Задача

Реализовать backend для сервиса аренды велосипедов, который предоставляет RESTful API для выполнения основных операций.
Используйте Django и Django Rest Framework. Реализуйте асинхронные задачи с помощью Celery.
Для работы с базой данных используйте Django ORM. Развертывание системы должно осуществляться с использованием Docker и CI/CD (например, GitLab CI).
Для работы с облачными технологиями выберите одного из популярных провайдеров (AWS, Google Cloud, Яндекс.Облако) и реализуйте интеграцию с ним.

Основные функции

Регистрация нового пользователя:

- Реализовать API для регистрации нового пользователя.
- Пользователь должен предоставить информацию: имя, электронную почту и пароль.
- Пароль должен храниться в зашифрованном виде.

Авторизация пользователя:

- Реализовать API для авторизации пользователя.
- Использовать JWT (JSON Web Token) для управления сеансами.

Получение списка доступных велосипедов:

- Реализовать API для получения списка всех доступных велосипедов.
- Учитывать текущий статус велосипеда (доступен или арендован).

Аренда велосипеда:

- Реализовать API для аренды велосипеда.
- Пользователь может арендовать только один велосипед одновременно.
- Учитывать время начала аренды.

Возврат велосипеда:

- Реализовать API для возврата велосипеда.
- Учитывать время окончания аренды и расчет стоимости аренды.

Получение истории аренды пользователя:

- Реализовать API для получения истории аренды велосипедов текущего пользователя.

Дополнительные требования

Асинхронные задачи:

- Используйте Celery для обработки асинхронных задач (например, расчет стоимости аренды).

Тестирование:

- Реализовать модульные тесты с использованием PyTest.
- Реализовать интеграционные тесты для проверки работы API.

Развертывание:

- Используйте Docker для контейнеризации приложения.
- Настройте CI/CD систему (например, GitLab CI) для автоматического тестирования и развертывания.

Интеграция с облачными технологиями:

- Выберите одного из популярных облачных провайдеров (AWS, Google Cloud, Яндекс.Облако).
- Реализуйте интеграцию с облаком для хранения данных или других сервисов (например, отправка уведомлений).

Стек технологий

- **Backend**: Django, Django Rest Framework
- **Асинхронность**: Celery
- **База данных**: PostgreSQL
- **Тестирование**: PyTest
- **Контейнеризация**: Docker
- **CI/CD**: GitLab CI
- **Облачные технологии**: AWS, Google Cloud или Яндекс.Облако

Ожидаемые результаты

- Репозиторий с исходным кодом приложения.
- Документация по API (например, с использованием Swagger или DRF YASG).
- Файл `docker-compose.yml` для развертывания всех сервисов.
- Скрипты CI/CD для автоматического тестирования и деплоя.
- Инструкции по развертыванию и локальному запуску проекта.
- Примеры запросов для тестирования API.