.beam

Тестовое задание: Разработка бэкенда на Django

Общие сведения:

Срок выполнения: 24 часа с момента получения задания
Стек технологий: Django, Django REST Framework, WebSockets (Channels), Swagger/OpenAPI
Деплой: Бесплатный тестовый сервер (Render, Railway, Heroku и др.)
Дополнительно: Docker-контейнеризация, видео-презентация

Технические задачи:

1. Развертывание Django-проекта

- Инициализация нового проекта Django
- Настройка базовой конфигурации
- Подготовка окружения для деплоя

2. Реализация CRUD с различными типами связей

Модели для примера (на выбор):

- Пользователи (User) и Товары (Product)
- Блоги (Post) и Комментарии (Comment)

Типы связей:

- Один к одному (OneToOne)
- Один ко многим (ForeignKey)
- Многие ко многим (ManyToMany)
- Многие к одному (обратная связь)

3. Интеграция WebSockets

- Использование **Django Channels**
- Реализация уведомлений в реальном времени:
    - При создании/изменении сущности (например, нового пользователя)
    - Рассылка сообщений всем подключенным клиентам

4. REST API + Документация

- Реализация API на базе **Django REST Framework**
- Документирование через **Swagger/OpenAPI**
- Подробное описание всех эндпоинтов

5. Веб-интерфейс для тестирования

- Страница с кнопками для проверки CRUD-операций
- Визуализация связей между моделями
- Блок для демонстрации работы WebSockets

Требования к сдаче

1. **Исходный код:**
    - Доступ к **GitHub-репозиторию** с историей коммитов
    - Чистый, структурированный код
    - Подробный README с инструкциями по запуску
2. **Деплой**
    - Развертывание на **бесплатном хостинге** (Render, Railway и др.)
    - Доступ к **админ-панели Django** (логин/пароль)
    - Доступ к **API** (Swagger/OpenAPI)

3. Docker

- Настроенный **Dockerfile**
- Готовый **docker-compose.yml** (при необходимости)
- Контейнер должен быть переносимым

4. Видео-презентация

- **Качественная запись** (хороший звук, четкое изображение)
- Демонстрация:
    - Работоспособности CRUD
    - Связей между моделями
    - WebSockets-уведомлений
- Размещение на **Google Drive** или **YouTube** (открытый доступ)

.beam

Рекомендации

- Используйте **Django ORM** для работы с моделями
- Для WebSockets: **Django Channels + Redis** (брокер сообщений)
- Для API: **Django REST Framework + drf-yasg/swagger**
- Для фронтенда: можно использовать **шаблоны Django** или **простой JavaScript**

Желаем успехов в выполнении задания!