|  |  |  |
| --- | --- | --- |
|  |  |  |
| 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 № 4**

по дисциплине «Разработка серверных частей интернет-ресурсов»

**Тема практической работы:** Конфигурация NGINX

|  |  |
| --- | --- |
| Студент группы ИКБО-32-21 | Быченков А.К. |

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

(подпись студента)

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

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

(подпись руководителя)

Работа представлена «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2023 г.

Допущен к работе «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2023 г.

Москва 2023

**Цель работы:** получить навыки настройки конфигурации NGINX.

**Задание**

Вам необходимо создать конфигурацию docker-compose для вашего веб-сервера.

Docker контейнер веб сервера должен включать:

* ОС Linux
* Веб-сервер Apache

Docker контейнер базы данных должен включать:

* ОС Linux
* Базу данных Mysql

Docker контейнер NGINX должен включать:

* ОС Linux
* NGINX

Docker-compose:

* Docker-контейнеры, созданные ранее
* Примонтированные тома
* Настройки портов
* Связь между контейнерами
* Переменные внешнего окружения
* Настройки базы данных

В рамках данного курса вам предлагается создать один большой проект маркетплейса, некоторые из практик, например, как эта, содержат кусок того проекта.

1. Создать 2 статичные html страницы для маркетплейса, где на одной странице будет информация по доставке, а на другой будут контакты и реквизиты вашего маркетплейса. (Эти страницы будут в финальном проекте)

2. Нужно сделать таким образом, чтобы статические страницы хранились в контейнере вместе с сервисом обратного прокси-сервера, а также настроить NGINX так, чтобы он мог их выдавать по запросу.

3. Создать 2 статичные html страницы. Динамические страницы могут быть взяты из предыдущих практических заданий или же написаны с нуля. Однако, необходимо сделать так, чтобы запросы и ответы на динамические страницы проходили через NGINX.

**Ход работы**

Была написана конфигурация docker-compose для запуска apache, mysql, nginx, листинг 1.

Листинг 1 – compose.yaml

services:

apache:

build: .

restart: always

expose:

- 80

environment:

MYSQL\_HOST: db

env\_file:

- .env

volumes:

- ./src:/var/www/html

depends\_on:

- db

db:

image: mysql:latest

restart: always

environment:

MYSQL\_DATABASE: ${MYSQL\_DATABASE}

MYSQL\_USER: ${MYSQL\_USER}

MYSQL\_PASSWORD: ${MYSQL\_PASSWORD}

MYSQL\_ROOT\_PASSWORD: ${MYSQL\_ROOT\_PASSWORD}

expose:

- 3306

volumes:

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

- ./mysql:/var/lib/mysql

nginx:

image: nginx:latest

restart: always

ports:

- 80:80

volumes:

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

- ./static:/usr/share/nginx/html

depends\_on:

- apache

Конфигурация NGINX показана на листинге 2.

Листинг 2 – nginx.conf

events {}

http {

charset UTF-8;

server {

listen 80;

location / {

root /usr/share/nginx/html;

}

location ~ \.php$ {

proxy\_pass http://apache;

proxy\_redirect http://apache /;

proxy\_set\_header Host $host;

proxy\_set\_header X-Real-IP $remote\_addr;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

proxy\_set\_header X-Forwarded-Host $server\_name;

}

}

}

Были созданы две статические HTML-страницы, которые помещены в директорию /usr/share/nginx/html. Их отдает сам NGINX. Все запросы, адрес которых заканчивается на .php, перенаправляются на Apache-сервер.

Тестирование работы NGINX показано на рисунках 1, 2, 3, 4.

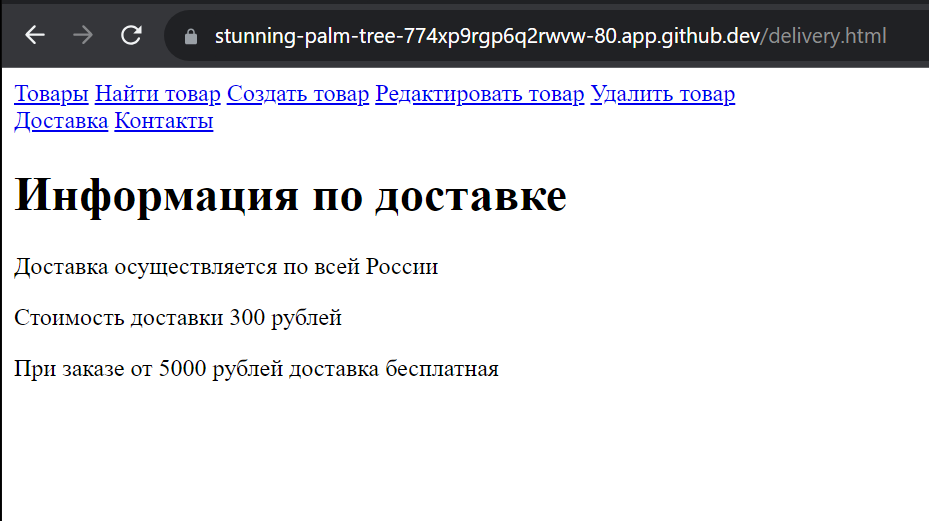


Рисунок 1 – Пример первой статической HTML-страницы

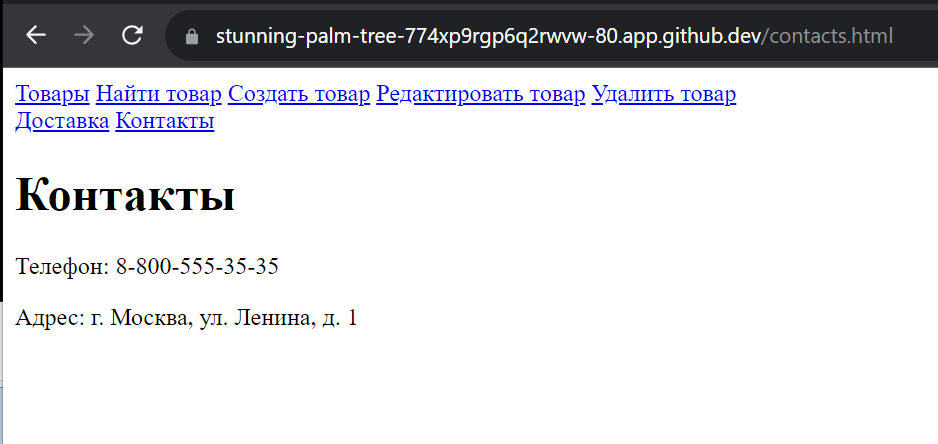


Рисунок 2 – Пример второй статической HTML-страниц

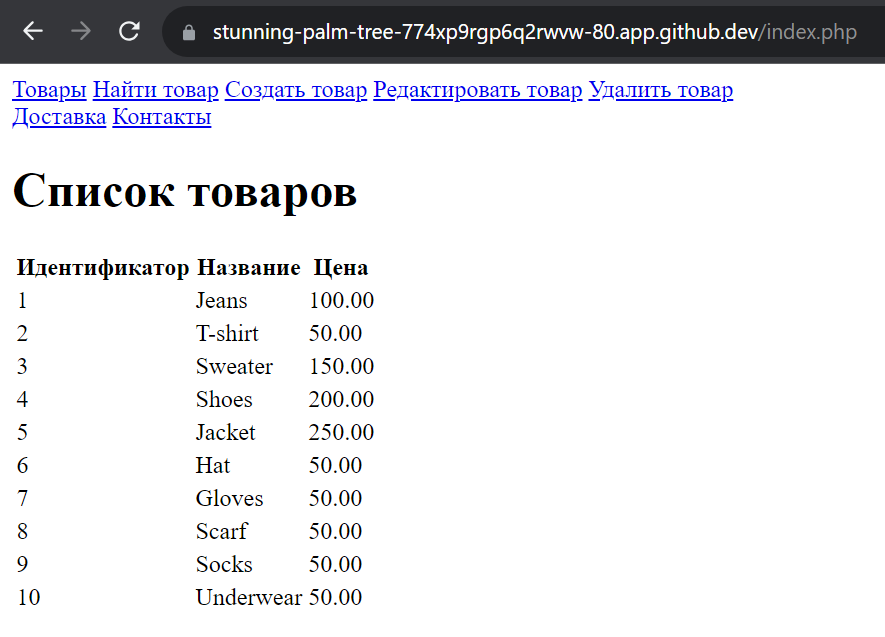


Рисунок 3 – Пример первой динамической страницы

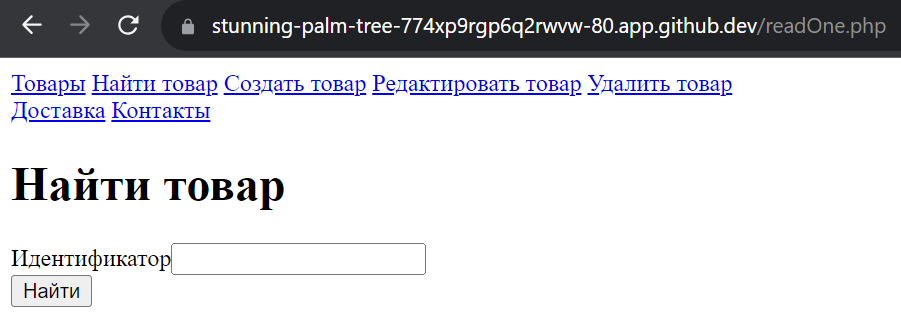


Рисунок 4 – Пример второй динамической страницы

**Вывод**

В ходе выполнения практической работы были получены навыки настройки конфигурации NGINX.

**Ответы на вопросы**

1. **Что такое обратный прокси сервер**

Обратный прокси-сервер (reverse proxy server) — это сервер, который принимает запросы от клиентов (как правило, браузеров) и передает их на один или несколько бэкенд-серверов (серверов приложений или других ресурсов), затем принимает ответы от бэкенд-серверов и передает их клиентам. Обратные прокси-серверы используются для балансировки нагрузки, защиты от атак, кеширования и улучшения производительности.

1. **Как можно использовать NGINX**

NGINX — это высокопроизводительный веб-сервер и прокси-сервер, который может быть использован для различных целей, включая:

Сервер статических файлов: NGINX может обслуживать статические файлы, такие как HTML, CSS, JavaScript и изображения, без использования дополнительных серверов.

Обратный прокси: NGINX может работать как обратный прокси, направляя запросы на бэкенд-сервера, что позволяет балансировать нагрузку и улучшать безопасность.

Кеширование: NGINX поддерживает кеширование, что может увеличить скорость обработки запросов и снизить нагрузку на бэкенд-серверы.

1. **Как работает кэширование в NGINX**

Кеширование в NGINX работает путем сохранения копий ответов от бэкенд-сервера на определенный период времени. Это позволяет NGINX быстро обслуживать запросы клиентов, не обращаясь к бэкенд-серверу, если запрос уже был обработан ранее. Настройка кеширования в NGINX включает следующие этапы:

* Определение кэша и его параметров в конфигурации NGINX.
* Настройка правил кеширования для определенных URL или запросов.
* Конфигурирование времени жизни кэша и условий его обновления.
* Использование директивы proxy\_cache для настройки места хранения кеша.

1. **Как в конфигурации NGINX добавить несколько веб-серверов динамических страниц**

Для добавления нескольких веб-серверов динамических страниц в конфигурацию NGINX, вы можете использовать проксирование на разные бэкенд-серверы с помощью директивы location. Пример конфигурации для двух разных бэкенд-серверов может выглядеть так:

**Ссылка на удаленный репозиторий проекта**

<https://github.com/descenty/server-side-5-sem>

**Список использованной литературы**

1. Официальная документация NGINX: https://nginx.org/ru/
2. Статья по настройке NGINX в качестве обратного прокси сервера: https://help.reg.ru/support/servery-vps/oblachnyye-servery/ustanovka-programmnogo-obespecheniya/kak-nastroit-nginx-v-kachestve-obratnogo-proksi-dlya-apache
3. Сбор статей по настройке NGINX: https://www.dmosk.ru/categories.php?object=nginx