ГБПОУ НСО «Новосибирский политехнический колледж»

# Отчет по заданию 3 УП

Тема: **–** Технологии для реализации веб-сервера

Выполнили:

Плотников Иван Станиславович

Захаров Даниил Вячеславович

Группа: 023 ИСП

Дата: 30.01.2025

Оглавление

[Отчет по заданию 3 УП 0](#_Toc194007575)

[1. Введение 3](#_Toc194007576)

[2. Ознакомление с программными решениями для создания веб сервера 4](#_Toc194007577)

[2.1 Apache HTTP Server 4](#_Toc194007578)

[2.2 Nginx 4](#_Toc194007579)

[2.3 Microsoft IIS (Internet Information Services) 5](#_Toc194007580)

[2.4 Lighttpd 6](#_Toc194007581)

[3. Ознакомление с языком PHP и его замены 7](#_Toc194007582)

[3.1 Можно ли его заменить ? 7](#_Toc194007583)

[3.1.1 JavaScript (Node.js) 7](#_Toc194007584)

[3.1.2 Python (Django, Flask) 8](#_Toc194007585)

[3.1.3 Ruby (Ruby on Rails) 8](#_Toc194007586)

[3.1.4 Go (Golang) 8](#_Toc194007587)

[4. Ознакомление с различными SQL СУБД 9](#_Toc194007588)

[4.1 Подходящие для веб-разработки 9](#_Toc194007589)

[4.1.1 MySQL / MariaDB 9](#_Toc194007590)

[4.1.2 PostgreSQL 9](#_Toc194007591)

[4.1.3 SQLite 10](#_Toc194007592)

[4.2 Нецелесообразные для веб-разработки 10](#_Toc194007593)

[4.2.1 Microsoft SQL Server 10](#_Toc194007594)

[4.2.2 Oracle Database 10](#_Toc194007595)

[5. Поверхностное ознакомление с NoSQL СУБД (Несколько примеров) 11](#_Toc194007596)

[5.1 Документоориентированные СУБД: 11](#_Toc194007597)

[5.2 Ключ-значение СУБД: 11](#_Toc194007598)

[6. Процессы установки(Задание) 13](#_Toc194007599)

[7. Выводы 20](#_Toc194007600)

[8. Список литературы 21](#_Toc194007601)

# 1. Введение

Веб-серверы играют ключевую роль в функционировании интернет-ресурсов, обеспечивая обработку запросов клиентов, хранение и управление данными, а также выполнение серверных сценариев. Мы будем работать на операционной системе Windows 11. Целью данного отчета является ознакомление с основными технологиями, лежащими в основе веб-серверов, их реализацией и настройкой.

А именно мы рассмотрим структуру веб-сервера, включающая в себя:

Технологии реализации самого веб-сервера: Изучение принципов работы веб-сервера, его архитектуры и функций.

Языки, используемые для поддержания функционала веб-сервера: Анализ роли языков программирования, таких как PHP, в создании динамических веб-сайтов.

СУБД (Средства управления базами данных): Рассмотрение функций системы управления базами данных (СУБД), таких как MySQL, и их значения в хранении и управлении данными веб-приложений.

Практическая часть отчета посвящена установке, настройке и функционированию компонентов, составляющих полноценный веб-сервер.

# 2. Ознакомление с программными решениями для создания веб сервера

Для развертывания веб-сервера существует множество программных решений, каждое из которых обладает своими особенностями, преимуществами и недостатками. В данном разделе рассматриваются наиболее популярные из них, включая Apache, Nginx, IIS и Lighttpd.

## 2.1 Apache HTTP Server

Преимущества:

Высокая надежность и стабильность – Apache является одним из самых проверенных веб-серверов, работающим десятилетиями.

Гибкость настройки – поддерживает модульную архитектуру (.htaccess, модули mod\_rewrite, mod\_ssl и др.).

Кроссплатформенность – работает на Windows, Linux, macOS и других ОС.

Широкая документация и поддержка – большое сообщество и множество готовых решений.

Недостатки:

Высокое потребление ресурсов при высокой нагрузке по сравнению с событийно-ориентированными серверами (Nginx).

Менее эффективен в обработке статического контента при большом количестве одновременных подключений.

Особенности:

Использует мультипроцессорную или мультипоточную модель (MPM – Multi-Processing Module).

Подходит для динамических сайтов на PHP, Python (mod\_wsgi), Perl.

## 2.2 Nginx

Преимущества:

Высокая производительность благодаря асинхронной архитектуре (обрабатывает тысячи соединений с минимальными затратами CPU и RAM).

Эффективная раздача статики – оптимален для CDN и медиаконтента.

Поддержка proxy и балансировки нагрузки – часто используется как фронтенд для Apache или приложений на Node.js, Python.

Недостатки:

Сложность динамической конфигурации – нет аналога .htaccess, изменения требуют перезагрузки.

Меньшая гибкость для некоторых CMS (например, WordPress иногда требует дополнительной настройки).

Особенности:

Часто применяется в связке Nginx + PHP-FPM для ускорения работы PHP.

Популярен в высоконагруженных проектах (например, Netflix, WordPress.com).

## 2.3 Microsoft IIS (Internet Information Services)

Преимущества:

Интеграция с Windows – глубокая оптимизация под ОС от Microsoft, поддержка .NET, ASP.

Графический интерфейс управления (IIS Manager) – удобен для администраторов Windows.

Поддержка технологий Microsoft (MS SQL, Active Directory, PowerShell).

Недостатки:

Закрытый код – ограниченная кастомизация по сравнению с OpenSource-аналогами.

Привязка к ОС Windows – не подходит для Linux-серверов.

Особенности:

Оптимален для корпоративных решений на стеке Microsoft (ASP.NET, C#).

## 2.4 Lighttpd

Преимущества:

Минимальное потребление ресурсов – подходит для встраиваемых систем и слабых серверов.

Быстрая обработка статики – сравним с Nginx по эффективности.

Простая конфигурация (меньше "тяжелых" модулей, чем у Apache).

Недостатки:

Ограниченная функциональность – меньше модулей и интеграций.

Узкое сообщество – сложнее найти готовые решения для редких задач.

Особенности:

Используется в проектах, где критична экономия ресурсов (например, Raspberry Pi, IoT-устройства).

# 3. Ознакомление с языком PHP и его замены

PHP (Hypertext Preprocessor) – это серверный язык программирования, изначально созданный для разработки веб-приложений. Он интегрируется с веб-серверами (Apache, Nginx) и взаимодействует с базами данных (MySQL, PostgreSQL), что делает его ключевым элементом в стеке LAMP (Linux, Apache, MySQL, PHP) и его аналогах.

Его преимущества:

Простота и быстрое развертывание – низкий порог входа, обширная документация. Широкая поддержка хостингов – PHP работает почти на всех веб-серверах. Интеграция с популярными CMS (WordPress, Joomla, Drupal, OpenCart). Большое количество фреймворков (Laravel, Symfony, CodeIgniter) для сложных проектов. Встроенные функции для работы с HTTP, базами данных, файлами.

Его недостатки:

Не всегда высокая производительность (по сравнению с компилируемыми языками). Гибкость = риск плохого кода – допускает "спагетти-код" без строгих ограничений. Устаревшие черты в legacy-версиях (до PHP 8.0).

## 3.1 Можно ли его заменить?

Да, но выбор альтернативы зависит от задач. Основные конкуренты:

### 3.1.1 JavaScript (Node.js)

Плюсы:

Единый язык для фронтенда и бэкенда.

Асинхронная модель (быстрая обработка запросов).

Огромная экосистема (Express.js, NestJS).

Минусы:

Сложнее в настройке, чем PHP + Apache.

Менее удобен для традиционных CMS (например, WordPress).

### 3.1.2 Python (Django, Flask)

Плюсы:

Чистый и читаемый синтаксис.

Мощные фреймворки для сложных приложений.

Лучше подходит для AI/ML и анализа данных.

Минусы:

Медленнее PHP в некоторых веб-сценариях.

Хостинг-поддержка менее распространена.

### 3.1.3 Ruby (Ruby on Rails)

Плюсы:

Быстрая разработка (Convention over Configuration).

Элегантный код.

Минусы:

Меньше вакансий и хостингов, чем для PHP.

Снижение популярности в последние годы.

### 3.1.4 Go (Golang)

Плюсы:

Высокая производительность.

Простота многопоточности.

Минусы:

Меньше готовых решений для типовых веб-задач.

# 4. Ознакомление с различными SQL СУБД

Веб-разработка требует надежных систем управления базами данных (СУБД), способных эффективно работать с динамическим контентом. Рассмотрим 4+ популярных SQL-СУБД, разделив их на подходящие и нецелесообразные для веб-приложений.

## 4.1 Подходящие для веб-разработки

### 4.1.1 MySQL / MariaDB

Плюсы:  
Оптимизирована для веба – используется в 90% веб-проектов (LAMP-стек). Высокая скорость на операциях чтения/записи. Бесплатная (MariaDB – open-source форк MySQL). Интеграция с PHP, Python, Node.js. Поддержка хостингами (включая дешевые shared-серверы).

Минусы: Ограниченная масштабируемость (без шардинга). Нет полноценной поддержки JSON (как в PostgreSQL).

Когда использовать:

WordPress, Joomla, Drupal.

Стартапы и малые/средние проекты.

### 4.1.2 PostgreSQL

Плюсы:  
Расширенные возможности SQL (сложные запросы, оконные функции). Поддержка JSON/NoSQL-функций.Надежность и ACID-совместимость. Лучшая масштабируемость, чем у MySQL.

Минусы: Требует больше ресурсов.Менее распространен на дешевых хостингах.

Когда использовать:

Сложные веб-приложения (финансовые системы, аналитика).

Проекты с геоданными (PostGIS).

### 4.1.3 SQLite

Плюсы: Встраиваемая – не требует сервера, хранит данные в файле. Простота – идеальна для тестов и малых проектов.

Минусы: нет многопользовательского доступа. Медленная работа при высокой нагрузке.

Когда использовать:

Локальные приложения (например, мобильные бэкенды).

Прототипирование.

## 4.2 Нецелесообразные для веб-разработки

### 4.2.1 Microsoft SQL Server

Почему не подходит для большинства веб-проектов: Платный (бесплатна только Express-версия с ограничениями). Привязка к Windows (хотя есть Linux-версия). Избыточен для типовых веб-сайтов.

Когда использовать:

Корпоративные приложения на .NET.

Проекты, требующие интеграции с Active Directory.

### 4.2.2 Oracle Database

Почему не подходит: очень дорогая лицензия. Сложная настройка для небольших проектов. Избыточная функциональность для простых сайтов.

Когда использовать:

Крупные enterprise-системы (банки, ERP).

# 5. Поверхностное ознакомление с NoSQL СУБД (Несколько примеров)

NoSQL (Not Only SQL) — это класс нереляционных баз данных, которые отличаются от традиционных SQL-систем гибкой схемой данных, горизонтальной масштабируемостью и оптимизацией под специфические задачи (большие объемы данных, высокая нагрузка, неструктурированные данные).

## 5.1 Документоориентированные СУБД:

MongoDB: Одна из самых популярных NoSQL СУБД. Хранит данные в формате документов, похожих на JSON, что обеспечивает гибкость схемы и упрощает работу с данными.

Примеры использования: Управление контентом, каталоги продуктов, мобильные приложения, аналитика в реальном времени.

Couchbase: Документоориентированная база данных, оптимизированная для интерактивных веб-, мобильных и IoT-приложений.

Примеры использования: Управление пользовательскими профилями, каталоги, мобильные приложения.

Amazon DocumentDB (with MongoDB compatibility): полностью управляемая, масштабируемая и высокодоступная документоориентированная база данных, совместимая с MongoDB API.

Примеры использования: Миграция существующих MongoDB-приложений в облако AWS, создание новых документоориентированных приложений.

## 5.2 Ключ-значение СУБД:

Redis: Быстрая, в памяти база данных, используемая как кэш, брокер сообщений и база данных. Поддерживает различные структуры данных, такие как строки, списки, хеши, множества и отсортированные множества.

Примеры использования: Кэширование сессий, рейтинги в реальном времени, очереди сообщений, аналитика.

Memcached: Распределенная система кэширования в памяти, используемая для ускорения динамических веб-приложений за счет снижения нагрузки на базу данных.

Примеры использования: Кэширование результатов запросов к базе данных, кэширование веб-страниц.

Amazon DynamoDB: полностью управляемая NoSQL база данных, предлагающая надежную производительность в любом масштабе.

Примеры использования: Игровые приложения, рекламные технологии, розничная торговля.

# 6. Процессы установки (Задание)

На рисунке 1 мы перекидываем папку апачи-сервера в удобную нам директорию

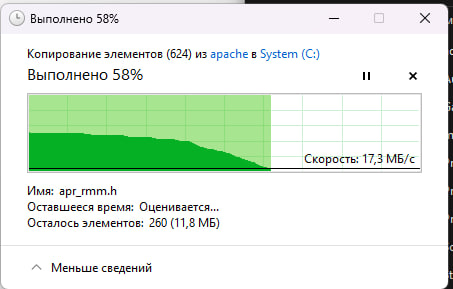


Рисунок 1 – Apache

На рисунке 2 мы повторили за кадром с php то же самое, что и с apache, и уже подключили его к нашему серверу.

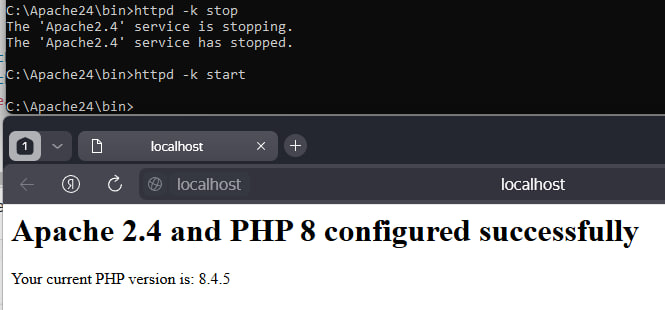


Рисунок 2 – Демонстрация работы Apache и PHP

На дальнейших рисунках 4–13 будет процесс установки MySQL.

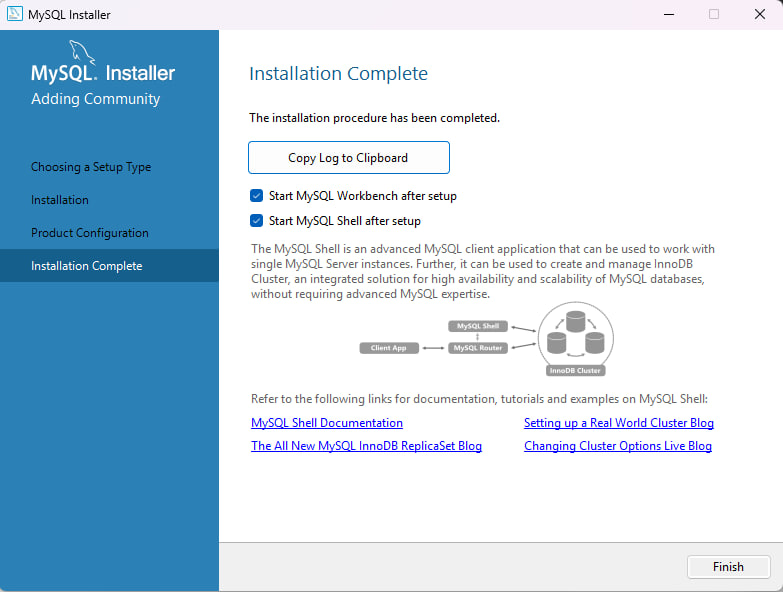


Рисунок 3

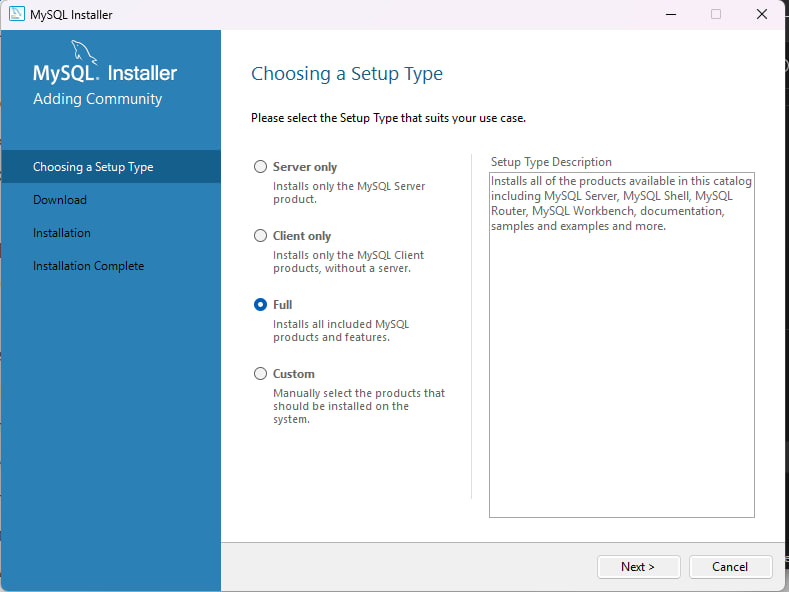


Рисунок 4

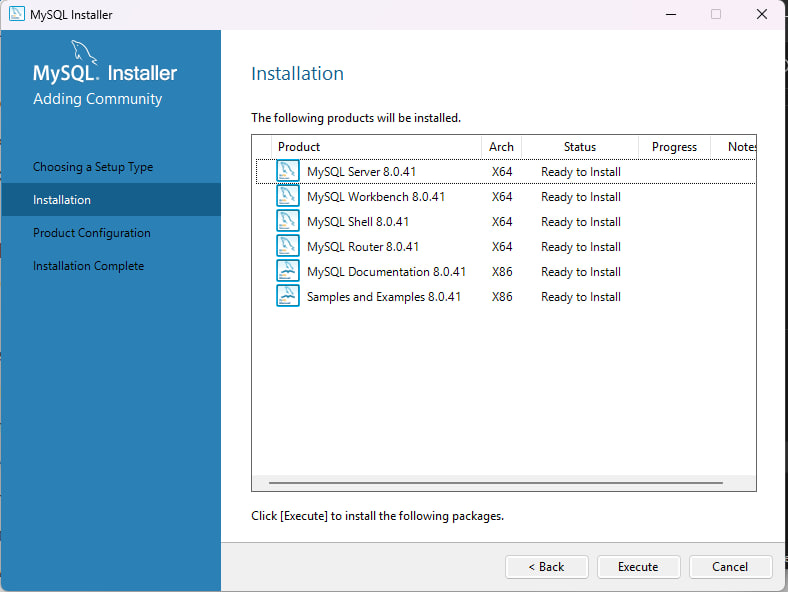


Рисунок 5

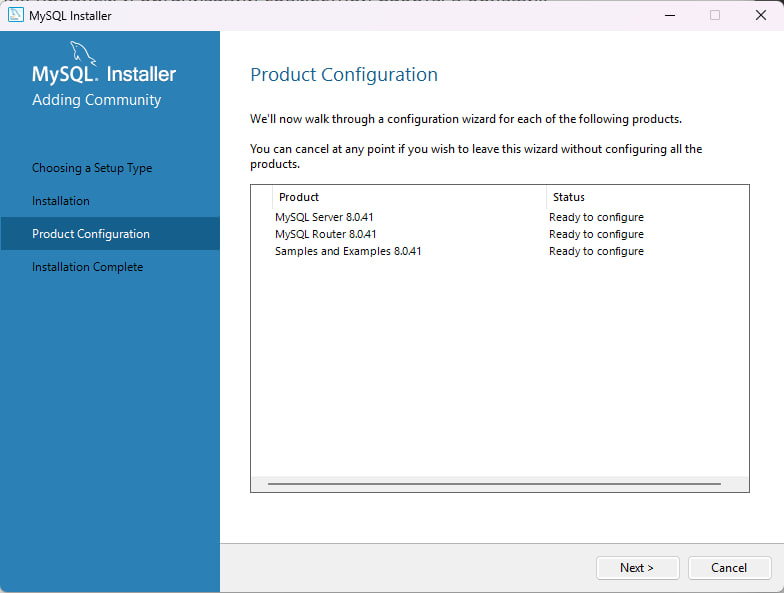


Рисунок 6

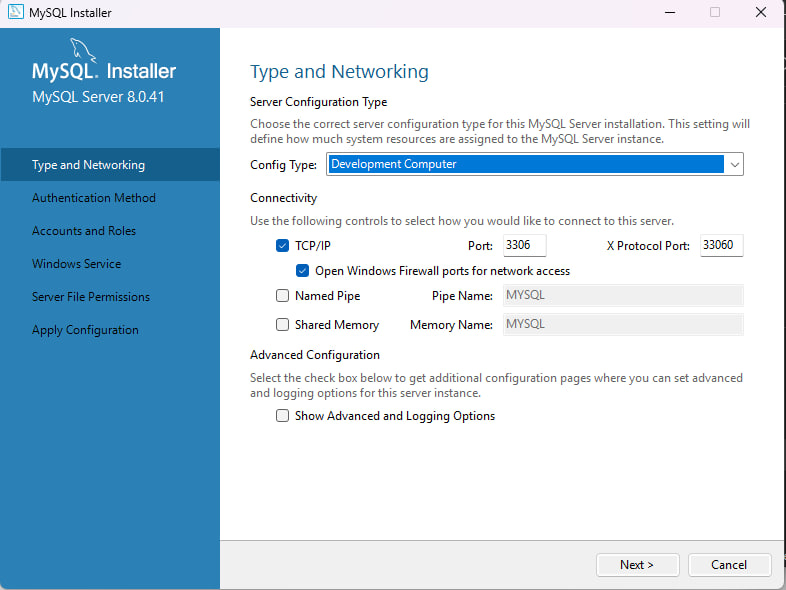


Рисунок 7

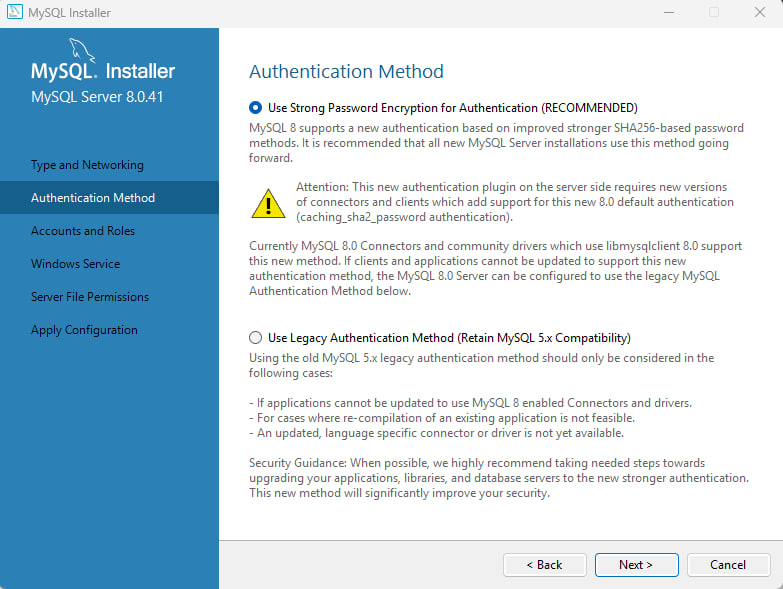


Рисунок 8

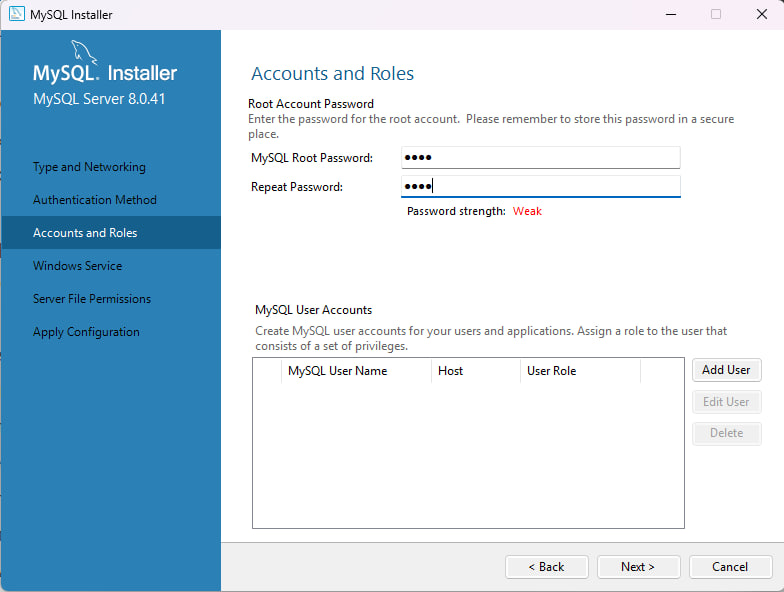


Рисунок 9

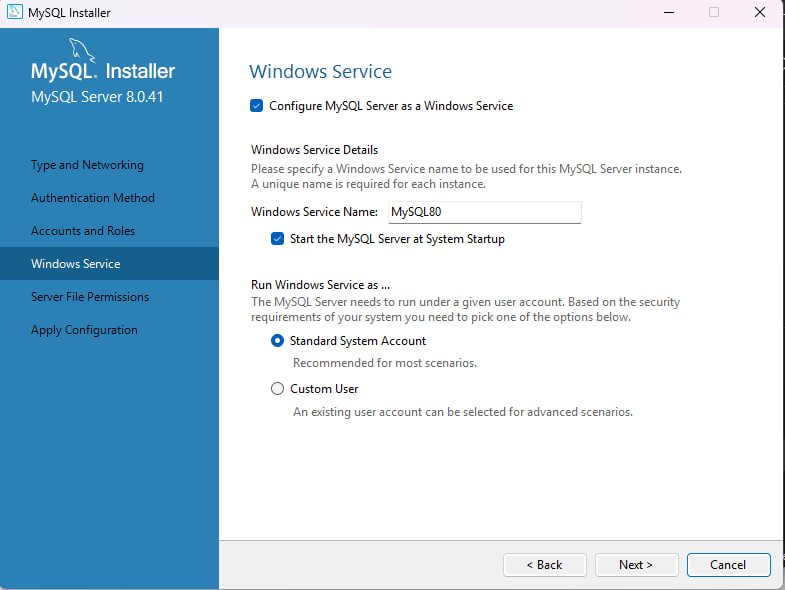


Рисунок 10

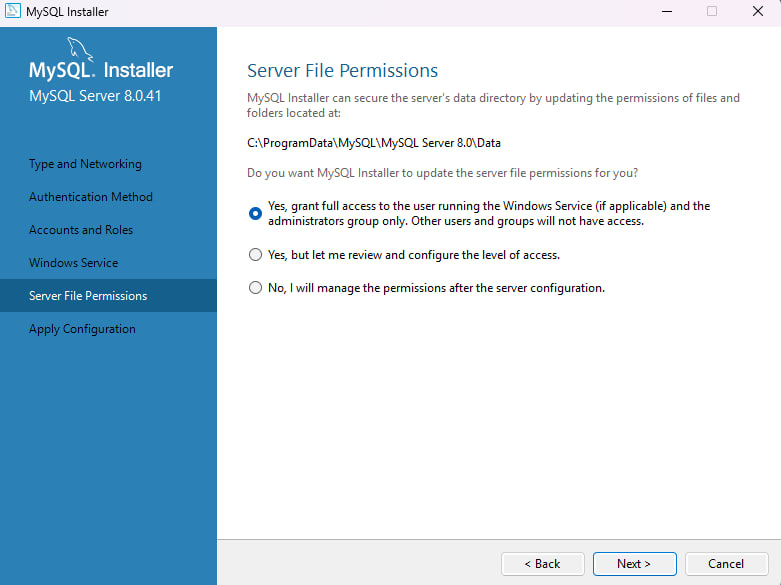


Рисунок 11

Изображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 12

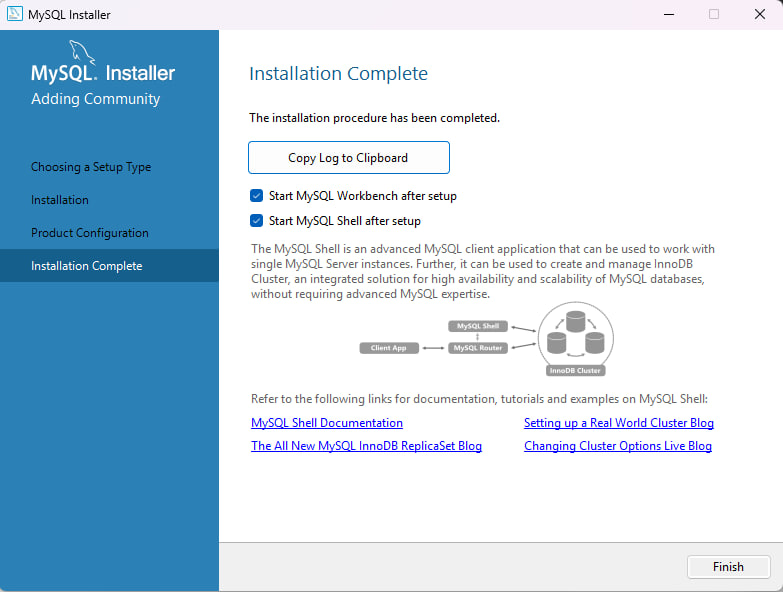


Рисунок 13

На рисунке 14 показан рабочий интерфейс phpMyAdmin’a

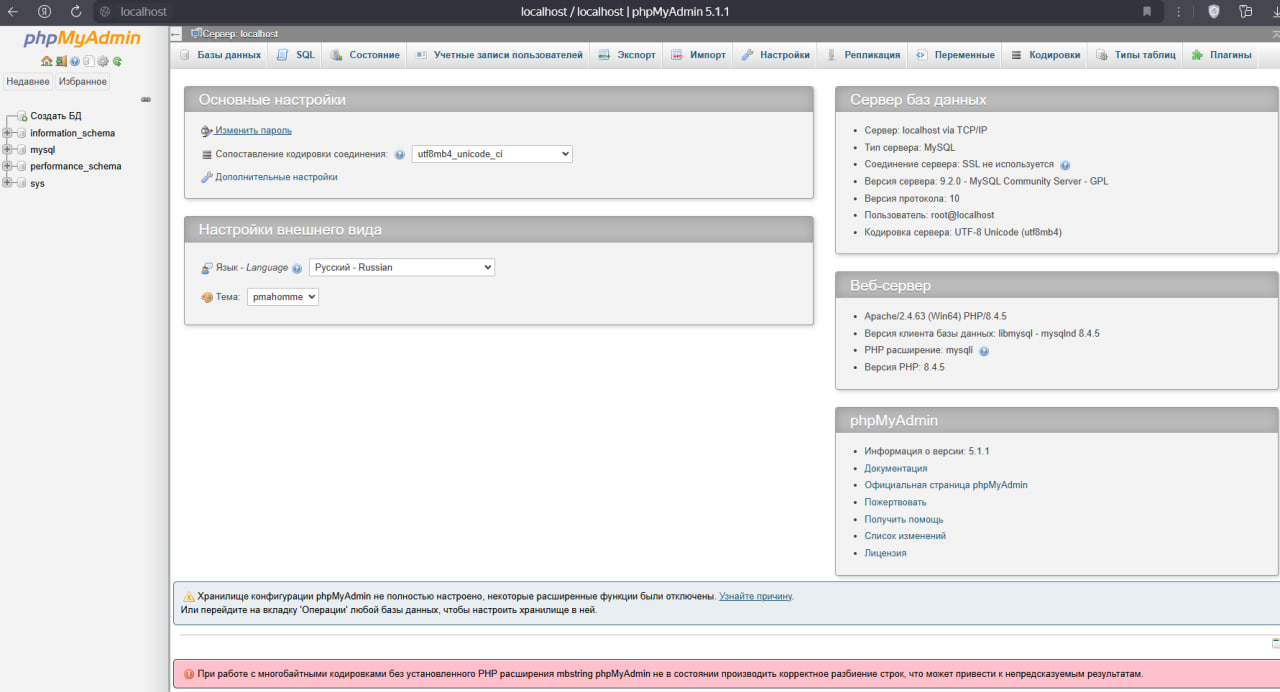


Рисунок 14 – phpMyAdmin работает

# 7. Выводы

В ходе выполнения задания были рассмотрены ключевые аспекты развертывания и настройки веб-серверного окружения, включая теоретическое изучение технологий и практическую установку необходимого ПО на Windows 11. Работа позволила получить комплексное представление о стеке технологий, необходимых для запуска веб-сервера, а также закрепить навыки их настройки. Выбранная связка Apache + PHP + MySQL подтвердила свою универсальность для большинства типовых веб-проектов. Для более сложных сценариев можно рассмотреть альтернативы (Nginx, PostgreSQL, NoSQL), но базовый вариант остается оптимальным для быстрого старта.

Выполнение показало, что даже на Windows 11 можно развернуть полноценный веб-сервер, что полезно для обучения, локальной разработки и отладки перед размещением проектов на хостинге.

# 8. Список литературы

1. Apache - https://httpd.apache.org/download.cgi

2. PHP - https://www.php.net/downloads.php

3. phpMyAdmin - https://www.phpmyadmin.net/downloads/

4. mysql - https://www.mysql.com/downloads/