

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

**Институт информационных технологий (ИИТ)**

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

ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ

по дисциплине «Разработка серверных частей интернет ресурсов»

Практическое задание № 3

|  |  |
| --- | --- |
| Студента группы | *ИКБО-03-20 Голиков Александр* |
| (подпись) |
| Старший преподаватель | Волков М. Ю. |
|  | (подпись) |
| Отчет представлен | « » 2022 г. |

Москва 2022 г

**СОДЕРЖАНИЕ**

1. [Цель работы 3](#_bookmark0)
2. [Ход работы 4](#_bookmark1)
3. [Выводы 8](#_bookmark2)
4. [Ответы на вопросы 9](#_bookmark3)

# Цель работы

Вариант 6 - библиотека

В задании предлагается создать сложную серверную конфигурацию, состоящую из связки apache+nginx+php+База данных. Возможно использование связки apache+php как единый компонент. В данной конфигурации предполагается создание как минимум 3 элементов(контейнеров) или использование как основы серверной конфигурации, созданной в практической работе № 1. В этой конфигурации предполагается акселерированное проксирование без кэширования.

# Ход работы

Создам файл docker-compose в котором будет описана логика контейнеризации приложения(рис. 1).

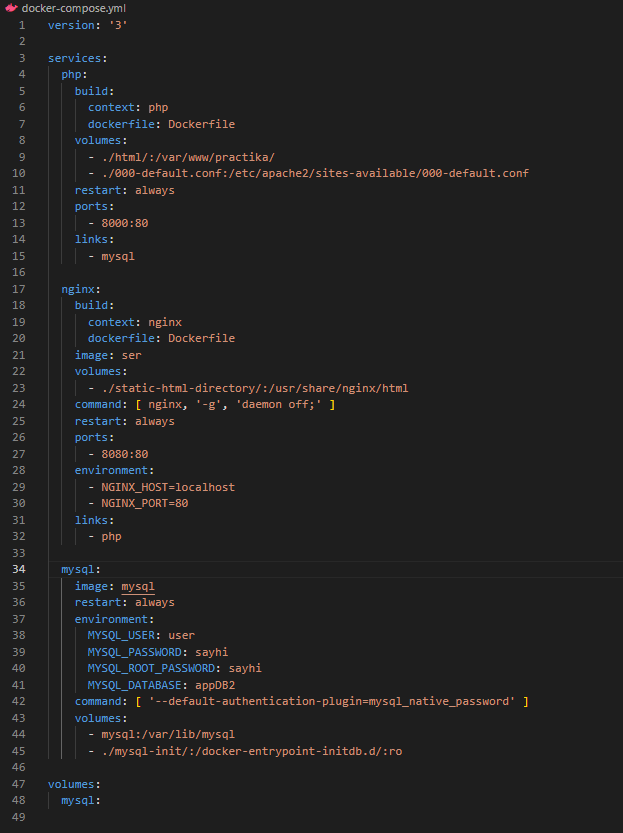


Рисунок 1 – Код docker-compose

В конфигурации nginx укажем файлы которые должны обрабатываться на стороне nginx, а какие на стороне apache.

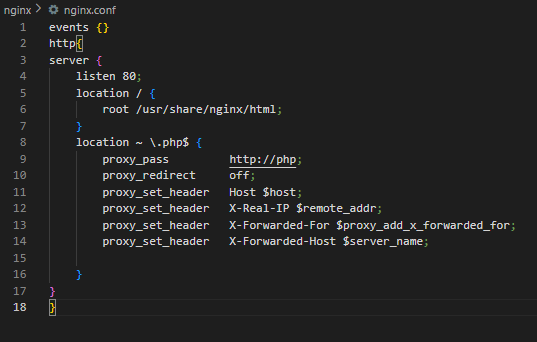


Рисунок 2 – конфигурация nginx

В файле 000-default.conf будет реализована авторизация пользователя используя созданную базу данных (Рисунок 3)

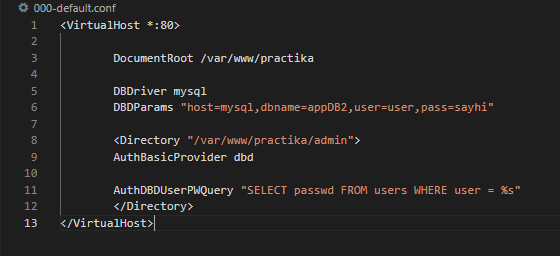


Рисунок 3 – Конфигурация apache

В файле init.sql создадим таблицу пользователей а также таблицу книг в библиотеке согласно варианту

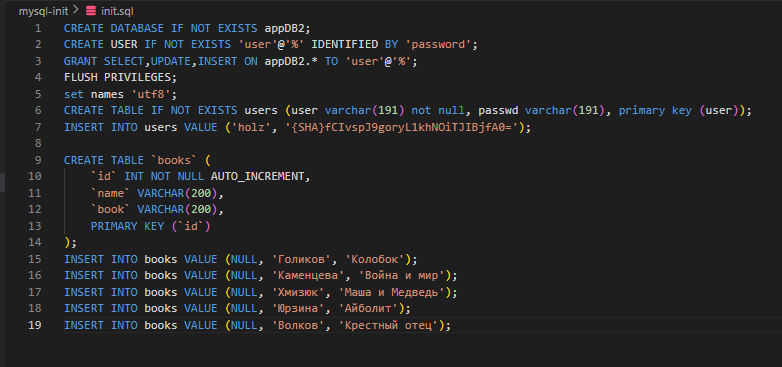


Рисунок 4 – Пример создания базы данных

В файле index.php (Рисунок 5) реализуем логику библиотеки, в которой пользователь может узнать наличие книги по названию.



Рисунок 5 – Файл index.php

Пример работы представленного кода (Рисунок 6.1, 6.2, 6.3)

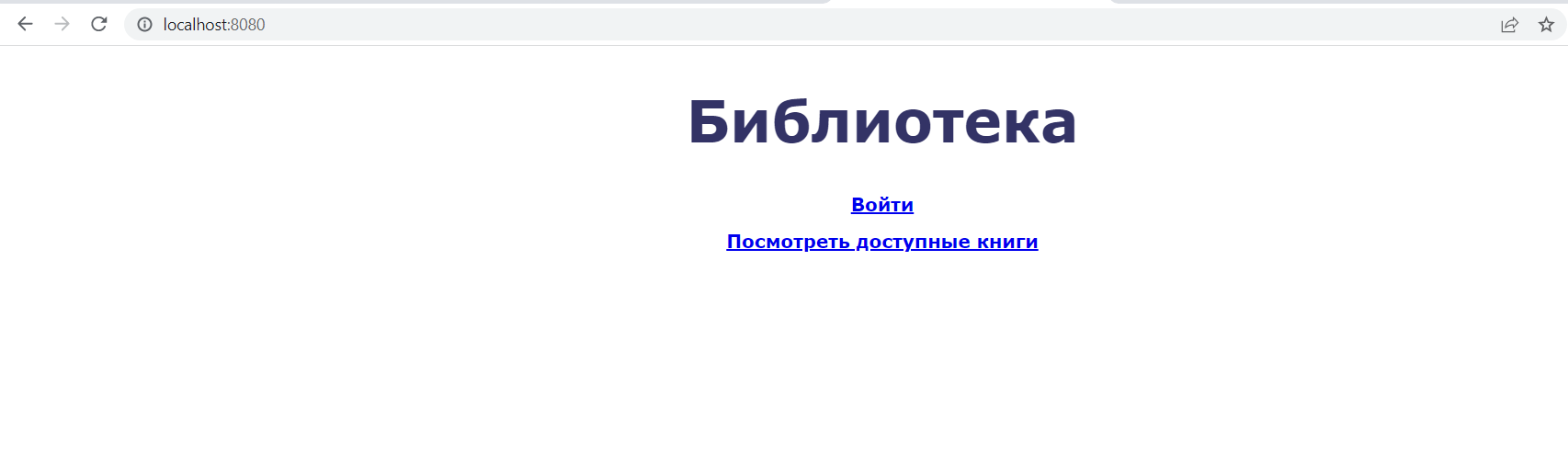


Рисунок 6.1 – Пример работы index.php

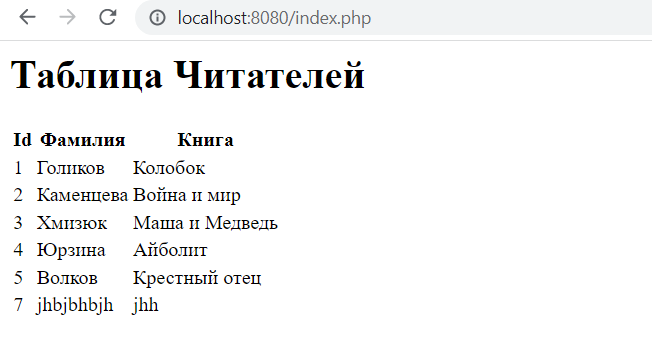


Рисунок 6.2 – Пример работы index.php

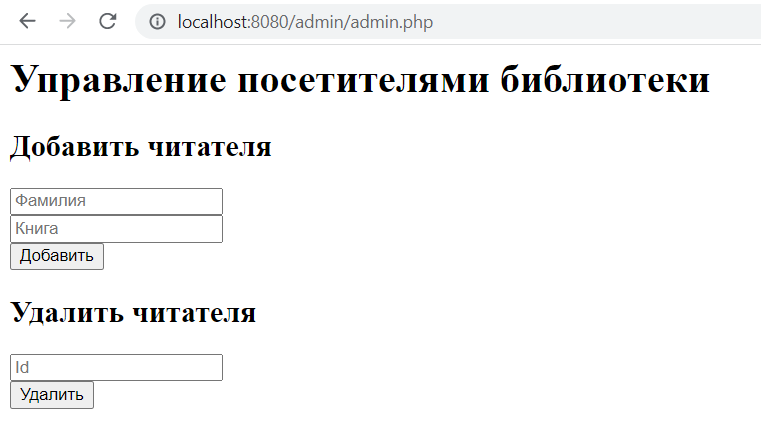


Рисунок 6.3 – Пример работы index.php

Файл Dockerfile содержит установку требуюмых для выполнения упражнений библиотек (Рисунок 7).

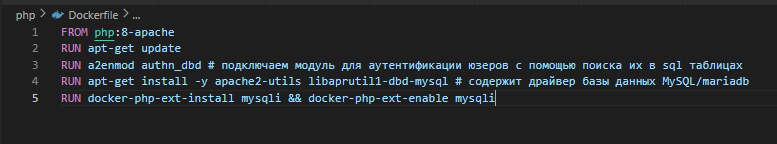


Рисунок 7 – Файл Dockerfile

# 

# Выводы

В ходе работы была создана конфигурация серверного программного обеспечения, состоящего из:

1. Веб-сервера Apache;
2. Сервера Nginx;
3. СУБД MySQL;
4. Языка программирования PHP.

А также были приобретены знания в области создания авторизации с помощью модуля .htaccess

# Ответы на вопросы к практической работе

1. Веб-сервер это 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.
2. Что такое сервер приложения и чем он отличается от веб-сервера?— Веб-сервер включает только веб-контейнер. В то время как сервер приложений включает в себя веб-контейнер, а также контейнер EJB. 2. Веб-сервер полезен или приспособлен для статического контента.
3. Кратко опишите историю развития интернета в рамках развития вебсерверов. — 29 октября 1969 UCLA’s Network Measurement Center, Stanford Research Institute (SRI), University of California-Santa Barbara и University of Utah устанавливают ноды, ставшие первыми известными публичными серверами. Первое Интернет-сообщение “LO” было попыткой студента Чарльза Клайна “войти”(login) на компьютер SRI из Университета.
4. Кратко опишите протокол HTTP - HTTP — протокол прикладного уровня передачи данных, изначально — в виде гипертекстовых документов в формате HTML, в настоящее время используется для передачи произвольных данных.

5) Опишите механизм взаимодействия HTTP-сервера, HTTP-клиента и

пользователя.– HTTP — это клиент-серверный протокол, то есть запросы отправляются какой-то одной стороной — участником обмена (user-agent) (либо прокси вместо него). Чаще всего в качестве участника выступает веб-браузер, но им может быть кто угодно, например, робот, путешествующий по Сети для пополнения и обновления данных индексации веб-страниц для поисковых систем.

Каждый запрос (англ. request) отправляется серверу, который обрабатывает его и возвращает ответ (англ. response). Между этими запросами и ответами как правило существуют многочисленные посредники, называемые прокси, которые выполняют различные операции и работают как шлюзы или кэш, например.

6) Опишите цели и задачи веб-сервера. Главная задача веб сервера принимать HTTP-запросы от пользователей, обрабатывать их, переводить в цифровой компьютерный код. Затем выдавать HTTP-ответы, преобразуя их из миллионов нолей и единичек в изображения, медиа-потоки, буквы, HTML страницы..

7) Опишите технологию SSI. технология позволяющая удобно «собирать» веб-страницы из частей, вставлять в них результаты выполнения CGI-скриптов и придавать страницам прочие элементы динамики.

8) Что такое система управления контентом? сайта, так называемый «движок». Как работают и для чего применяются ЦМС.

9) Верно ли, что сервер приложения умеет работать с протоколом HTTP? Верно.

10) Что такое CGI? стандарт интерфейса, используемого внешней программой для связи с веб-сервером.

11) Как работает система с использованием интерфейс шлюза - CGI? приложение выполняет все необходимые операции и формирует результаты в виде HTML. Сформированный гипертекст возвращается веб-серверу через стандартный поток вывода (stdout)

12) Назовите достоинства и недостатки CGI. FastCGI снимает множество ограничений CGI-программ. Недостаток CGI-программ в том, что они должны быть перезапущены веб-сервером при каждом запросе

13) Что такое FastCGI? - Интерфейс FastCGI — клиент-серверный протокол взаимодействия веб-сервера и приложения

14) Назовите основные отличия CGI от FastCGI. В отличие от CGI, который создает новый процесс для каждого веб-запроса, FastCGI использует непрерывный процесс для обработки серии веб-запросов, которые управляются диспетчером процессов FastCGI, а не веб-сервером.

15) Что такое менеджер процессов? Его суть состоит в организации рабочих процессов с учетом деления на специальные зоны ответственности.

16) Что такое PHP-FPM? является альтернативной реализацией PHP FastCGI с несколькими дополнительными возможностями, обычно используемыми для высоконагруженных сайтов.

17) Что такое Spawn-fcgi? Разделение привилегий без необходимости suid-исполняемого файла или запуска сервера с привилегиями root.

18) Что такое Lighttpd? веб-сервер, разрабатываемый с расчётом на скорость и защищённость, а также соответствие стандартам.

19) Что такое chroot окружение? Chroot-окружение – это системный вызов, который временно перемещает root каталог в новую папку.

20) Опишите механизм взаимодействия серверов с использованием FastCGI. Интерфейс FastCGI — клиент-серверный протокол взаимодействия веб-сервера и приложения, дальнейшее развитие технологии CGI. По сравнению с CGI является более производительным и безопасным.

PHP интерпретатор запускается как независимый сервер, обрабатывающий входящие запросы на исполнение PHP скриптов по протоколу FastCGI, что позволяет ему работать с любым веб-сервером, поддерживающим этот протокол

более производительный и безопасный

вместо того чтобы создавать новые процессы для каждого нового запроса, использует постоянно запущенные процессы для обработки множества запросов

использует Unix Domain Sockets или TCP/IP для связи с сервером

могут быть запущены не только на этом же сервере, но и где угодно в сети

возможна обработка запросов несколькими FastCGI-процессами, работающими параллельно

в кластере должен находиться только FastCGI-процесс, а не целый веб-сервер

обеспечивает дополнительную безопасность, такую как, например, запуск FastCGI-процесса под учётной записью пользователя, отличного от пользователя веб-сервера, а также может находиться в chroot'е, отличном от chroot'а веб-сервера

может быть использован в любом языке, поддерживающем сокеты

21) Опишите процесс выбора встроенного или внешнего менеджера процессов. Зависит от конфигурации сервера, если существует один сервер, который просто является прокси для остальных, то лучше встроенный

22) Что такое интерфейс шлюза? этот сетевой стандарт позволяет Web-серверу запускать любую программу и передавать Web-браузеру данные в виде текстовой или двоичной (графической, звуковой) информации.

23) Что такое SCGI? протокол по взаимодействию приложений с веб серверами, разработанный как альтернатива Common Gateway Interface

24) Что такое PCGI? PCGI — библиотека к языку программирования Perl для работы с интерфейсом CGI

25) Что такое PSGI? PSGI или Perl Web Server Gateway Interface - спецификация, предназначенная для отделения среды веб-сервера от кода веб-фреймворка

26) Что такое WSGI? WSGI — стандарт взаимодействия между Python-программой, выполняющейся на стороне сервера, и самим веб-сервером, например Apache.

27)Опишите механизм взаимодействия серверов Apache и PHP.

Apache обычно обслуживает файлы, извлекая файл и отправляя поток вниз по HTTP-соединению. Однако с PHP Apache извлекает файл, передает его в двоичный файл PHP и отправляет выход поток из команды вниз по HTTP-соединению.

28)Опишите преимущества веб-сервера Apache.

• Высокий уровень надежности

• Гибкие настройки

• Свободный доступ к программе

• Регулярные обновления и патчи

• Удобство и легкость настройки

29)Опишите недостатки веб-сервера Apache.

• Проблемы с производительностью на высоконагруженных сайтах

• Большое кол-во параметров настройки может привести к уязвимости в конфигурации

• Некоторая вероятность наличия вредоносного кода в модулях от независимых разработчиков

30)Опишите архитектуру веб-сервера Apache.

Ядро Apache включает в себя основные функциональные возможности, такие как обработка конфигурационных файлов, протокол HTTP и система загрузки модулей.

Система конфигурации Apache, основанная на текстовых конфигурационных файлах.

Apache имеет встроенный механизм виртуальных хостов. Он позволяет полноценно обслуживать на одном IP-адресе множество сайтов (доменных имён), отображая для каждого из них собственное содержимое.

31)Опишите функции ядра веб-сервера Apache.

Основные функции ядра:

• Передача данных по HTTP

• Обработка файлов

• Загрузка и поддержка модулей

32)Опишите конфигурацию веб-сервера Apache.

Конфигурацию Apache можно разделить на три основных уровня:

• Конфигурация сервера

• Конфигурация виртуального хоста

• Конфигурация уровней каталога

33)Что такое URI, URL и чем они различаются.

URI – имя и адрес ресурса в сети, включает в себя URL и URN

URL – адрес ресурса в сети, определяет местонахождение и способ обращения к нему

URN – имя ресурса в сети, определяет только название ресурса, но не говорит как к нему подключиться

# Ссылка на удалённый репозиторий проекта

<https://github.com/Rpqshka/University-5-Sem/tree/main/РСЧИР/Практика%203>

1. **Список использованных источников**
2. Видео “Введение в Докер” на английском языке от создателя: Introduction to Docker (https://[www.youtube.com/watch?v=Q5POuMHxW-0)](http://www.youtube.com/watch?v=Q5POuMHxW-0))
3. Статья о назначении докера простыми словами: https://habr.com/ru/post/309556/
4. Более сложная и подробная статья про докер: https://habr.com/ru/post/277699/