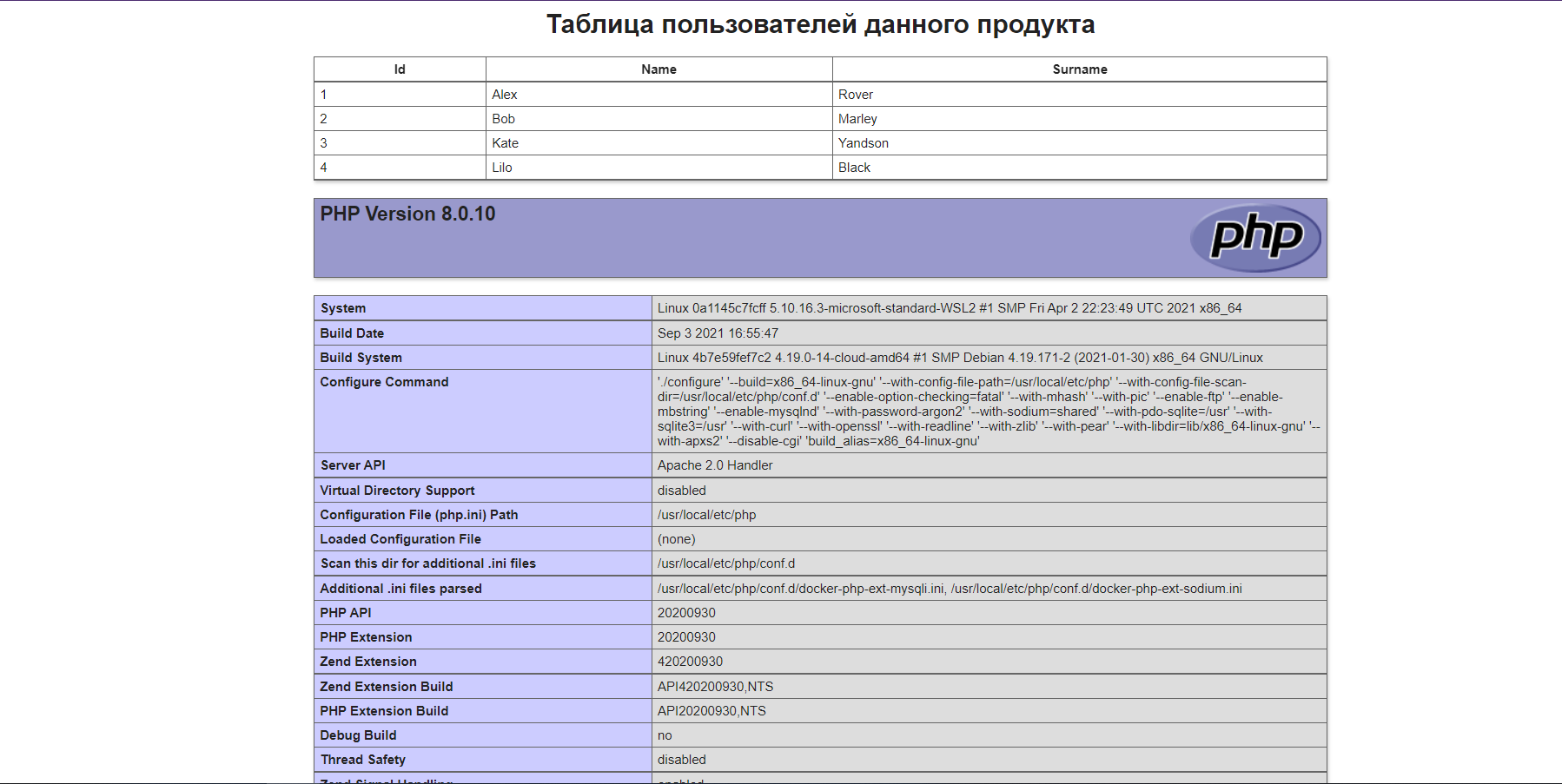
****

Рисунок 2. Тестовая страница

**Вопросы к практической работе.**

1. Сервер и клиент.

«**Клиент** — **сервер**» (англ. client–**server**) — вычислительная или сетевая архитектура, в которой задания или сетевая нагрузка распределены между поставщиками услуг, называемыми серверами, и заказчиками услуг, называемыми **клиентами**. Фактически **клиент** и **сервер** — это программное обеспечение.

1. База данных.

**База данных** — это упорядоченный набор структурированной информации или **данных**, которые обычно хранятся в электронном виде в компьютерной системе. **База данных** обычно управляется системой управления базами **данных** (СУБД).

1. API.

API — описание способов, которыми одна компьютерная программа может взаимодействовать с другой программой. Обычно входит в описание какого-либо интернет-протокола, программного каркаса или стандарта вызовов функций операционной системы.

1. Сервис, отличия от сервера.

Веб-сервер — идентифицируемая уникальным веб-адресом программная система со стандартизированными интерфейсами. Веб-службы могут взаимодействовать друг с другом и со сторонними приложениями посредством сообщений, основанных на определённых протоколах и соглашениях.

Сервер - сервер, принимающий HTTP-запросы от клиентов, обычно веб-браузеров, и выдающий им HTTP-ответы, как правило, вместе с HTML-страницей, изображением, файлом, медиа-потоком или другими данными

1. Архитектура клиент-сервер.

**Архитектура «клиент-сервер»**определяет общие принципы организации взаимодействия в сети, где имеются серверы, узлы-поставщики некоторых специфичных функций (сервисов) и клиенты (потребители этих функций).  
Практические реализации такой архитектуры называются **клиент-серверными технологиями.**

1. Виды сервисов.

Делятся на SOAP, REST

1. Масштабируемость.

Масштаби́руемость — в электронике и информатике означает способность системы, сети или процесса справляться с увеличением рабочей нагрузки при добавлении ресурсов.

1. Протоколы передачи данных.

Протокол передачи данных — набор определённых правил или соглашений интерфейса логического уровня, который определяет обмен данными между различными программами.

1. Тонкий и толстый клиенты.

**Толстые клиенты** работают с информацией на основе собственных аппаратных и программных возможностей, в то же время **тонкие** применяют ПО центрального сервера только чтобы обработать данные, предоставляя системе лишь требуемый графический интерфейс для выполнения работы пользователем

1. Паттерн MVC: общие тезисы.

Статическая страница на HTML не умеет реагировать на действия пользователя. Для двухстороннего взаимодействия нужны динамические веб-страницы. MVC — ключ к пониманию разработки динамических веб-приложений, поэтому разработчику нужно знать эту модель.

MVC расшифровывается как модель-представление-контроллер (от англ. model-view-controller). Это способ организации кода, который предполагает выделение блоков, отвечающих за решение разных задач. Один блок отвечает за данные приложения, другой отвечает за внешний вид, а третий контролирует работу приложения. Компоненты MVC:

* Модель — этот компонент отвечает за данные, а также определяет структуру приложения. Например, если вы создаете To-Do приложение, код компонента model будет определять список задач и отдельные задачи.
* Представление — этот компонент отвечает за взаимодействие с пользователем. То есть код компонента view определяет внешний вид приложения и способы его использования.
* Контроллер — этот компонент отвечает за связь между model и view. Код компонента controller определяет, как сайт реагирует на действия пользователя. По сути, это мозг MVC-приложения.

1. Паттерн MVC: Model-View-Presenter.

Model-View-Presenter — шаблон проектирования, производный от MVC, который используется в основном для построения пользовательского интерфейса.

1. Паттерн MVC: Model-View-View Model.

Model-View-ViewModel — шаблон проектирования архитектуры приложения. Представлен в 2005 году Джоном Госсманом как модификация шаблона Presentation Model. Ориентирован на современные платформы разработки, такие как Windows Presentation Foundation, Silverlight от компании Microsoft, ZK framework

1. Паттерн MVC: Model-View-Controller.

Model-View-Controller — схема разделения данных приложения, и управляющей логики на три отдельных компонента: модель, представление и контроллер — таким образом, что модификация каждого компонента может осуществляться независимо. Модель предоставляет данные и реагирует на команды контроллера, изменяя своё состояние

1. Docker: общие тезисы и определения.

Docker — это программное обеспечение для автоматизации развёртывания и управления приложениями в средах с поддержкой контейнеризации.

1. Dockerfile.

В файлах Dockerfile содержатся инструкции по созданию образа. С них, набранных заглавными буквами, начинаются строки этого файла. После инструкций идут их аргументы. Инструкции, при сборке образа, обрабатываются сверху вниз. Вот как это выглядит:

1. Docker Compose.

**Docker Compose** — **это** средство, разработанное для помощи в определении и совместном использовании многоконтейнерных приложений.

1. LAMP.

LAMP — акроним, обозначающий набор серверного программного обеспечения, широко используемый во Всемирной паутине. LAMP назван по первым буквам входящих в его состав компонентов: Linux — операционная система Linux; Apache — веб-сервер; MariaDB / MySQL — СУБД;

Вывод : В этой работы мы сделали конфигурацию в связке LAMP.

**Список использованной литературы:**

**1.** Сейерс, Э. Х. Docker на практике / Э. Х. Сейерс, А. Милл ; перевод с английского Д. А. Беликов. — Москва : ДМК Пресс, 2020. — 516 с. — ISBN 978-5-97060-772-5. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/131719 (дата обращения: 15.09.2021). — Режим доступа: для авториз. пользователей.

2. Кочер, П. С. Микросервисы и контейнеры Docker : руководство / П. С. Кочер ; перевод с английского А. Н. Киселева. — Москва : ДМК Пресс, 2019. — 240 с. — ISBN 978-5-97060-739-8. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/123710 (дата обращения: 15.09.2021). — Режим доступа: для авториз. пользователей.

3. Поляков, Е. В. PHP на примерах : учебное пособие / Е. В. Поляков. — Санкт-Петербург : Наука и Техника, 2017. — 256 с. — ISBN 978-5-94387-733-9. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/101553 (дата обращения: 15.09.2021). — Режим доступа: для авториз. пользователей.

4. Филиппов, Ф. В. HTTP + PHP в примерах и задачах : учебное пособие / Ф. В. Филиппов, А. Н. Губин. — Санкт-Петербург : СПбГУТ им. М.А. Бонч-Бруевича, 2015. — 67 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/180044 (дата обращения: 15.09.2021). — Режим доступа: для авториз. пользователей.