

Тестовое задание: API-сервис для вопросов и ответов

Функциональные требования

Модели:

1. Question – вопрос:

- id: int
- text: str (текст вопроса)
- created_at: datetime

2. Answer – ответ на вопрос:

- id: int
- question_id: int (ссылка на Question)
- user_id: str (идентификатор пользователя, например uuid)
- text: str (текст ответа)
- created_at: datetime

Методы API:

1. Вопросы (Questions):

- `GET /questions/` — список всех вопросов
- `POST /questions/` — создать новый вопрос
- `GET /questions/{id}` — получить вопрос и все ответы на него
- `DELETE /questions/{id}` — удалить вопрос (вместе с ответами)

2. Ответы (Answers):

- `POST /questions/{id}/answers/` — добавить ответ к вопросу
- `GET /answers/{id}` — получить конкретный ответ
- `DELETE /answers/{id}` — удалить ответ

Логика:

- Нельзя создать ответ к несуществующему вопросу.
- Один и тот же пользователь может оставлять несколько ответов на один вопрос.
- При удалении вопроса должны удаляться все его ответы (каскадно).

Технические требования

- Использовать **FastAPI** или **Django**.
- Работа с БД через **ORM**.
- Использовать **PostgreSQL**.
- Использовать миграции.
- Обернуть приложение в **Docker** и запустить через **docker-compose**.

- Приветствуется: логгирование, тесты (**pytest**)
-

 **Что нужно предоставить**

- Ссылку на репозиторий (GitHub/GitLab).
 - **README.md** с инструкцией по запуску (`docker-compose up`) и описанием проекта.
-

 **Критерии оценки:**

- Архитектура проекта.
- Читаемость и качество кода.
- Корректность бизнес-логики (валидация, каскадное удаление).
- Работа с Docker и docker-compose.
- Наличие тестов и миграций (хотя бы одного теста и одной миграции).

 **Частые ошибки:**

1. Не структурированный проект (все в одном или двух файлах, отсутствие модульности).
2. Нечитаемый код (дублирование функций, смешивание бизнес-логики и работы с БД).
3. Отсутствие типизации (игнорирование `type hints` и Pydantic-схем).
4. Отсутствие валидации входных данных (например, пустые тексты вопросов/ответов).
5. Нет описания запуска проекта в **README.md**.