**УСТАНОВКА PGADMIN И POSTGRESQL SERVER**

**Теория:**

1. PostgreSQL Server - серверная система управления реляционными базами данных (СУБД). PostgreSQL поддерживает язык SQL и предлагает множество дополнительных функций, таких как транзакции, расширения, хранимые процедуры и мощные инструменты для работы с данными.
2. pgAdmin - это графический пользовательский интерфейс (GUI) для работы с PostgreSQL.
3. PostgreSQL обладает рядом преимуществ перед другими СУБД благодаря своей расширенной функциональности и гибкости. Одним из ключевых достоинств является возможность создавать пользовательские типы данных, что позволяет лучше адаптировать структуру базы под специфические требования приложения. Кроме того, PostgreSQL поддерживает выполнение скриптов на различных языках программирования, включая Python, благодаря расширению PL/Python, что дает разработчикам возможность создавать сложную бизнес-логику непосредственно внутри базы данных. Если сравнивать с конкурентами, PostgreSQL максимально похож на MySQL, но имеет больший функционал перечисленный ранее.
4. Если работа в pgAdmin покажется неудобной из-за его специфичного интерфейса, рекомендую попробовать альтернативный инструмент — DBeaver. Это мощный и универсальный GUI для управления базами данных, который поддерживает PostgreSQL и множество других СУБД.

Рекомендуемая версия: Community edition, версия 24.3.4 и выше.

Ссылка: <https://dbeaver.io/download/>

**Процесс установки:**

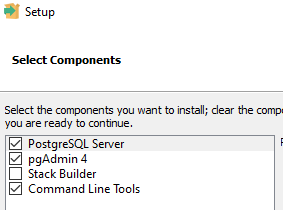
Ссылка: <https://www.postgresql.org/ftp/pgadmin/pgadmin4/v8.14/windows/>

1. Скачайте pgadmin4-8.14-x64.exe файл.

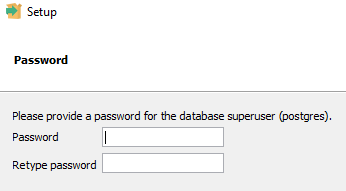
A screenshot of a browser

AI-generated content may be incorrect.

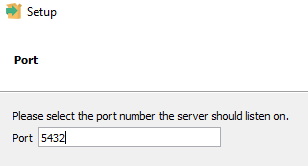
1. Запустите pgadmin4-8.14-x64.exe файл, выберете пути установки и выберите следующие пункты для установки (pgAdmin и PostgreSQL Server):



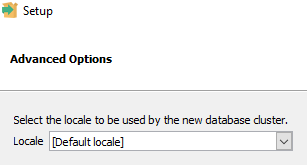
1. Укажите пароль для пользователя postgres (обязательно укажите пароль и сохраните его чтобы не потерять, он нужен для подключения к БД):



1. Укажите порт для сервера БД:



1. Оставьте default locale:



По итогу у вас установится pgAdmin и PostgreSQL Server.

1. Необходимо создать базу данных в pgAdmin, откройте pgAdmin 4 и нажмите в первом окне на servers:



1. Далее отобразится окно подключения к серверу PostgreSQL17, вводите пароль пользователя postgres:

A screenshot of a computer

AI-generated content may be incorrect.

1. Появляется такая иерархия в интерфейсе, необходимо нажать правой кнопкой мыши на Databases, далее нажимае:

A screenshot of a computer

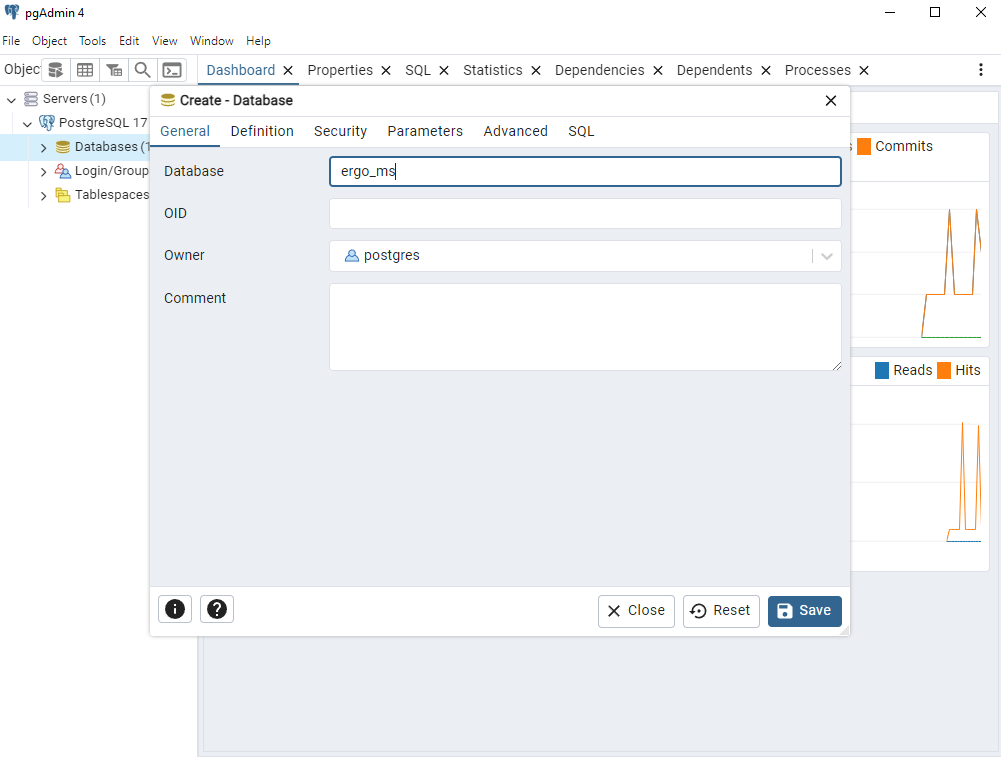
AI-generated content may be incorrect.

1. Нажимаете create, потом database:

A screenshot of a computer

AI-generated content may be incorrect.

1. Введите название базы данных (рекомендую - ergo\_ms) и нажмите Save:



1. По итогу у вас должна создаться база данных и появиться в выпадающем списке Databases:

A screenshot of a computer

AI-generated content may be incorrect.

Установка и настройка базы данных завершена. Создавать таблицы через pgAdmin не требуется. Все таблицы проекта должны создаваться и изменяться исключительно с помощью инструмента миграций Django. В процессе разработки структура базы данных формируется путем описания моделей в файлах models.py и последующего выполнения команд миграции. Более подробное описание работы с базой данных будет приведено в разделе, посвященном API проекта.

По итогу сформированы следующие данные для подключения к БД:

1. Хост: 127.0.0.1, localhost.
2. Порт: 5432 (если не меняли с стандартного).
3. Пользователь: postgres.
4. Пароль пользователя: указанный при инсталляции.
5. Имя базы данных: ergo\_ms.

**УСТАНОВКА PYTHON**

Рекомендуется устанавливать самую актуальную стабильную версию Python, например 3.12.8. Версии 3.11 не рекомендованы к использованию, так как многие разработчики библиотек уже прекратили их поддержку и обновления. Также стоит воздержаться от установки версии 3.13, поскольку она совсем недавно вышла, и множество библиотек еще не адаптированы под нее, что может привести к проблемам совместимости в процессе разработки.

Ссылка: <https://www.python.org/downloads/release/python-3128/>

**УСТАНОВКА NODE.JS**

**Определения:**

1. Node.js - это серверная платформа для работы с JavaScript.
2. npm (Node Package Manager) - это стандартный менеджер пакетов для Node.js. Он позволяет управлять зависимостями и устанавливать сторонние библиотеки для разработки приложений.

**Процесс установки:**

Рекомендованная версия Node.js: 22.11.0 и выше

Ссылка: <https://nodejs.org/en/download/current/>

1. Нажимаете на кнопку Windows Installer (.msi):

A green box with white text

AI-generated content may be incorrect.

1. При установке Node.js рекомендуется использовать стандартные настройки установщика без внесения изменений. После завершения установки важно проверить корректность установки обеих утилит. Откройте командную строку Windows (нажав Win + S и введя cmd, либо комбинацией Win + R и вводом cmd) и выполните команды node -v и npm -v. Если команды выводят версии Node.js и npm, это свидетельствует об успешной установке и готовности инструментов к работе.

Корректный результат выполнения команд:

A black screen with white text

AI-generated content may be incorrect.

**УСТАНОВКА VISUAL STUDIO CODE**

Ссылка: <https://code.visualstudio.com/download>

A screenshot of a computer

AI-generated content may be incorrect.

При инсталяции оставляете все настройки стандартными.

Расширения, которые рекомендую установить:

A black background with white text

AI-generated content may be incorrect.

A black background with white text

AI-generated content may be incorrect.

A black background with white text

AI-generated content may be incorrect.

A screenshot of a black background with white text

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

Самые часто используемые функции Visual Studio Code:

**УСТАНОВКА GIT**

**Теория:**

1. Git это распределенная система контроля версий, которая позволяет разработчикам отслеживать изменения в коде, работать совместно над проектами и управлять разными версиями программного обеспечения.
2. GitHub это облачная платформа для хостинга Git-репозиториев и совместной работы над проектами.

**Процесс установки:**

Ссылка: <https://git-scm.com/downloads>

A screenshot of a computer

AI-generated content may be incorrect.

Рекомендуется использовать практически стандартные настройки при установке Git, за исключением одного важного момента — рекомендуется снять галочку с опции Windows Explorer Integration. Эта настройка добавляет в интерфейс проводника Windows дополнительные кнопки для открытия файлов и папок через Git. Однако данный функционал редко используется на практике и может загромождать интерфейс. Более того, убрать эти кнопки впоследствии непросто — для этого потребуется установка специальных утилит для настройки интерфейса проводника, поиск соответствующих записей в реестре или полная переустановка Git.

Все остальные настройки при установке выбирайте по умолчанию.

Проверьте установку Git следующей командой: git –version.

Рекомендуется зарегистрироваться на [GitHub](https://github.com/) и подключиться в среде разработки (IDE) к аккаунту.

**УСТАНОВКА ПРОЕКТА**

**ТЕОРИЯ И РЕКОМЕНДАЦИИ ПО РАБОТЕ С ПРОЕКТОМ**

API

HTTP запросы

Swagger

Виртуальное окружение

Сервер daphne

.yaml и .env файлы для конфигурации проекта

Django фреймворк

Конфиг Django

Celery

Redis

Работа с БД в Django (Django ORM):

Модели Django

Миграции Django

Вспомогательные классы для работы с БД.

Сериализаторы Django

Представления Django

Команды Django

Poetry

Poetry команды

Генерация методов API при помощи .yaml

**ОБЩИЕ РЕКОМЕНДАЦИИ ПО ПРОГРАММИРОВАНИЮ**

**Лучшие на февраль 2025 года генеративные модели:**

Qwen - <https://chat.qwenlm.ai/>

DeepSeek - <https://chat.deepseek.com/>

Claude - <https://claude.ai/> (работает только с VPN и регистрацией через зарубежную сим-карту)

**IDE с интеграцией ИИ:**

Cursor AI — это интеллектуальная платформа для автоматизации работы с кодом с использованием искусственного интеллекта. Она помогает разработчикам ускорить процесс программирования, предлагая интеллектуальные подсказки и автозавершение кода на основе контекста и анализа.

Ссылка: <https://www.cursor.com/>

Некоторые возможности Cursor AI:

* Автодополнение кода. Платформа может предсказывать продолжение вашего кода и автоматически заполнять части программного кода, основываясь на контексте.
* Cursor AI может анализировать ваш код, находить ошибки и предлагать исправления или оптимизации.
* Интеллектуальные комментарии. Платформа может генерировать описание функций, классов или методов на основе их контекста, что помогает лучше понимать и документировать код.

Cursor AI схож с такими инструментами, как GitHub Copilot, и предоставляет возможности для более эффективной работы, улучшая качество кода и ускоряя процесс разработки.