Задача:

Разработка серверной части простого веб-приложения для управления списком задач с использованием aiohttp, PostgreSQL, Docker Compose и аутентификацией через Bearer-токен (с access и refresh токенами).
Описание:

Серверная часть:
Создайте RESTful API на Python с использованием фреймворка fastapi.
Реализуйте CRUD-функции для задач:
Создание задачи (POST /tasks) с полями: название, описание, статус (например, "в процессе" или "завершена").
Получение списка всех задач (GET /tasks) с возможностью фильтрации по статусу.
Обновление задачи (PUT /tasks/{id}) — редактирование названия, описания и статуса.
Удаление задачи (DELETE /tasks/{id}).

Аутентификация:
Добавьте аутентификацию пользователей по Bearer-токену:
Используйте access и refresh токены для подтверждения операций с задачами.
Маршруты для аутентификации:
Регистрация пользователя (POST /auth/register): принимает username и password.
Вход в систему (POST /auth/login): возвращает access и refresh токены.
Обновление access токена (POST /auth/refresh) с использованием refresh токена.

База данных:
Используйте PostgreSQL и библиотеку asyncpg для хранения информации о пользователях и задачах.
Реализуйте схемы таблиц:
Таблица users: содержит id, username, password_hash.
Таблица tasks: содержит id, название, описание, статус, пользователь_id.
Использование Redis:

Хранение refresh токенов в Redis для управления их сроком жизни и обеспечения безопасности.

Docker Compose:
Настройте приложение для работы в контейнерах с помощью Docker Compose, включая контейнеры для aiohttp-сервера и PostgreSQL базы данных.

Требования:
Асинхронная работа с базой данных и HTTP запросами.
Инструкция по установке и запуску приложения в Docker.