|  |  |  |
| --- | --- | --- |
|  |  |  |
| 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 №3**

по дисциплине «Разработка серверных частей интернет-ресурсов»

**Тема практической работы:** Настройка сервера по заданным характеристикам, размещение проекта

**Студент группы** ИКБО-16-20 Косогоров Кирилл Станиславович

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

(подпись студента)

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

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

(подпись руководителя)

Работа представлена «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_ 2022 г.

Допущен к работе «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2022 г.

Москва 2022

Оглавление

[1. ЦЕЛЬ РАБОТЫ 3](#_Toc126012273)

[2. ХОД РАБОТЫ 3](#_Toc126012274)

[3. ВЫВОДЫ 8](#_Toc126012275)

[4. ОТВЕТЫ НА ВОПРОСЫ 9](#_Toc126012276)

[5. Ссылка на репозиторий GitHub 14](#_Toc126012277)

[6. СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ 15](#_Toc126012278)

1. **ЦЕЛЬ РАБОТЫ**

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. Схематично предполагаемый алгоритм работы изображен на рисунке 1.

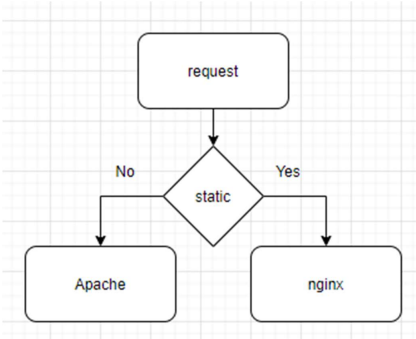


Рисунок 1 – Схематичное изображение предполагаемой функциональности

Предполагается, что сервер nginx будет отображать статический контент, а apache динамический и в связке мы получим быстродейственную и эффективную систему. Также по необъяснимым обстоятельствам следует изменить root директории с базовых на другие. Для доступа к администрированию предусмотреть базовую авторизацию и аутентификацию с применением httaccess и хранением пользователей в БД (без использования htpasswd). Для тестирования данной конфигурации предполагается создать тестовое веб-приложение.

**2. ХОД РАБОТЫ**

Тема: “Магазин дайверского оборудования”.

Для начала создадим docker-compose.yml, который будет содержать инструкции, необходимые для запуска и настройки сервисов (Листинг 1).

Листинг 1 – файл docker-compose.yml

version: '3'

services:

  apache:

    build: ./apache

    volumes:

      - ./apache/src:/var/www/html/

      - ./apache/000-default.conf:/etc/apache2/sites-available/000-default.conf

    environment:

      - DB\_HOST=my-sql

      - DB\_PORT=3306

      - DB\_NAME=appDB

      - DB\_USER=user

      - DB\_PASSWORD=password

  my-sql:

    image: mysql:latest

    volumes:

      - ./db:/docker-entrypoint-initdb.d

      - db\_volume:/var/lib/mysql

    environment:

      - MYSQL\_ROOT\_PASSWORD=root

  nginx:

    build: ./nginx

    volumes:

      - ./nginx/src:/etc/nginx/html/

      - ./nginx/nginx.conf:/etc/nginx/nginx.conf

    ports:

      - 80:80

volumes:

  db\_volume:

Далее создадим директорию apache. В ней будут храниться файлы для отображения динамической информации, а также файлы dockerfile и 000-default.conf (Листинги 2-5).

Листинг 2 – файл 000-default.conf

<VirtualHost \*:80>

    ServerName test

    DBDriver mysql

    DBDParams "host=${DB\_HOST} port=${DB\_PORT} dbname=${DB\_NAME} user=${DB\_USER} pass=${DB\_PASSWORD}"

    <Directory "/var/www/html">

        AuthType Basic

        AuthName "Restricted Content"

        AuthBasicProvider dbd

        Require valid-user

        AuthDBDUserPWQuery "SELECT password FROM users WHERE login = %s"

    </Directory>

</VirtualHost>

Листинг 3 – файл prices.php

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Цены</title>

</head>

<body>

    <h1>Цены Оборудования</h1>

    <table>

        <tr>

            <th>Название</th>

            <th>Цена</th>

        </tr>

        <?php

        $mysqli = new mysqli("my-sql", "user", "password", "appDB");

        $result = $mysqli->query("SELECT name, price FROM products");

        foreach ($result as $row) {

            echo "<tr><td>{$row['name']}</td><td>{$row['price']}</td></tr>";

        }

        ?>

    </table>

</body>

</html>

Листинг 4 – файл products.php

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Ассортимент Оборудования</title>

</head>

<body>

    <h1>Оборудование</h1>

    <ol>

        <?php

        $mysqli = new mysqli("my-sql", "user", "password", "appDB");

        $result = $mysqli->query("SELECT name FROM products");

        foreach ($result as $row) {

            echo "<li>{$row['name']}</li>";

        }

        ?>

    </ol>

</body>

</html>

Листинг 5 – Dockerfile в директории Apache

FROM php:apache

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

    a2enmod authn\_dbd && \

    apt-get update && \

    apt-get install -y libaprutil1-dbd-mysql

В директории db содержится файл init.sql (Листинг 6).

Листинг 6 – файл init.sql

C CREATE DATABASE IF NOT EXISTS appDB;

CREATE USER IF NOT EXISTS 'user'@'%' IDENTIFIED BY 'password';

GRANT SELECT,UPDATE,INSERT ON appDB.\* TO 'user'@'%';

FLUSH PRIVILEGES;

USE appDB;

CREATE TABLE IF NOT EXISTS users (

  ID INT(11) NOT NULL AUTO\_INCREMENT,

  login VARCHAR(20) NOT NULL,

  password VARCHAR(40) NOT NULL,

  PRIMARY KEY (ID)

);

CREATE TABLE IF NOT EXISTS products (

  ID INT(11) NOT NULL AUTO\_INCREMENT,

  name VARCHAR(40) NOT NULL,

  price INTEGER,

  PRIMARY KEY (ID)

);

INSERT INTO users (login, password)

SELECT \* FROM (SELECT 'kirill', '{SHA}QL0AFWMIX8NRZTKeof9cXsvbvu8=') AS tmp

WHERE NOT EXISTS (

    SELECT login FROM users WHERE login='kiril' AND password='123'

) LIMIT 1;

INSERT INTO products (name, price)

SELECT \* FROM (SELECT 'Гидрокостюм Aqualung Diveflex', 23400) AS tmp

WHERE NOT EXISTS (

    SELECT name FROM products WHERE name = 'Гидрокостюм Aqualung Diveflex' AND price = 23400

) LIMIT 1;

INSERT INTO products (name, price)

SELECT \* FROM (SELECT 'Жилет компенсатор Prodive PRO 1000 den', 36000) AS tmp

WHERE NOT EXISTS (

    SELECT name FROM products WHERE name = 'Жилет компенсатор Prodive PRO 1000 den' AND price = 36000

) LIMIT 1;

INSERT INTO products (name, price)

SELECT \* FROM (SELECT 'Перчатки неопреновые 2,5 мм, кевлар', 2450) AS tmp

WHERE NOT EXISTS (

    SELECT name FROM products WHERE name = 'Перчатки неопреновые 2,5 мм, кевлар' AND price = 2450

) LIMIT 1;

В директории nginx хранятся файлы, отображающие статический контент и nginx.conf, в которой указана конфигурация для nginx (Листинг 7) и dockerfile (Листинг 8).

Листинг 7 – файл nginx.conf

events {}

http {

    server {

        listen 80;

        location ~ \.php{

            proxy\_pass http://apache;

        }

    }

}

Листинг 8 – Dockerfile в директории nginx

FROM nginx

Протестируем нашу серверную конфигурацию (Рисунки 1-5).

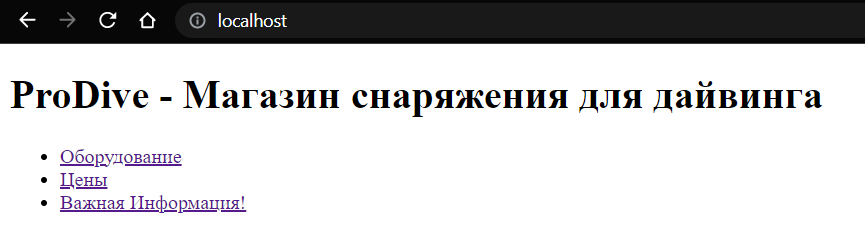


Рисунок 1 – Главная страница

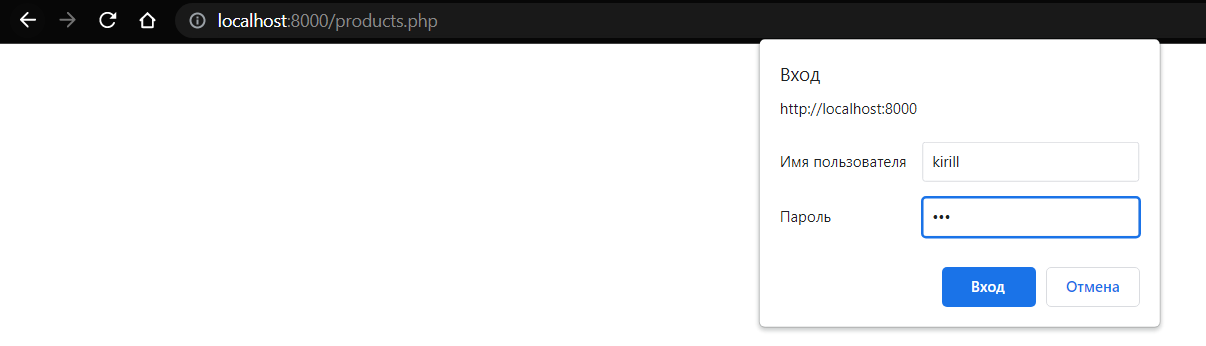


Рисунок 2 – Страница с блоком авторизации



Рисунок 3 – Страница с оборудованием

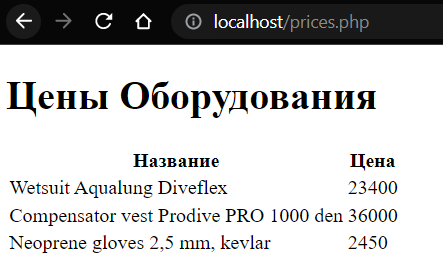


Рисунок 4 – Страница с оборудованием и ценами

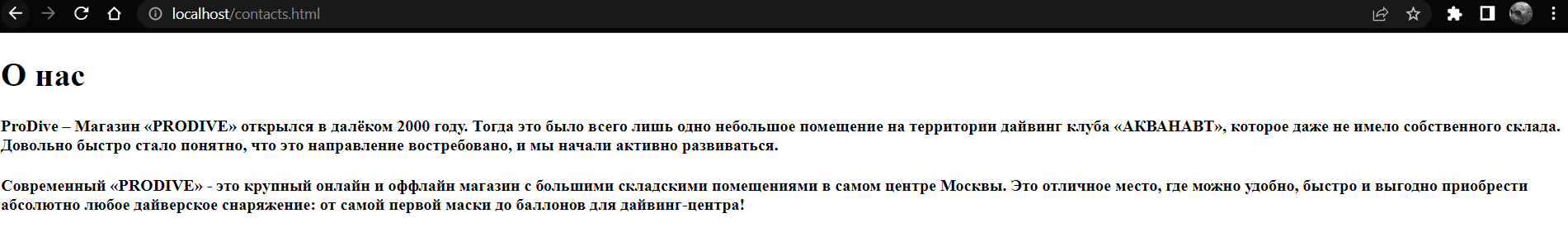


Рисунок 5 – Страница “О нас”

**3. ВЫВОДЫ**

В ходе выполнения работы была разработана серверная конфигурация, состоящей из связки apache+nginx+php+База данных. В данной конфигурации было создано 3 элемента(контейнера), 2 веб-страницы со статическим контентом и две веб-страницы с динамическим контентом, взятым из базы данных.

**4. ОТВЕТЫ НА ВОПРОСЫ**

**1. Что такое веб-сервер?**

Веб-сервер – компьютерное ПО и базовое оборудование, которое принимает запросы через HTTP.

**2. Что такое сервер приложения и чем он отличается от веб-сервера?**

Сервер приложений – программа сервера, работающая в распределенной сети и обеспечивающая среду выполнения для прикладных программ. Сервер приложений взаимодействует с веб-сервером и возвращает динамический, настраиваемый ответ на клиентский запрос.

**3. Кратко опишите протокол HTTP.**

Протокол HTTP реализован по клиент-серверной технологии и работает по принципу запрос-ответ без сохранения состояния.

**4. Опишите механизм взаимодействия HTTP-сервера, HTTP-клиента и пользователя.**

В процессе работы HTTP-сервера получает запрос клиента, обрабатывает его и либо выдает ему запрашиваемый ресурс, либо сообщает, что это сделать невозможно. Работа HTTP-сервер показана на рис. 4.1.

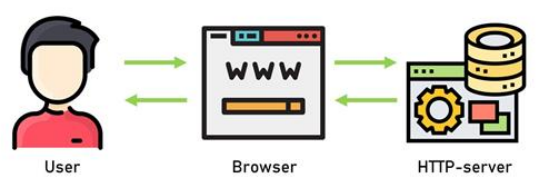


Рисунок 4.1 – Схема взаимодействия клиента и HTTP-сервера.

**5. Опишите цели и задачи веб-сервера.**

Основные цели и задачи такого веб-сервера сводятся к обработке HTTP-запросов клиента и возвращении пользователю результатов этой обработки. Заметим, что Веб-сервер не способен к самостоятельной генерации контента, он может обрабатывать только статическое содержимое.

**6. Опишите технологию SSI.**

SSI (от английского Server Side Includes – включение на стороне сервера) – это язык, разработанный для динамического создания и «сборки» веб-страниц на сервере из отдельных составных частей и выдачи клиенту полученного HTML-документа.

Технология SSI дает возможность добавления на веб-страницы дополнительное динамическое содержимое, такое, например, как актуальная дата на странице. Технология дополнительно дает возможность выполнять несложные условия и работать с переменными.

**7. Что такое система управления контентом?**

Информационная система или компьютерная программа, используемая для обеспечения и организации совместного процесса создания, редактирования и управления содержимым.

**8. Верно ли, что сервер приложения умеет работать с протоколом HTTP?**

Сервер приложений не работает с прото- 82 колом HTTP и не обрабатывает пользовательские запросы. Все эти действия по-прежнему выполняет веб-сервер.

**9. Что такое CGI?**

CGI (от английского Common Gateway Interface) или интерфейс общего шлюза, представляет собой стандарт интерфейса, используемого внешней программой для связи с веб-сервером.

**10. Как работает система с использованием интерфейс шлюза - CGI?**

Рассмотрим процесс работы такой системы подробнее. Получив запрос от браузера пользователя, веб-сервер определяет, что запрошено динамическое содержимое и формирует специальный запрос, который через интерфейс CGI передается веб-приложению. При его получении приложение запускается и выполняет запрос, результатом которого служит HTML-код динамически сформированной страницы, который передается назад веб-серверу, после чего приложение завершает свою работу.

**11. Назовите достоинства и недостатки CGI.**

К достоинствам CGI можно отнести языковую и архитектурную независимость: CGI-приложение может быть написано на любом языке и одинаково хорошо работать с любым веб-сервером. Учитывая простоту и открытость стандарта, это привело к активному развитию веб-приложений. Однако, кроме достоинств, CGI обладает и существенными недостатками. Основной из них − высокие накладные расходы на запуск и остановку процесса, что влечет за собой повышенные требования к аппаратным ресурсам и невысокую производительность. Использование стандартных потоков ввода вывода ограничивает возможности масштабирования и обеспечения высокой доступности, так как требует, чтобы веб-сервер и сервер приложений находились в пределах одной операционной системы (ОС).

**12. Что такое FastCGI?**

FastCGI представляет собой клиент-серверный протокол для взаимодействия веб-сервера и сервера приложений, обеспечивающий высокую производительность и безопасность.

**13. Назовите основные отличия CGI от FastCGI.**

FastCGI – развитие технологии CGI, ее более производительная и современная версия. В отличие от CGI, в данном режиме для каждого скрипта не запускается отдельный процесс, благодаря чему меньше расходуются ресурсы. Особенности: работает несколько медленнее, чем модуль Apache и PHP-FPM, но быстрее CGI.

**14. Что такое менеджер процессов?**

Для управления FastCGI-процессами и распределением нагрузки служат менеджеры процессов, они могут быть как частью веб-сервера, так и отдельными приложениями.

**15. Что такое PHP-FPM?**

PHP-FPM первоначально был набором патчей к PHP от Андрея Нигматулина, решавший ряд вопросов управления FastCGI-процессами, начиная с версии 5.3 является частью проекта и входит в поставку PHP. PHP-FPM умеет динамически управлять количеством процессов PHP в зависимости от нагрузки, перезагружать пулы без потери запросов, аварино перезапускать сбойные процессы и представляет собой менеджер с расширенным функционалом.

1. **Что такое Spawn-fcgi?**

spawn-fcgi используется для запуска удаленных и локальных [FastCGI](http://vds-admin.ru/" \o "FastCGI) процессов. Почему лучше использовать spawn-fcgi:

Разделение привилегий без необходимости *suid*-исполняемого файла или запуска сервера с привилегиями root.

Возможность отдельного перезапуска как *fastcgi* приложения, так и сервера

Возможность запуска в изолированном окружении ( *chroot* ).

Ваше *FastCGI* приложение не зависит от используемого веб сервера что дает возможность, использовать различные веб серверы.

1. **Что такое Lighttpd?**

lighttpd — [веб-сервер](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80), разрабатываемый с расчётом на скорость и защищённость, а также соответствие стандартам.

1. **Что такое chroot окружение?**

chroot — операция изменения [корневого каталога](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D1%80%D0%BD%D0%B5%D0%B2%D0%BE%D0%B9_%D0%BA%D0%B0%D1%82%D0%B0%D0%BB%D0%BE%D0%B3) в [Unix-подобных](https://ru.wikipedia.org/wiki/Unix-%D0%BF%D0%BE%D0%B4%D0%BE%D0%B1%D0%BD%D0%B0%D1%8F_%D0%BE%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0" \o "Unix-подобная операционная система) [операционных системах](https://ru.wikipedia.org/wiki/%D0%9E%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0).

1. **Опишите механизм взаимодействия серверов с использованием FastCGI.**

 FastCGI-процессы используют [Unix Domain Sockets](https://ru.wikipedia.org/wiki/Unix_domain_socket" \o "Unix domain socket) или [TCP/IP](https://ru.wikipedia.org/wiki/TCP/IP) для связи с сервером.

1. **Что такое SCGI?**

SCGI-это протокол, который определяет связь между веб - сервером и сервером приложений.

1. **Что такое PCGI?**

PCGI (Perl Common Gateway Interface) — библиотека к языку программирования Perl для работы с интерфейсом CGI (Common Gateway Interface). Библиотека позволяет с высокой скоростью обрабатывать входящий поток данных. Основное достоинство заключается в том, что библиотека позволяет совершенно безопасно принимать сколь угодно крупные объёмы данных, при этом очень экономично потребляя оперативную память.

1. **Что такое PSGI?**

PSGI или Perl Web Server Gateway Interface - спецификация, предназначенная для отделения среды веб-сервера от кода веб-фреймворка. PSGI не является программным интерфейсом (API) для веб-приложений.

1. **Что такое WSGI?**

WSGI ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) Web Server Gateway Interface) — стандарт взаимодействия между [Python](https://ru.wikipedia.org/wiki/Python" \o "Python)-программой, выполняющейся на стороне сервера, и самим [веб-сервером](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80).

1. **Опишите механизм взаимодействия серверов Apache и PHP.**

Запрос от клиента поступает на сервер, если запрашиваемый документ статический и находится на сервере, то сервер просто отдает его. Если же документ(тип документа) является динамическим, то будет поднят новый процесс, который через интерфейс CGI запустит php-скрипт, а результат его работы вернет клиенту.

1. **Опишите преимущества веб-сервера Apache.**

Бесплатное программное обеспечение как для личного использования, так и для коммерческого;

гибкость программы. Открытый исходный код позволяет подстроить программу под специфические нужды пользователя;

поддержка многих языков программирования (PHP, Python, Ruby, Perl, ASP, Tcl), что дает доступ к различным серверным ПО;

удобная и надежная конфигурация системы. Приложение легко настраивается через текстовые конфигурационные файлы и может быстро масштабироваться;

Apache хорошо работает со многими системами управления контентом ([CMS](https://help.reg.ru/hc/ru/articles/4408054695313)), такими как WordPress, Joomla, Drupal;

поддерживает технологии FastCGI и CGI;

может использовать СУБД для аутентификации пользователей;

большая база инструкций. Так как продукт уже давно на рынке, разработчики и пользователи уже написали множество инструкций под разные ситуации.

1. **Опишите недостатки веб-сервера Apache.**

При высоком трафике снижается производительность. На высокопосещаемых сайтах может значительно снижаться скорость работы программы. Это связано с особенностью работы сервиса. Каждый пользовательский запрос обрабатывается отдельным потоком. Когда потоки заканчиваются, остальные операции ждут своей очереди. Если планируется большое количество посещений сайта, то лучше использовать [Nginx](https://help.reg.ru/hc/ru/articles/4408047693329" \t "_blank);

много параметров настройки. Если допустить ошибку при настройке конфигурации, это может привести к появлению уязвимостей во всей системе;

риск заражения вирусом. Создавать модули могут не только добросовестные разработчики, но и злоумышленники. В модуль может быть внедрён вредоносный код. Устанавливать модули стоит только от проверенных разработчиков.

1. **Опишите архитектуру веб-сервера Apache.**

Apache состоит из: ядра, системы конфигурации, многопроцессорной модели, системы модулей, механизм виртуальных хостов.

1. **Опишите функции ядра веб-сервера Apache.**

Apache состоит из ядра и модулей. Ядро выполняет основные функции: обработка конфигурационных файлов, работа с протоколом HTTP, система загрузки модулей.

1. **Что такое URI, URL и чем они различаются.**

URI является либо URL, либо URN, либо одновременно обоими.

URL — это URI, который, помимо идентификации ресурса, предоставляет ещё и информацию о местонахождении этого ресурса. А URN — это URI, который только идентифицирует ресурс в определённом пространстве имён (и, соответственно, в определённом контексте), но не указывает его местонахождение.

**5. Ссылка на репозиторий GitHub**

<https://github.com/pe4enka2515/DCPIR/tree/main/Pr3>

**6. СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ**

1. PHP: Руководство по PHP – Manual. – URL: <https://www.php.net/manual/ru/index.php> (дата обращения 14.10.2022)
2. Apache. – URL: https://www.apache.org (дата обращения 16.10.2022)

NGINX – URL:  [https://www.nginx.com l](https://handyhost.ru/help/term/chto-takoe-phpini.html) (дата обращения 16.10.2