



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 №1**

по дисциплине «Разработка серверных частей интернет-ресурсов»

**Тема практической работы:**

**Студент группы ИКБО-30-20**

**Земар Аким Саидович**

---

(подпись студента)

**Руководитель практической работы**

---

(подпись руководителя)

Работа представлена

«\_\_\_» \_\_\_\_\_ 2022 г.

Допущен к работе

«\_\_\_» \_\_\_\_\_ 2022 г.

Москва 2022

## Цель работы:

Создать свою конфигурацию серверного программного обеспечения, в которой должны присутствовать веб-сервер, операционная система, язык программирования и база данных.

## Постановка задачи:

Для проверки работоспособности конфигурации требуется инициализировать базу данных: создать отдельного пользователя для работы с ней, создать базу данных, в которой создать таблицу пользователя с полями: идентификационный номер, имя, фамилия. Также для проверки конфигурации требуется сгенерировать тестовую страничку, содержащую выборку из созданной таблицы и информационное сообщение о версии языка программирования, его настройках и конфигурации.

## Ход работы:

Мною было принято решение использовать технологии WAMP — Microsoft Windows, Apache, MySQL, PHP. Для этого необходимо было создать файл `docker-compose.yml`, в котором будут прописаны инструкции для запуска и настройки необходимых нам сервисов. Структура проекта будет выглядеть следующим образом (Рисунок 1).

core	08.09.2022 21:27	Папка с файлами	
db	08.09.2022 20:44	Папка с файлами	
.env	08.09.2022 21:24	Файл "ENV"	1 КБ
docker-compose.yml	08.09.2022 21:24	Файл "YML"	1 КБ

Рисунок 1 – Структура проекта.

Также в директории `server` присутствует `Dockerfile`, необходимый для загрузки образа `php apache` и установки нужных расширений (Рисунок 2).

```
FROM php:apache
WORKDIR /var/www/html/
COPY . .
RUN docker-php-ext-install mysqli
EXPOSE 80
```

Рисунок 2 – Содержимое Dockerfile.

В файле docker-compose.yml прописаны инструкции для запуска необходимых нам технологий (Рисунок 3).

```
1      version: '3.9'
2
3  >> services:
4  >   apache:
5       build: ./core
6       restart: always
7       volumes:
8         - ./core:/usr/local/apache2/htdocs/
9       ports:
10        - "8080:80"
11       depends_on:
12         - db
13
14  >   db:
15       image: mysql:latest
16       environment:
17         MYSQL_ROOT_PASSWORD: 123456
18         MYSQL_DATABASE: app_db
19         MYSQL_USER: mysql
20         MYSQL_PASSWORD: 123456
21       ports:
22        - "3306:3306"
23       volumes:
24        - ./db:/docker-entrypoint-initdb.d
25
```

Рисунок 3 – Инструкции для базы данных.

Через консоль осуществим запуск с помощью команды `docker-compose up` (Рисунок 4).

```
PS C:\Users\akimz\Desktop\учеба\PHP\PHP_prac1> docker-compose up
php_prac1_db_1 is up-to-date
php_prac1_apache_1 is up-to-date
php_prac1_phpmysqladmin_1 is up-to-date
Attaching to php_prac1_db_1, php_prac1_apache_1, php_prac1_phpmysqladmin_1
db_1 | 2022-09-08 18:17:23+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 8.0.30-1.el8 started.
db_1 | 2022-09-08 18:17:23+00:00 [Note] [Entrypoint]: Switching to dedicated user 'mysql'
db_1 | 2022-09-08 18:17:23+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 8.0.30-1.el8 started.
db_1 | 2022-09-08 18:17:23+00:00 [Note] [Entrypoint]: Initializing database files
db_1 | 2022-09-08T18:17:23.334828Z 0 [Warning] [MY-011068] [Server] The syntax '--skip-host-cache' is deprecated and will be removed in a future release.
db_1 | 2022-09-08T18:17:23.334899Z 0 [System] [MY-013169] [Server] /usr/sbin/mysqld (mysqld 8.0.30) initializing of server in progress as daemon
db_1 | 2022-09-08T18:17:23.362254Z 1 [System] [MY-013576] [InnoDB] InnoDB initialization has started.
db_1 | 2022-09-08T18:17:26.601761Z 1 [System] [MY-013577] [InnoDB] InnoDB initialization has ended.
db_1 | 2022-09-08T18:17:29.762108Z 6 [Warning] [MY-010453] [Server] root@localhost is created with an empty password ! Please consider using long and random passwords instead of empty ones.
db_1 | 2022-09-08 18:17:41+00:00 [Note] [Entrypoint]: Database files initialized
db_1 | 2022-09-08 18:17:41+00:00 [Note] [Entrypoint]: Starting temporary server
db_1 | 2022-09-08T18:17:41.652686Z 0 [Warning] [MY-011068] [Server] The syntax '--skip-host-cache' is deprecated and will be removed in a future release.
db_1 | 2022-09-08T18:17:41.654141Z 0 [System] [MY-010116] [Server] /usr/sbin/mysqld (mysqld 8.0.30) starting as process 131
db_1 | 2022-09-08T18:17:41.670525Z 1 [System] [MY-013576] [InnoDB] InnoDB initialization has started.
db_1 | 2022-09-08T18:17:41.846275Z 1 [System] [MY-013577] [InnoDB] InnoDB initialization has ended.
db_1 | 2022-09-08T18:17:42.185272Z 0 [Warning] [MY-010068] [Server] CA certificate ca.pem is self signed.
db_1 | 2022-09-08T18:17:42.185317Z 0 [System] [MY-013602] [Server] Channel mysql_main configured to support TLS. Encrypted connections are now supported.
db_1 | 2022-09-08T18:17:42.194780Z 0 [Warning] [MY-011810] [Server] Insecure configuration for --pid-file: Location '/var/run/mysqld' in the path is problematic. See https://mariadb.org/kb/en/mysqld_safe_pid_file_path/ for more details.
```


Рисунок 4 – Запуск docker-compose.

Переходим по localhost:8000 и видим наш готовый результат (Рисунок 5).

Таблица пользователей данного продукта

Id	Name	Surname
1	Alex	Rover
2	Bob	Marley
3	Kate	Yandson
4	Lilo	Black

PHP Version 8.1.10



System	Linux db610cb4e593 5.10.16.3-microsoft-standard-WSL2 #1 SMP Fri Apr 2 22:23:49 UTC 2021 x86_64
Build Date	Sep 1 2022 21:15:46
Build System	Linux 072284680fe0 5.10.0-13-cloud-amd64 #1 SMP Debian 5.10.106-1 (2022-03-17) x86_64 GNU/Linux
Configure Command	'/configure' '--build=x86_64-linux-gnu' '--with-config-file-path=/usr/local/etc/php' '--with-config-file-scan-dir=/usr/local/etc/php/conf.d' '--enable-option-checking=fatal' '--with-mhash' '--with-pic' '--enable-ftp' '--enable-mbstring' '--enable-mysqlnd' '--with-password-argon2' '--with-sodium=shared' '--with-pdo-sqlite=/usr' '--with-sqlite3=/usr' '--with-curl' '--with-iconv' '--with-openssl' '--with-readline' '--with-zlib' '--disable-phpdbg' '--with-pear' '--with-libdir=lib/x86_64-linux-gnu' '--disable-cgi' '--with-apxs2' 'build_alias=x86_64-linux-gnu'
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/usr/local/etc/php
Loaded Configuration File	(none)
Scan this dir for additional .ini files	/usr/local/etc/php/conf.d
Additional .ini files parsed	/usr/local/etc/php/conf.d/docker-php-ext-mysql.ini, /usr/local/etc/php/conf.d/docker-php-ext-sodium.ini
PHP API	20210902
PHP Extension	20210902
Zend Extension	420210902
Zend Extension Build	API420210902.NTS
PHP Extension Build	API20210902.NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled
Zend Memory Manager	enabled
Zend Multibyte Support	provided by mbstring
IPv6 Support	enabled
DTrace Support	disabled
Registered PHP Streams	https, ftps, compress.zlib, php, file, glob, data, http, ftp, phar
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, tls, tlsv1.0, tlsv1.1, tlsv1.2, tlsv1.3
Registered Stream Filters	zlib.*, convert.iconv.*, string.rot13, string.toupper, string.tolower, convert.*, consumed, dechunk

Рисунок 5 – Результат.

Ссылка на репозиторий: [https://github.com/MycroftBrooks/PHP\\_mirea](https://github.com/MycroftBrooks/PHP_mirea)

### **Выводы:**

В ходе выполнения практической работы была успешно установлена конфигурация WAMP и выполнены поставленные задачи.

### **Ответы на вопросы:**

1. Клиент – компьютерное устройство, которое отсылает запросы серверу, касающиеся выполнения определенных задач или предоставления конкретной информации.  
Сервер – компьютерное устройство, которое взаимодействует с многочисленными клиентами и решает поставленные ими задачи.
2. База данных — это упорядоченный набор структурированной информации или данных, которые обычно хранятся в электронном виде в компьютерной системе.
3. Интерфейс прикладного программирования (application programming interface, API) позволяет прикладной программе использовать определенные данные или функции другой программы.
4. Сервис, отличия от сервера. 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.
5. Архитектура клиент-сервер определяет лишь общие принципы взаимодействия между компьютерами, детали взаимодействия определяют различные протоколы. Данная концепция нам говорит, что нужно разделять машины в сети на клиентские, которым всегда что-то надо и на серверные, которые дают то, что надо. При этом взаимодействие всегда начинается клиент, а правила, по которым происходит взаимодействие описывает протокол.
6. Виды сервисов - сервисы делятся на два вида по способу использования: Started и Bound, и на два вида по способу взаимодействия с пользователем: Background и Foreground. Started Service означает, что при создании объекта Intent было передано имя класса сервиса. Bound Service привязывается к компоненту вызовом метода `bindService(Intent service, ServiceConnection serviceConnection, int flags)`.
7. Масштабируемость - это возможность увеличить производительность проекта за минимальное время путем добавления ресурсов.
8. Протоколы передачи данных - это стек протоколов TCP и IP. Первый обеспечивает и контролирует надёжную передачу данных и следит за

её целостностью. Второй же отвечает за маршрутизацию для отправки данных.

9. Толстый и тонкий клиенты - Толстые клиенты работают с информацией на основе собственных аппаратных и программных возможностей, в то же время тонкие применяют ПО центрального сервера только чтобы обработать данные, предоставляя системе лишь требуемый графический интерфейс для выполнения работы пользователем.
10. Паттерн MVC: общие тезисы - MVC расшифровывается как модель-представление-контроллер (от англ. model-view-controller). Это способ организации кода, который предполагает выделение блоков, отвечающих за решение разных задач. Один блок отвечает за данные приложения, другой отвечает за внешний вид, а третий контролирует работу приложения. Компоненты: модель, представление и контроллер.
11. Паттерн MVC: Model-View-Presenter - MVP (Model-View-Presenter) — паттерн разработки пользовательского интерфейса. Шаблон MVP является производным от MVC, но при этом имеет несколько иной подход. Основное отличие — представление (presenter) не так сильно связано моделью (model). Модель представляет данные для отображения пользователю.
12. Паттерн MVC: Model-View-ViewModel - Model-View-ViewModel (MVVM) — шаблон проектирования архитектуры приложения. Представлен как модификация шаблона Presentation Model. Ориентирован на современные платформы разработки, такие как Windows Presentation Foundation, Silverlight от компании Microsoft, ZK framework.
13. Паттерн MVC: Model-View-Controller - Model-View-Controller — схема разделения данных приложения и управляющей логики на три отдельных компонента: модель, представление и контроллер — таким образом, что модификация каждого компонента может осуществляться независимо. Модель предоставляет данные и реагирует на команды контроллера, изменяя своё состояние.
14. Docker — это платформа, которая позволяет упаковать в контейнер приложение со всем окружением и зависимостями, а затем доставить и запустить его в целевой системе. Приложение, упакованное в контейнер, изолируется от операционной системы и других приложений.
15. Dockerfile — это текстовый файл с инструкциями, необходимыми для создания образа контейнера.
16. Docker Compose — это средство для определения и запуска приложений Docker с несколькими контейнерами. При работе в Compose вы используете файл YAML для настройки служб

приложения. Затем вы создаете и запускаете все службы из конфигурации путем выполнения одной команды.

17. LAMP — это стек программного обеспечения, устанавливаемого на сервер и предназначенного для сайтов и веб-приложений. LAMP – это аббревиатура, она расшифровывается как: Linux, Apache, MySQL и PHP.