Задание 1:

1. Перейдите на официальный сайт Docker по ссылке https://docs.docker.com/get-docker/ и следуйте инструкциям для установки Docker на вашу операционную систему.
2. После установки Docker убедитесь, что Docker работает, запустив команду `docker --version` в терминале или командной строке.
3. Создайте новый файл с именем `Dockerfile` в корневой директории вашего проекта.
4. Добавьте в `Dockerfile:`
   - инструкции по установке в образ утилит vim и htop;
   - наследуйте образ вашего будущего контейнера от базового образа python 3.11;
   - установите рабочую директорию внутри контейнера в директорию ./app/src/;
   - установите переменные окружения;
   - установите poetry;
   - перенесите в образ файлы для устновки зависимостей;
   - установите зависимости с помощью poetry;
   - скопируйте исходный код проекта в контейнер;
   - добавьте команду для запуска проекта
5. Сохраните файл `Dockerfile`.

Задание 2:

1. Создайте новый файл с именем `docker-compose.yaml` в корневой директории вашего проекта.
2. Добавьте в `docker-compose.yaml` файл сервисы web и db:
   1. Web сервис должен наружу предоставить порт 5000, а внутри проект должен быть доступен по порту 8000;
   2. web сервис не должен стартовать пока не поднимится база данных;
   3. web сервис должен использовать образ `Dockerfile` из предыдущего задания;
   4. сервис базы данных должен наследоваться от образа postgres 14 версии;
   5. установите переменные окружения с названием базы данных, имени пользователя и пароль
3. Запустите сервисы и убедитесь, что они работают
4. Для проверки работоспособности базы данных используйте клиент dbeaver (можно использовать любой другой)
5. Для проверки работоспособности web сервиса убедитесь, что по адресу http://127.0.0.1:5000 отдается главная страница проекта

Задание 3:

Добавьте новое приложение players в проект, файл с моделями должен содержать модель **`Players`** с полями:

- id (Primary Key)
- username (Имя игрока)
- email (Электронная почта)
- password (Хэшированный пароль)
- date_joined (Дата регистрации)

Добавьте новое приложение vehicles в проект, файл с моделями должен содержать модель **Tanks** с полями:

- id (Primary Key)
- tank_name (Название танка)
- tank_type (Тип танка: легкий, средний, тяжёлый и т.д.)
- tank_description (Описание танка)
- damage_points (Урон танка)
- armor_points (Броня танка)

- speed (Скорость танка)
- cost (Стоимость танка в игровой валюте)

Добавьте в приложение players модель **`PlayerVehicles`** с полями:

- id (Primary Key)
- player_id (Foreign Key - ссылка на таблицу "Игроки")
- vehicle_id (Foreign Key - ссылка на таблицу "Танки")
- experience_points (Очки опыта для техники)

Добавьте новое приложение achievements в проект, файл с моделями должен содержать модель **Achievements** с полями:

- id (Primary Key)
- player_id (Foreign Key - ссылка на таблицу "Игроки")
- achievement_name (Название достижения)
- achievement_description (Описание достижения)
- date_achieved (Дата достижения)

Добавьте в приложение achievements модель **PlayerRanking** с полями:

- id (Primary Key)
- player_id (Foreign Key - ссылка на таблицу "Игроки")
- ranking_points (Очки рейтинга)

Задание 4:

Подготовьте файлы с миграциями этих данных и примените их на созданную базу данных.

Задание 5:

Подготовьте кастомную manage команду для заполнения таблиц задания 3 данными. Каждая таблица должна содержать 20 записей. Сделайте дамп данных с помощью встроенной manage команды и приложите его в репозиторий.