|  |  |  |
| --- | --- | --- |
|  |  |  |
| 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 №2**

по дисциплине «Разработка серверных частей интернет-ресурсов»

**Студент группы** ИКБО-01-21 Кузнецов А.А.

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

(подпись студента)

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

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

(подпись руководителя)

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

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

Москва 2023

# 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-compose:

• Docker-контейнеры, созданные ранее

• Примонтированные тома

• Настройки портов

• Связь между контейнерами

• Переменные внешнего окружения

• Настройки базы данных

1. Создать sql скрипт инициализации, где студенту необходимо реализовать любую одну сущность (например товары или студенты).

2. Реализовать связь между базой данных и веб-сервером.

3. Требуется создать 4 страницы php, где на каждой странице будет реализована одна из CRUD операций с выбранной вами сущностью.

# ХОД РАБОТЫ

Для создания образа веб-сервера было использовано два Dockerfile. Первый был использован для образа веб-сервера, изображенный на рисунке 1, а также второй для инициализации БД с помощью скрипта, изображенный на рисунке 2.

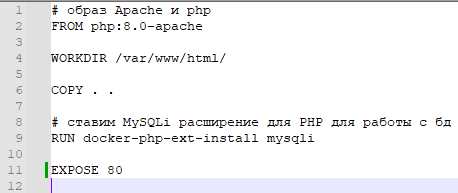


Рисунок 1 – Dockerfile для веб-сервера

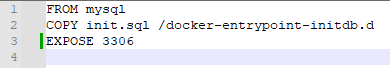


Рисунок 2 – Dockerfile для инициализации БД из файла

Затем мы создаем контейнер с помощью docker compose, чтобы связать образ базы данных с образом нашего веб-сервера, что показано на рисунке 3.



Рисунок 3 – docker-compose проекта

Для инициализации базы данных мы используем файл init.sql, показанный на рисунке 4.

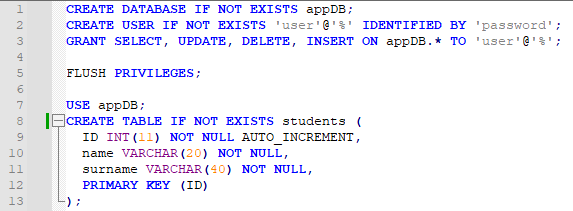


Рисунок 4 – Инициализация базы данных

Для реализации CRUD-команд были использованы php-скрипты. На рисунке 5 показан файл html-страницы со php-скриптом чтения записей read.php, а также его работа и проверка работоспособности в Postman изображена на рисунке 6.

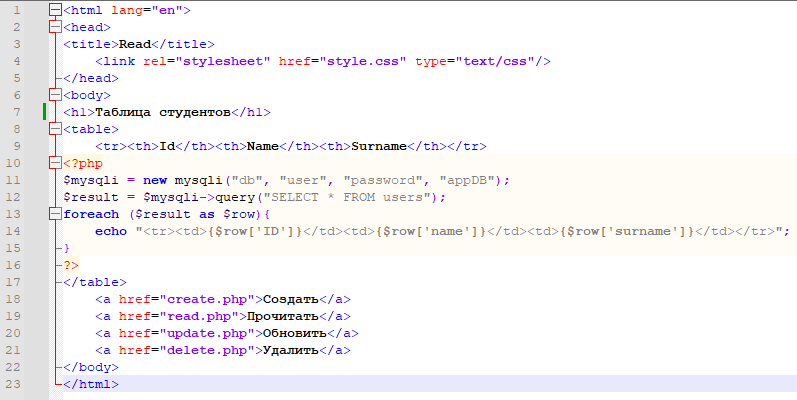


Рисунок 5 – Файл read.php

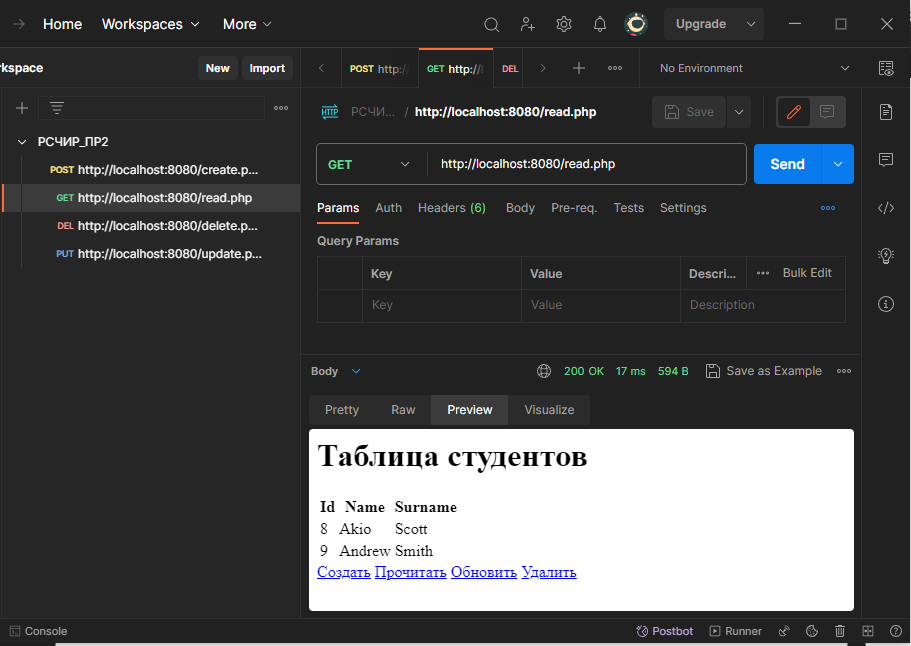


Рисунок 6 – Обработка запроса из Postman

На рисунке 7 показан файл html-страницы со php-скриптом добавления записей create.php, а также его работа и проверка работоспособности в Postman изображена на рисунке 8.



Рисунок 7 – Файл create.php

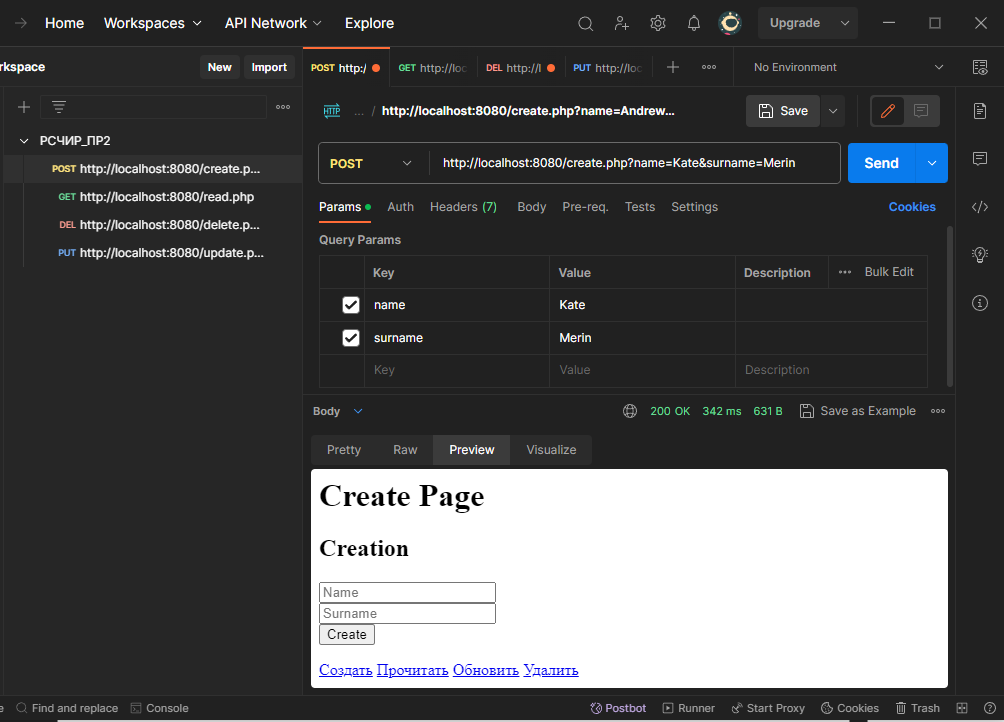


Рисунок 8 – Обработка запроса из Postman

На рисунке 9 показан файл html-страницы со php-скриптом изменения записей update.php, а также его работа и проверка работоспособности в Postman изображена на рисунке 10.

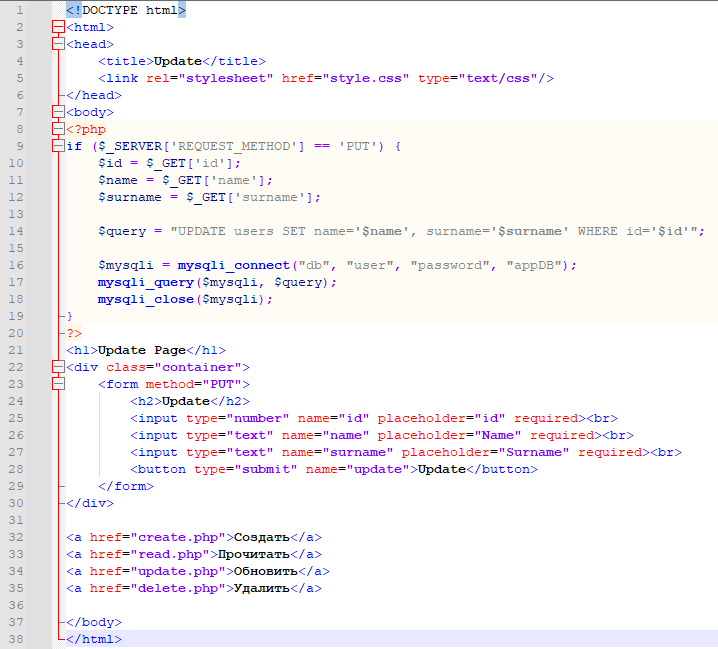


Рисунок 9 – Файл update.php

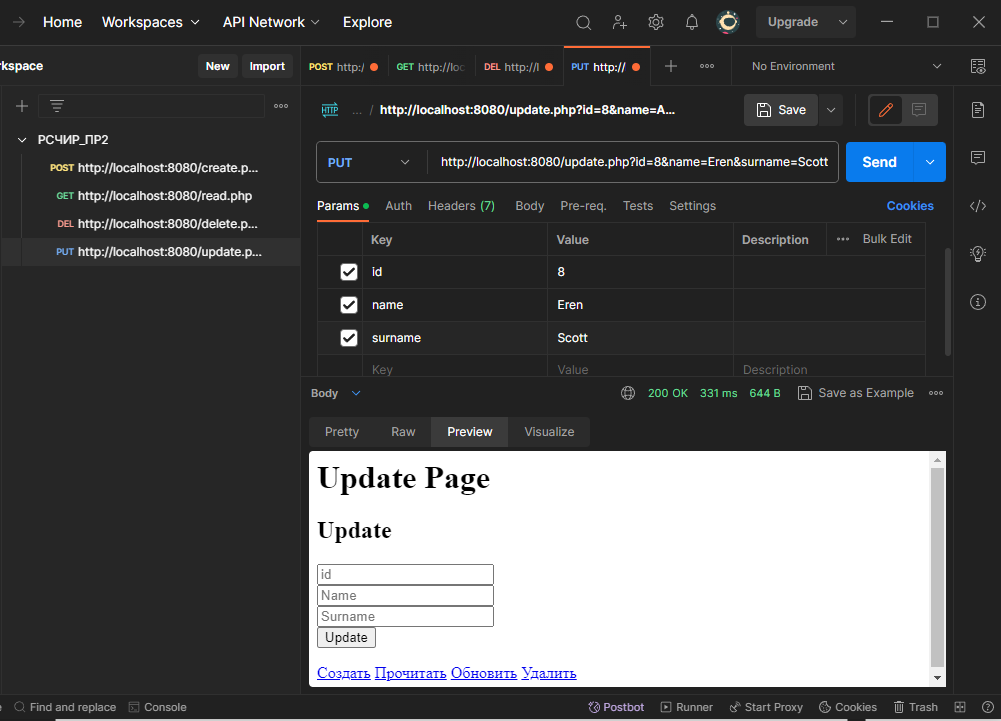


Рисунок 10 – Обработка запроса из Postman

На рисунке 11 показан файл html-страницы со php-скриптом удаления записей delete.php, а также его работа и проверка работоспособности в Postman изображена на рисунке 12.



Рисунок 11 – Файл delete.php

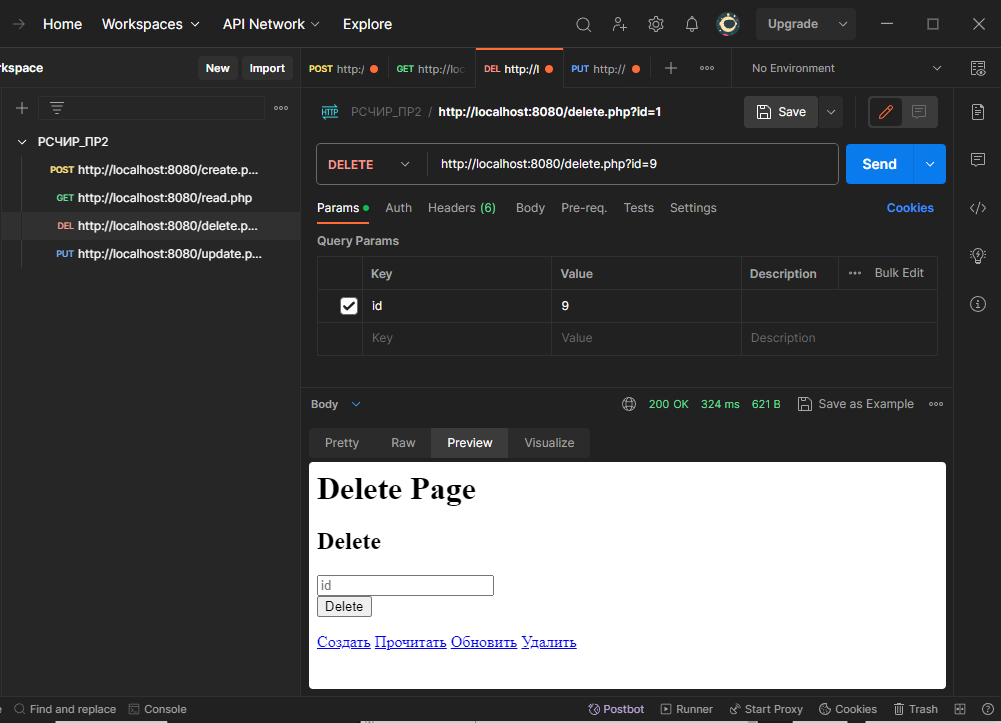


Рисунок 12 – Обработка запроса из Postman

Проверим результат работы ещё раз воспользовавшись запросом на получение записей, рисунок 13.

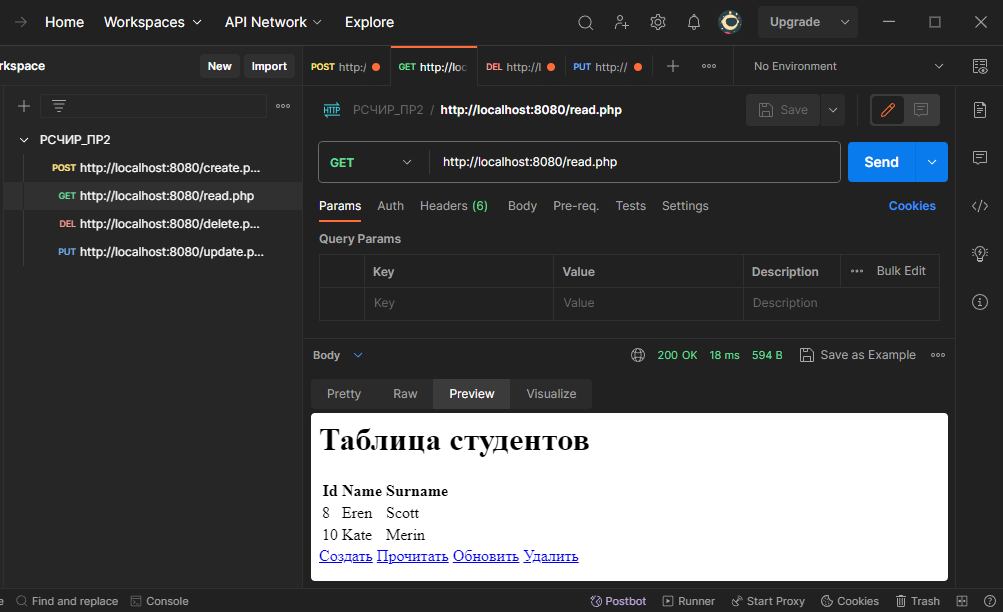


Рисунок 13 – Проверка результата выполнения запросов на сервер

# Вывод

Мы на практике убедились, что работа выполнена верно. В процессе работы было изучено создание конфигурации docker-compose для веб-сервера на базе Apache с использованием СУБД MySQL с примонтированным томом для БД и использованием php-скриптов на ряду с html-страницами с дальнейшей проверкой работы на них запросов в Postman.

# ОТВЕТЫ НА ВОПРОСЫ

1. В чём преимущество языка PHP для backend'а?

Преимущества PHP для backend'а включают:

* Простоту использования и изучения.
* Широкую поддержку баз данных и веб-серверов.
* Большое сообщество разработчиков и наличие множества библиотек и фреймворков.
* Встроенную поддержку веб-разработки, позволяющую вставлять PHP-код прямо в HTML-страницы.
* Возможность реализации как тонкого, так и толстого клиента.

1. Как достигается тонкий клиент при использовании PHP?

Тонкий клиент означает, что большая часть логики приложения выполняется на сервере. Для достижения этого с использованием PHP:

Логика приложения (контроллер) размещается на сервере, и сервер выполняет вычисления и обработку запросов.

Веб-сервер является "мозгом" приложения и отвечает за координацию работы сервера и клиентских устройств.

JavaScript на клиентской стороне используется для стандартных задач, таких как отображение данных и взаимодействие с пользователем.

Это позволяет минимизировать размер и сложность кода JavaScript, улучшить производительность и общее качество работы приложения.

1. Что такое SSR?

SSR (Server-Side Rendering) - это подход к созданию веб-страниц, при котором их начальный рендеринг происходит на сервере, а не на стороне клиента.

При использовании SSR веб-страницы генерируются на серверной стороне, и сервер отправляет готовый HTML-код браузеру для отображения пользователю.

SSR может быть использован с разными языками программирования, включая PHP, с использованием фреймворков и библиотек для рендеринга пользовательских интерфейсов.

Этот подход улучшает производительность и SEO-оптимизацию веб-страниц, так как страницы загружаются быстрее и более доступны для поисковых роботов.

# СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Документация // PHP URL: <https://www.php.net/manual/ru/index.php> (дата обращения: 14.09.2023). – Текст: электронный.

2. Статья о назначении докера простыми словами: <https://habr.com/ru/post/309556/> (дата обращения: 14.09.2023).

3. Методические указания по выполнению практической работы: <https://online-edu.mirea.ru/mod/resource/view.php?id=508421> (дата обращения: 10.09.2023).

4. Официальная документация докера: <https://docs.docker.com/> (дата обращения: 14.09.2023).

5. Более сложная и подробная статья про докер: <https://habr.com/ru/post/277699/> (дата обращения: 14.09.2023).