|  |  |  |
| --- | --- | --- |
|  |  |  |
| 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-30-20 Патина В.С.

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

(подпись студента)

**Руководитель практической работы** преподаватель Благирев М.М.

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

(подпись руководителя)

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

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

Москва 2022

**СОДЕРЖАНИЕ**

[Постановка задачи 1](#_Toc117875019)

[Выполнение работы 1](#_Toc117875020)

[Ссылка на удаленный репозиторий 9](#_Toc117875021)

[Ответы на вопросы к практической: 9](#_Toc117875022)

[ВЫВОДЫ 19](#_Toc117875023)

[СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ 20](#_Toc117875024)

Постановка задачи

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.

Выполнение работы

Создадим рабочую директорию со структурой папок, представленную на Рисунке 1.

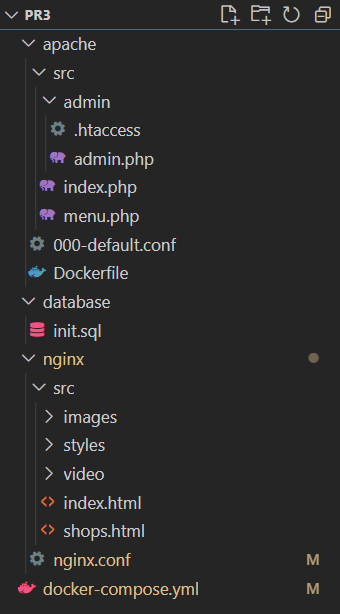


Рисунок 1 – Рабочая директория со структурой папок и файлов

Содержимое файла docker-compose.yml представлено на рисунке 2.

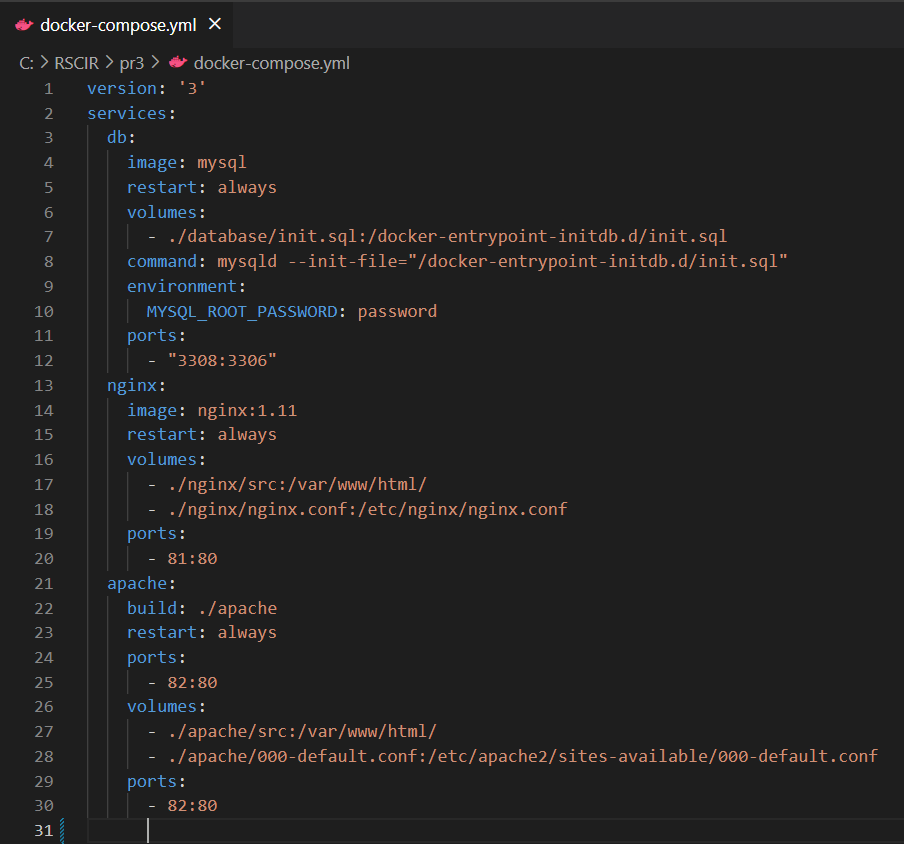


Рисунок 2 - Содержимое файла docker-compose.yml

Была создана конфигурация сервера nginx, содержимое файла nginx.conf представлено на рисунке 3.

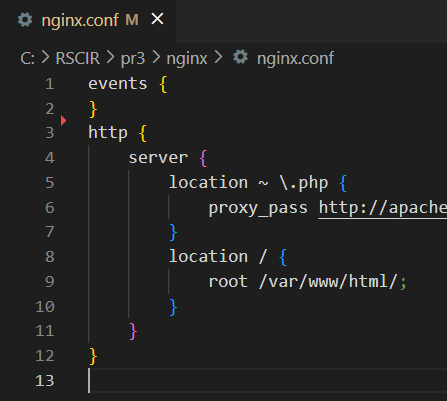


Рисунок 3 - Файл конфигурации сервера nginx

Был создан файл для создания схем и инициализации базы данных приложения, содержимое файла init.sql преставлено на рисунке 4.

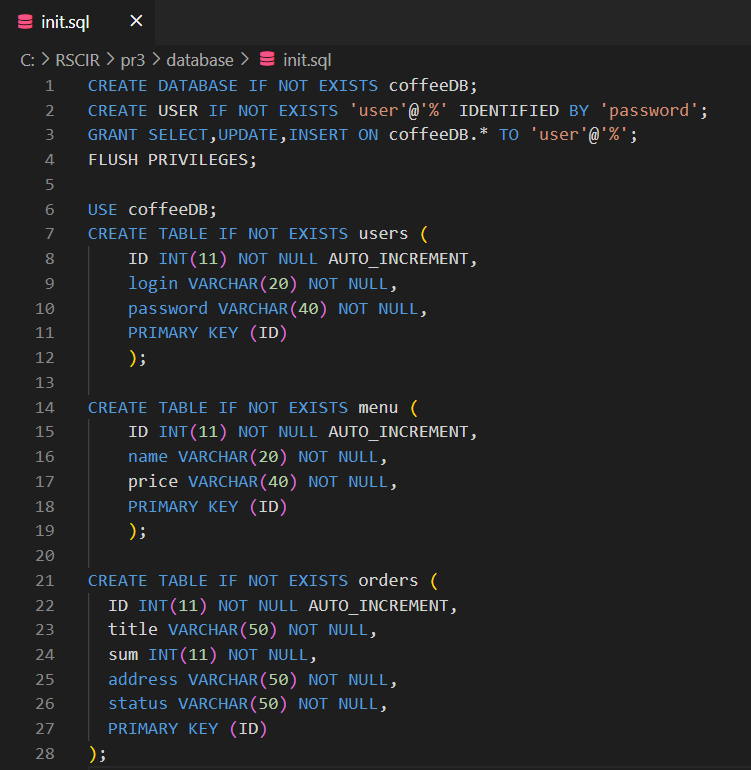


Рисунок 4 - Содержимое файла init.sql

Был написан dockerfile образа apache.

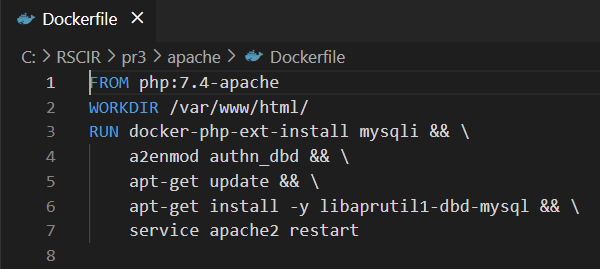


Рисунок 5 - Dockerfile для apache

Была создана конфигурация сервера apache, содержимое файла 000-default.conf представлено на рисунке 5.

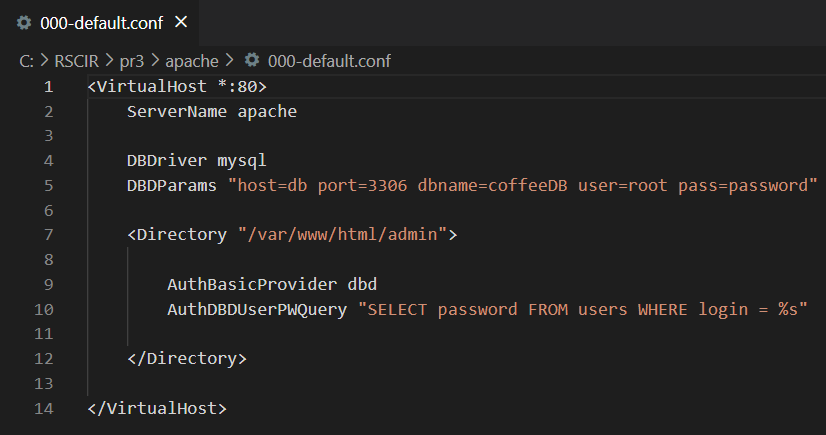


Рисунок 6 - Файл конфигурации сервера apache

Были созданы страницы для отображения контента из базы данных, содержимое php-файлов представлено на рисунках 7, 8.

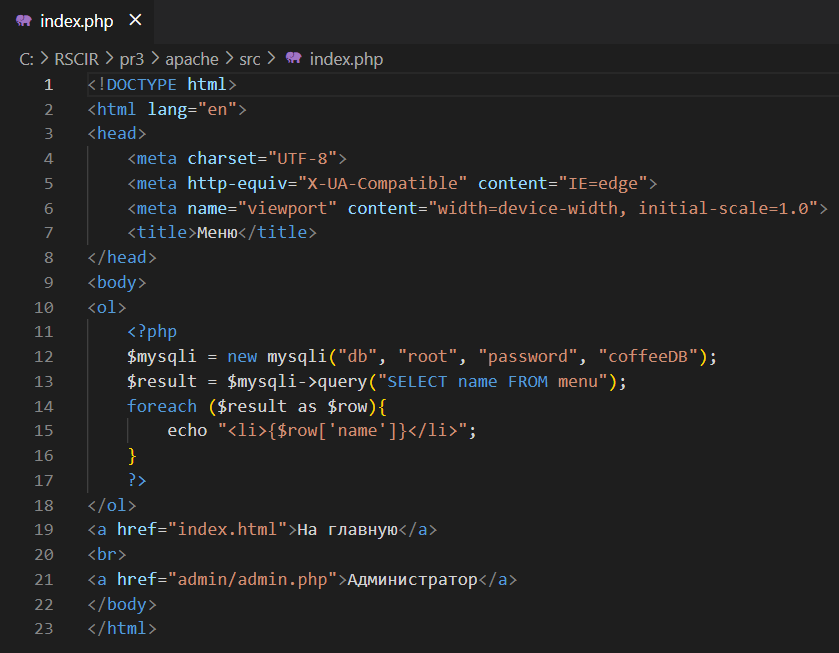


Рисунок 7 - Содержимое файла index.php

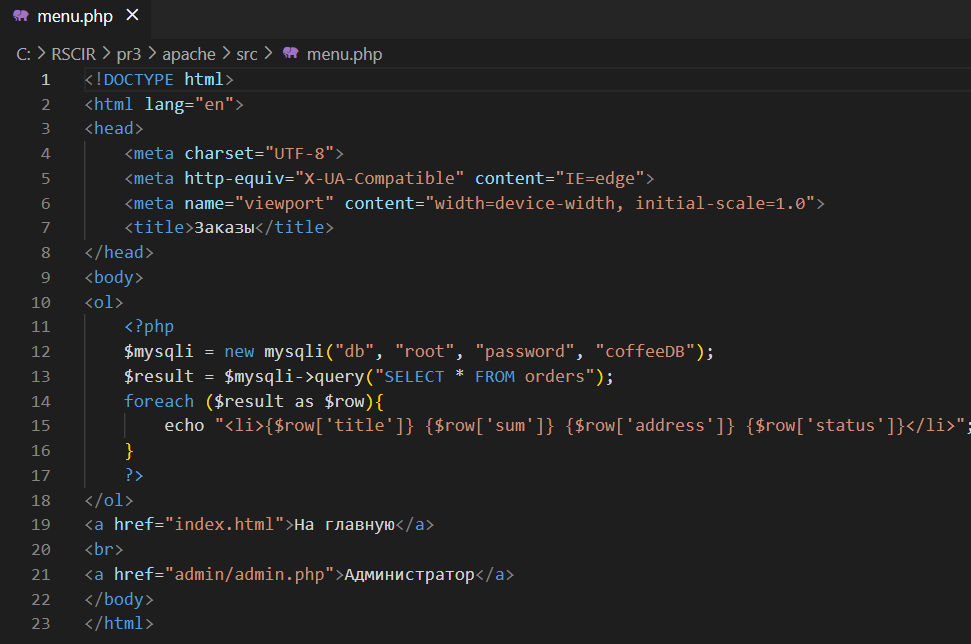


Рисунок 8 - Содержимое файла menu.php

Была создана страница для администраторов, доступ к которой открыт только после авторизации – admin.php (Рисунок 10), содержимое файла для авторизации представлено на рисунке 9.

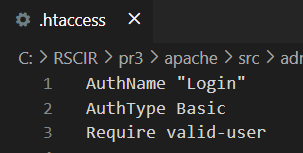


Рисунок 9 - Содержимое файла .htaccess

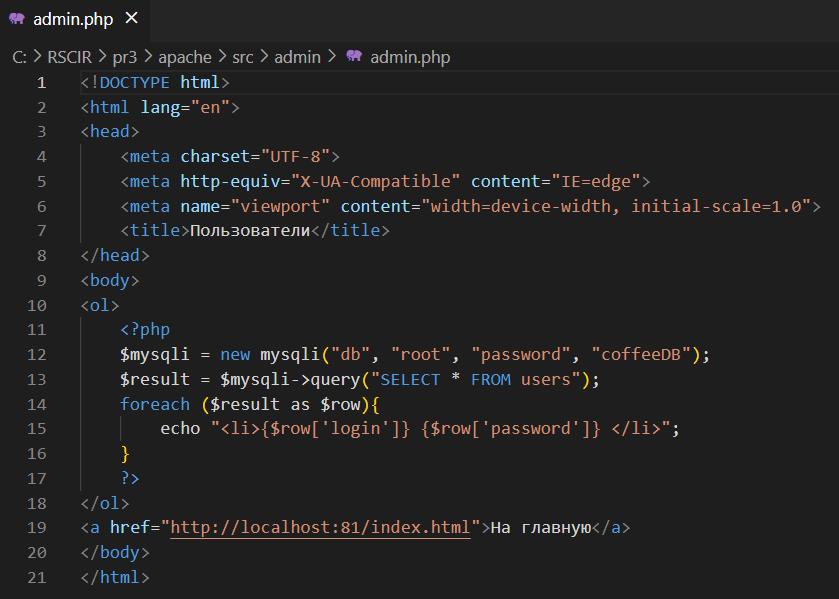


Рисунок 10 - Содержимое файла admin.php

Страницы со статистическим контентом представлены на рисунках 11, 12.

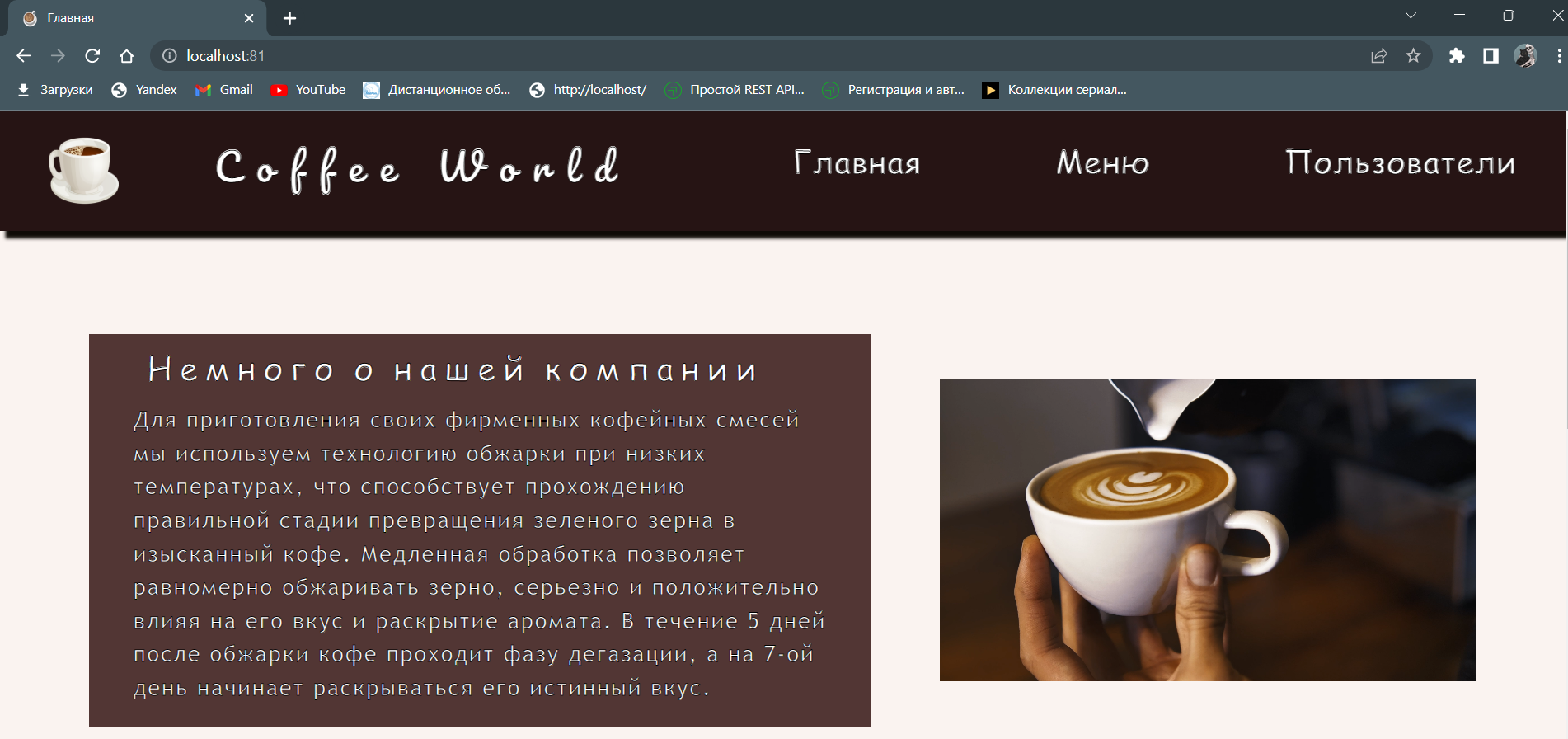


Рисунок 11 - Страница со статистическим контентом

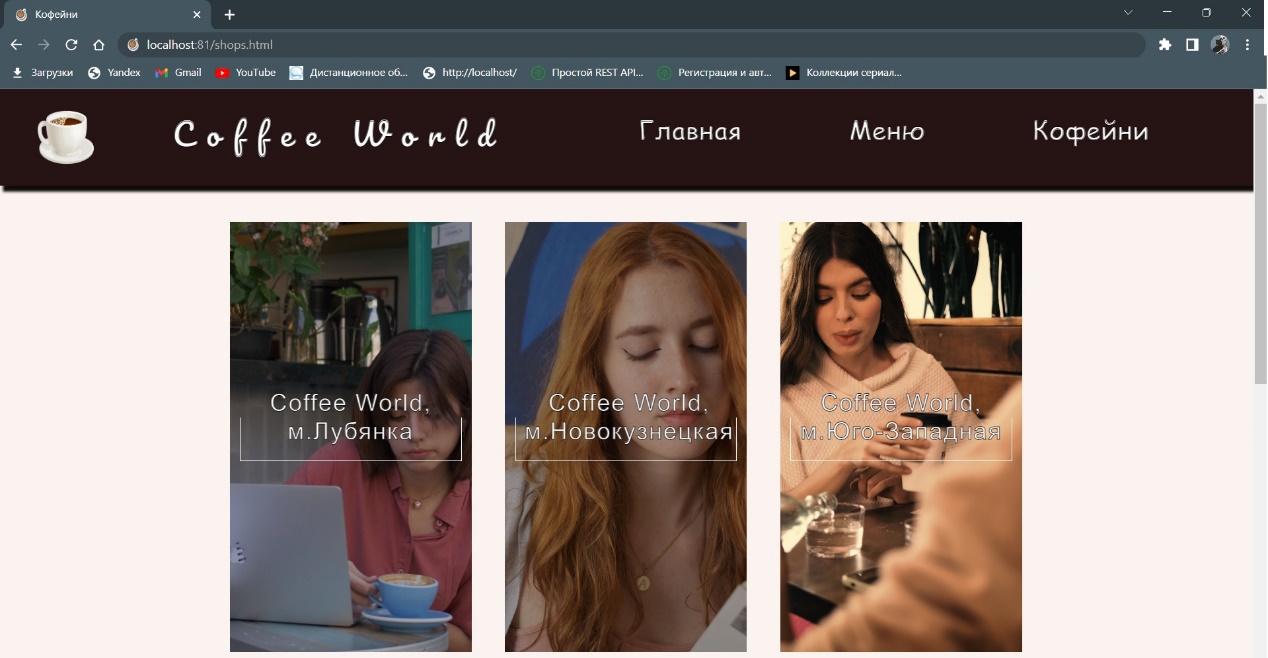


Рисунок 12 - Страница со статистическим контентом

Страницы с динамическим контентом представлены на рисунках 13-16.

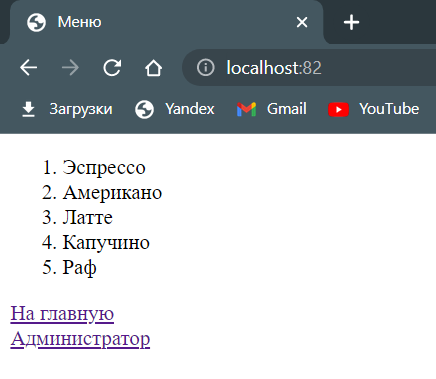


Рисунок 13 - Страница с динамическим контентом

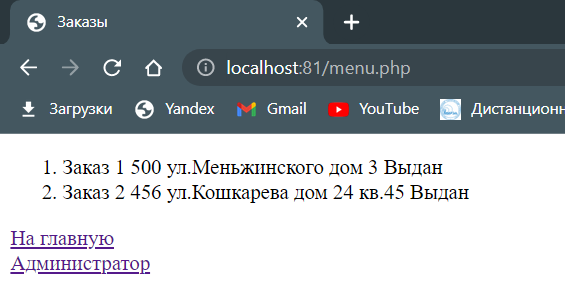


Рисунок 14 - Страница с динамическим контентом

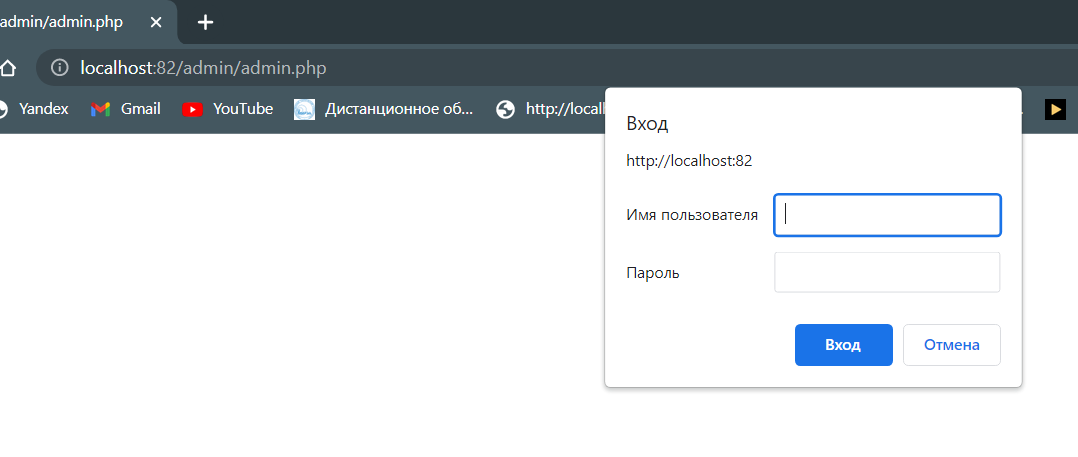


Рисунок 15 - Страница авторизации

Страница для администраторов после авторизации представлена на рисунке 15.

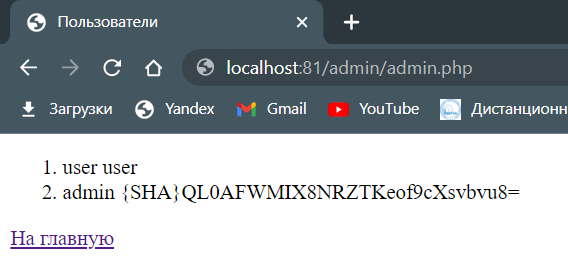


Рисунок 16 - Страница с динамическим контентом после авторизации

Ссылка на удаленный репозиторий

https://github.com/devilishgrapefruit/RSCIR/tree/main/pr3 (Рисунок 15).

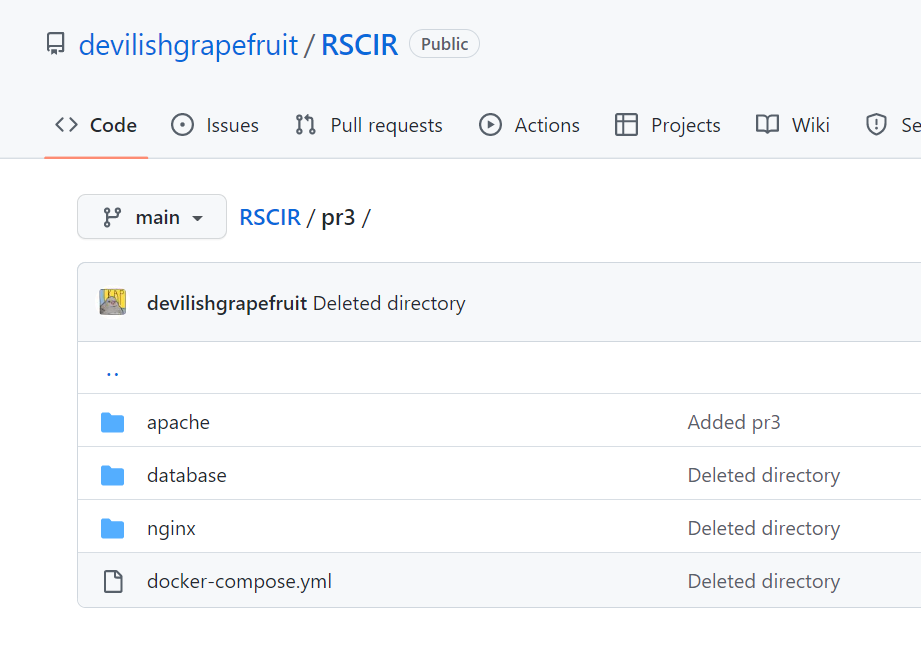


Рисунок 17 - Репозиторий

Ответы на вопросы к практической:

1. **Что такое веб-сервер?**

Веб-сервер в процессе своей работы обрабатывает запросы браузера и выдает в ответ веб-страницы. Работа веб-сервера сводится к приему HTTP-запросы от веббраузеров, выступающих в качестве и выдачи им в ответ соответствующих ответов (как правило это HTTP-ответы, включая HTML-страницы) а также файлы изображений, медиа-потоки, файлы других типов и различные данными. Важно заметить, что, по сути, веб-сервер — это информационная система, так как его функционирование невозможно как без специального программного обеспечения, так и соответствующей аппаратной части, т.е. компьютера.

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

Сервер приложений — это программная платформа, предназначенная для эффективного исполнения процедур (программ, скриптов), на которых построены приложения. Сервер приложений действует как набор компонентов, доступных разработчику программного обеспечения через API, определённый самой платформой. Основное отличие веб-сервера от сервера приложений заключается в том, что веб-сервер предназначен для обслуживания статических страниц, например HTML и CSS, тогда как сервер приложений отвечает за генерацию динамического содержимого путём выполнения кода на стороне сервера, например, JSP, EJB и т. п.

1. **Кратко опишите историю развития интернета в рамках развития веб-серверов.**

Самый первый сервер появился в прошлом веке - 6 августа 1991 года. Его создатель – учёный из Британии Тимоти Джон Бернерс-Ли. Со временем стали появляться все новые и новые серверы. Сейчас наиболее популярные – nginx и apache.

1. **Кратко опишите протокол HTTP.**

Для доступа к HTML документам посредством сети используется протокол HTTP (HyperText Transfer Protocol. Протокол HTTP реализован по клиент-серверной технологии и работает по принципу запрос-ответ без сохранения состояния.

1. **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.

Пользователь посредством HTTP-клиента (чаще всего это браузер) запрашивает у HTTP-сервера некий URL, сервер проверяет и отдает соответствующий этому URL-файл, обычно это HTML-страница. Запрашиваемый документ (в нашем примере это HTML-страница) может, в свою очередь, ссылаться на связанные ресурсы, например, изображения, файлы данных, документы. Для отображения таких связных ресурсов на странице, клиентом (браузером) последовательно запрашиваются у сервера все связные документы. Помимо изображений клиент может запрашивать таблицы стилей, скрипты, исполняемые на стороне клиента и т.д. После получения всех запрашиваемых ресурсов клиент (браузер) обработает их в соответствии с кодом HTML-документа и выдаст пользователю готовую страницу.

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

Основные цели и задачи 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.

1. **Опишите технологию SSI.**

SSI (от английского Server Side Includes – включение на стороне сервера) – это, как уже было сказано ранее, язык, разработанный для динамического создания и «сборки» веб-страниц на сервере из отдельных составных частей и выдачи клиенту полученного HTML-документа. Использование языка SSI дает возможность в код страницы инкапсулировать содержимое различных файлов. В этом случае, внесение дополнительных повторяющихся элементов, такие, например, как шапка (header), подвал (footer), меню и другие осуществляется в специальные обособленные файлы. Затем созданные таким образом файлы просто подключаются при окончательной сборке страницы.

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

Система управления содержимым — информационная система или компьютерная программа, используемая для обеспечения и организации совместного процесса создания, редактирования и управления содержимым, иначе — контентом.

1. **Верно ли, что сервер приложения умеет работать с протоколом**

**HTTP?**

Верно.

1. **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.

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

С помощью CGI можно создавать CGI-программы, называемые шлюзами, которые во взаимодействии с такими прикладными системами, как система управления базой данных, электронная таблица, деловая графика и др., смогут выдать на экран пользователя динамическую информацию. Программа-шлюз запускается сервером в реальном масштабе времени. Сервер обеспечивает передачу запроса пользователя шлюзу, а она в свою очередь, используя средства прикладной системы, возвращает результат обработки запроса на экран пользователя.

1. **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 вводавывода ограничивает возможности масштабирования и обеспечения высокой доступности, так как требует, чтобы веб-сервер и сервер приложений находились в пределах одной операционной системы (ОС).

1. **Что такое FastCGI?**

FastCGI представляет собой клиент-серверный протокол для взаимодействия веб-сервера и сервера приложений, обеспечивающий высокую производительность и безопасность. FastCGI устраняет основную проблему CGI − повторный запуск процесса веб-приложения на каждый запрос, FastCGI процессы запущены постоянно, что позволяет существенно экономить время и ресурсы

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

FastCGI не только наследует преимущества CGI, но также добавляет несколько новых функций.

Распределенные вычисления: мы можем запускать приложения FastCGI на машинах, отличных от веб-сервера. Распределенные вычисления - это проверенная технология, которая применяется для: настройки и подключения существующих совместных систем, полного использования производительности системы и повышения безопасности системы (например, брандмауэров).

Множество ролей и расширяемых ролей: приложения CGI обрабатывают HTTP-запросы и предоставляют возвращаемые значения. Функции FastCGI не ограничиваются этим, например, он выполняет модульную аутентификацию и проверки идентичности, а также конвертирует форматы данных. Эти особенности FastCGI позволяют играть больше ролей в будущем.

1. **Что такое менеджер процессов?**

Менеджер процессов является посредником между веб-сервером и серверами приложений. Это несколько усложняет схему, так как настраивать и сопровождать приходится большее количество служб, но в то же время открывает более широкие возможности, позволяя настроить каждый элемент сервера исключительно для решения своих задач.

1. **Что такое PHP-FPM?**

PHP-FPM — программный пакет, позволяющий выполнить обработку скриптов, написанных на языке PHP. Включен в состав PHP с версии 5.3.3, для более ранних версий необходима установка отдельно. Является альтернативой FastCGI — протоколу взаимодействия веб-сервера с программами. FPM расшифровывается как Fastcgi Process Manager.

1. **Что такое Spawn-fcgi?**

Spawn-fcgi используется для запуска удаленных и локальных FastCGI процессов.

1. **Что такое Lighttpd?**

Lighttpd — веб-сервер, разрабатываемый с расчётом на скорость и защищённость, а также соответствие стандартам. Это свободное программное обеспечение, распространяемое по лицензии BSD.

1. **Что такое chroot окружение?**

Chroot окружение — способ запуска программ, системный вызов и просто команда, позволяющая изменить корневой каталог в системе.

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

Веб сервер через FastCGI отправляет запрос к другому серверу (веб серверу, серверу баз данных и др), и получает ответ.

1. **Опишите процесс выбора встроенного или внешнего менеджера процессов.**

На практике, выбирая между встроенным менеджером и внешним, надо оценить ситуацию и выбирать именно тот инструмент, который наиболее подходит запросам. Например, создавая простой сервер для нескольких сайтов на типовых движках применение внешнего менеджера будет явно излишним. Данный подход хорош именно тем, что можно, как из конструктора, собрать именно то, что нужно для решения конкретной задачи.

1. **Что такое интерфейс шлюза?**

Интерфейс общего шлюза — стандарт интерфейса, используемого внешней программой для связи с веб-сервером.

1. **Что такое SCGI?**

SCGI (Simple Common Gateway Interface) — простой общий интерфейс шлюза — разработан как альтернатива CGI и во многом аналогичен FastCGI, но более прост в реализации. Все, что применимо к FastGCI, справедливо и для SCGI.

1. **Что такое PCGI?**

PCGI (Perl Common Gateway Interface) - библиотека Perl для работы с интерфейсом CGI, долгое время являлась основным вариантом работы с Perlприложениями через CGI, отличается хорошей производительностью при скромных потребностях в ресурсах и неплохой защиты от перегрузки.

1. **Что такое WSGI?**

WSGI (Web Server Gateway Interface) − предназначен для взаимодействия веб-сервера с сервером приложений для программ, написанных на языке Python.

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

Mod\_php - позволяет Apache выполнять php скрипты. Является наверно самым популярным и простым способом подружить Apache и PHP. Модуль не использует ни CGI, ни FastCGI. Есть свои минусы — скрипты работают под пользователем веб-сервера, невозможно использовать больше одной версии PHP.

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

К преимуществам HTTP сервера Apache, необходимо отнести высокий уровень надежности, гибкие настройки, свободный доступ к программе. В частности, к нему можно подключать большое количество внешних модулей, систем управления базами данных и т.п. Apache поддерживает интернетпротокол IPv6. Также к достоинствам Apache, необходимо отнести регулярно выпускаемые обновления и патчи (заплатки), которые позволяют быстро устранить различные проблемы с работой или безопасностью. Удобство и легкость настройки этого программного обеспечения, делают его одним из самых популярных вариантов как для начинающих, так и для опытных веб-мастеров.

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

Все недостатки Apache, являются логическим продолжением его достоинств. Наиболее вескими недостатками являются:

1. Возможные проблемы с производительностью на высоконагруженных сайтах с большим трафиком, «тяжелым» контентом или приложениями, которые требуют высоких вычислительных мощностей.

2. Большое количество параметров настройки может привести к возникновению уязвимостей, из-за невнимательности при конфигурации.

3. Существует вероятность того, что в модули от независимых разработчиков будет внедрен вредоносный код.

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

В архитектуру Apache входит: простое ядро, платформо-зависимый уровень (APR), и модули. Любое приложение для Apache - даже простейшее, обслуживающее "дефолтную" страницу Apache "It worked" - использует несколько модулей. Пользователи Apache не нуждаются в знании этого, но для разработчика программ, понимание модулей и API модуля Apache является необходимым знанием при работе с Apache. Большинство, но не все модули, связаны с различными аспектами обработки НТТР запроса. Преимущество модульного подхода состоит в том, что он позволяет фокусировать модуль на специфическую задачу, игнорируя при этом другие аспекты НТТР, не касающиеся данной задачи.

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

Ядро HTTP сервера Apache разрабатывается фондом Apache Software Foundation, который поддерживает огромное количество разработчиков по всему миру. Его основными функциями являются: 1. Передача данных по HTTP. 2. Обработка файлов. 3. Загрузка и поддержка модулей. Сервер может функционировать без дополнительных модулей, однако в этом случае, его возможности крайне ограничены.

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

Конфигурацию Apache, можно разделить на три основных уровня: 1. Конфигурация сервера. 2. Конфигурация виртуального хоста. 3. Конфигурация уровней каталога. Все настройки осуществляются при помощи специальных текстовых файлов, со своим собственным языком, который основан на блоках директив. С их помощью, могут быть изменены практически все параметры ядра. Большинство модулей, имеет свои собственные параметры. Также они достаточно часто используют для своей работы настроечные файлы операционных систем. Ряд настроек можно задавать с помощью параметров командной строки.

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

Целью запроса служит некий ресурс, который определяется единым идентификатором ресурса − URI (Uniform Resource Identifier), HTTP использует одну из разновидностей URI − URL (Uniform Resource Locator) − универсальный указатель ресурса, который помимо сведений о ресурсе определяет также его физическое местоположение.

ВЫВОДЫ

В работе была продолжена настройка сложной серверной конфигурации, состоящей из связки apache+nginx+php+База данных. В результате работы было создано веб-приложение с динамическими веб-страницами и веб-страницами со статическим контентом.

СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ

1. Разработка серверных частей интернет-ресурсов [Электронный ресурс]: Учебное пособие / Волков М.Ю., Литвинов В.В., Лобанов А.А. и др. — М.: МИРЭА – Российский технологический университет, 2021.
2. Документация Docker URL: https://docs.docker.com/ (дата обращения: 20.10.2022).