



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 «Проектирование и разработка серверных частей интернет-ресурсов»

Студент группы ИКБО-21-23

Муравьев А.О.

(подпись студента)

Руководитель практической работы

Благирев М.М.

(подпись руководителя)

Работа представлена

«___» _____ 2025 г.

Допущен к работе

«___» _____ 2025 г.

Москва 2025

СОДЕРЖАНИЕ

ЦЕЛЬ РАБОТЫ	3
ХОД РАБОТЫ	4
ВЫВОД	5
ОТВЕТЫ НА КОНТРОЛЬНЫЕ ВОПРОСЫ	6

ЦЕЛЬ РАБОТЫ

Предлагается создать свою конфигурацию серверного программного обеспечения, в которой должны присутствовать веб-сервер, операционная система, язык программирования и база данных. Данная конфигурация будет использоваться для выполнения следующих практических работ по данной дисциплине и для выполнения курсового проектирования.

Дается рекомендация использовать ОС Linux, язык программирования PHP, веб-сервер Apache и СУБД MySQL.

Для проверки работоспособности вашей конфигурации требуется инициализировать базу данных: создать отдельного пользователя для работы с ней, создать базу данных, в которой создать таблицу «пользователи» с полями: идентификационный номер, имя, фамилия.

Также для проверки вашей конфигурации требуется сгенерировать тестовую страничку, содержащую выборку из созданной таблицы и информационное сообщение о версии языка программирования, его настройках и конфигурации.

ХОД РАБОТЫ

ВЫВОД

Таким образом, был произведен корректный запуск приложенного к практической работе php скрипта генерации страницы с характеристиками веб-сервера.

Исходный код проекта расположен по адресу:

<https://github.com/alexomur/MireaBackend/tree/master/Prac1>

ОТВЕТЫ НА КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Сервер и клиент – Сервер предоставляет ресурсы/услуги, клиент их потребляет (например, веб-сервер и браузер).
2. База данных – Организованное хранилище данных (например, MySQL), управляемое СУБД.
3. API – Интерфейс для взаимодействия между программными компонентами (например, REST API).
4. Сервис, отличия от сервера – Сервис выполняет конкретную функцию (например, авторизация), сервер — физическая/виртуальная система, предоставляющая сервисы.
5. Архитектура клиент-сервер – Модель взаимодействия, где клиент запрашивает услуги, а сервер их предоставляет.
6. Виды сервисов – Веб-сервисы, микросервисы, облачные сервисы и т.д.
7. Масштабируемость – Способность системы работать при увеличении нагрузки (горизонтальная/вертикальная).
8. Протоколы передачи данных – Правила обмена данными (HTTP, TCP/IP, FTP).
9. Тонкий и толстый клиенты – Тонкий клиент минимально загружен логикой (браузер), толстый — содержит больше функций (десктопприложение).
10. Паттерн MVC: общие тезисы – Разделение приложения на Model (данные), View (отображение), Controller (логика).
11. MVC: Model-View-Presenter – Presenter mediates between View and Model, обрабатывает пользовательский ввод.
12. MVC: Model-View-View Model (MVVM) – View Model обеспечивает связь данных с View через привязки. 7
13. MVC: Model-View-Controller – Controller обрабатывает input, обновляет Model и View.

14. Docker: общие тезисы – Технология контейнеризации для изоляции и развёртывания приложений.

15. Dockerfile – Скрипт для сборки Docker-образа (инструкции: FROM, RUN, COPY и т.д.).

16. Docker Compose – Инструмент для оркестровки многоконтейнерных приложений.

17. LAMP – Стандартный стек серверного ПО: Linux, Apache, MySQL, PHP.