Необходимо установить следующие зависимости:

* Python (рекомендуется использовать 3.8);
* Pipenv;
* Docker;
* docker-compose;
* cookiecutter.

Установка зависимостей происходит со следующей последовательностью:

*Шаг 1 - Установка Docker и docker-compose.*

su

apt update; apt upgrade -y; apt install -y curl; curl -sSL https://get.docker.com/ | sh; curl -L https://github.com/docker/compose/releases/download/1.28.2/docker-compose-$(uname -s)-$(uname -m) -o /usr/local/bin/docker-compose && chmod +x /usr/local/bin/docker-compose

usermod -aG docker $USER

Затем нажмите Ctrl+D для того, чтобы выйти из учетной записи суперпользователя и перезагрузите компьютер.

*Шаг 2 - Устанавливаем cookiecutter*

Cookiecutter используется для генерации нового сайта из шаблона.

pip3 install cookiecutter

*Шаг 3 - Устанавливаем Pipenv*

Pipenv помогает управлять зависимостями в проекте. Он делает это лучше, чем стандартный pip.

pip3 install pipenv

Если вы на MacOS, то можете использовать brew:

brew install pipenv

Далее создаем новый проект или запускаем существующий. При создании нового проекта требуется воспроизвести следующие действия:

*Шаг 1 - Создать новый проект из шаблона*

Будет создана директория с проектом в интерактивном режиме.

cookiecutter https://github.com/garpixcms/garpixcms-empty-template

cd website



При выполнении этой команды можно указать имя проекта, в нашем случае мы использовали *website*.

В созданном нами директории с проектом мы видим следующую структуру файлов:


*Шаг 2 - Применить переменные окружения*

Необходимо скопировать пример файла с переменными окружениями в новый файл с названием *.env*

Это можно сделать, используя следующую команду(в Linux):
cp example.env .env
Либо просто скопировав и вставив файл через проводник(в Windows)

Загляните в файл *.env*. Вполне возможно, что вы захотите отредактировать содержимое.

*Шаг 3 - Установить зависимые пакеты*

Перейти в директорию с проектом и установить зависимости

pipenv install


Ожидаем завершения установки всех зависимостей(это может занять некоторое время), далее используем команду ниже для перехода в виртуальное окружение.

pipenv shell



*Шаг 4 - Поднять docker*

Поднимаем базу данных и другие сервисы через docker-compose.

docker-compose up -d



*В этом примере мы используем учётную запись root (команда sudo), поэтому при выполнении данной команды нас попросят ввести пароль.*

*Шаг 5 - Создать миграции и аккаунт администратора*

python3 backend/manage.py makemigrations

python3 backend/manage.py migrate

python3 backend/manage.py createsuperuser

python3 backend/manage.py runserver

После выполнения всех команд выше мы видим, что наш виртуальный сервер запущен по адресу http://127.0.0.1:8000


При запуске существующего проекта требуется воспроизвести следующие действия:

*Шаг 1 - Перейти в существующий проект*

Если вы присоединяетесь к проекту, то вам необходимо его клонировать из удаленного репозитория:

git clone <repo\_url>

После клонирования или если сайт уже находится локально, то перейдите в его директорию:

cd <website\_directory>

*Шаг 2 - Применить переменные окружения*

Необходимо скопировать пример файла с переменными окружениями в новый файл с названием *.env*

Это можно сделать, используя следующую команду(в Linux):
cp example.env .env
Либо просто скопировав и вставив файл через проводник(в Windows)

Загляните в файл *.env*. Вполне возможно, что вы захотите отредактировать содержимое.

*Шаг 3 - Установить зависимые пакеты*

Перейти в директорию с проектом и установить зависимости

pipenv install


Ожидаем завершения установки всех зависимостей(это может занять некоторое время), далее используем команду ниже для перехода в виртуальное окружение.

pipenv shell



*Шаг 4 - Поднять docker*

Поднимаем базу данных и другие сервисы через docker-compose.

docker-compose up -d



*В этом примере мы используем учётную запись root (команда sudo), поэтому при выполнении данной команды нас попросят ввести пароль.*

*Шаг 5 - Создать миграции и аккаунт администратора*

python3 backend/manage.py migrate

python3 backend/manage.py createsuperuser

*Шаг 6 - Запустить сервер*

python3 backend/manage.py runserver

После выполнения всех команд выше мы видим, что наш виртуальный сервер запущен по адресу http://127.0.0.1:8000


Если мы перейдем по этому адресу, мы увидим следующую страницу. Она означает, что установка прошла успешно и всё готово для использования:


Для перехода в Админ-панель используем адрес http://127.0.0.1:8000/admin и вводим данные от аккаунта администратора, который мы создали выше:





Для получения консультаций по процессу развёртывания, настройки экземпляра ПО иего функционирования можно написать письмо на электронную почту:info@garpix.com или позвонить потелефону:8-800-234-50-56