

WorkHub — интеграционный хаб задач и напоминаний

WorkHub — это персональный и командный сервис для сбора задач из разных источников, их маршрутизации и автоматического сопровождения (напоминания, дайджесты, аудит).

Идея проекта — **не ещё один todo-лист**, а слой над задачами: входящие события → правила → проекты → автоматические действия.

Что делает пользователь

Пользователь создаёт проекты и задачи, работает с ними сам или в команде.

Он может:

- вести несколько проектов,
- приглашать других пользователей и управлять их правами,
- получать задачи не только из UI, но и извне (Telegram, webhooks),
- настраивать дедлайны и получать напоминания,
- видеть историю действий и понимать, кто и что сделал.

Ключевая ценность продукта

1. **Единый вход задач**

Задачи могут приходить:

- из веб-интерфейса,
- из Telegram (/add текст),
- через HTTP webhook.

1. **Автоматизация, а не ручная рутина**

Система сама:

- создаёт напоминания по дедлайнам,
- отправляет ежедневные дайджесты,
- обрабатывает повторные события безопасно (idempotency).

1. **Командная работа с контролем доступа**

Каждый проект имеет:

- owner / editor / viewer,
- приглашения,
- аудит действий.

Архитектурно (без деталей реализации)

Frontend:

- Vue / Nuxt,
- auth flow,
- минимальный, функциональный UI без лишней графики.

Backend:

- NestJS,
- чёткое разделение слоёв (controller / service / domain),
- REST API с OpenAPI.

Инфраструктура:

- PostgreSQL,
- Redis + очереди,
- фоновые задачи и cron,
- Docker + nginx,
- деплой на VPS.

Что делает проект “взрослым”

- роли и права (RBAC),
- фоновые задачи и очереди,
- интеграции через webhooks и Telegram,
- идиомпотентность и rate limiting,
- аудит действий,
- тесты (e2e + unit),
- healthchecks и graceful shutdown,
- CI/CD.

Это не демонстрация фреймворков, а **работающая система**, которая:

- принимает события,
- обрабатывает их асинхронно,
- защищена от ошибок и повторов,
- наблюдаема в проде.

Как это выглядит на собеседовании (коротко)

Я сделал fullstack-продукт, который агрегирует задачи из разных источников.

В нём есть роли, фоновые процессы, интеграции и деплой.

Я развивал его несколько месяцев, постепенно добавляя сложности, не переписывая архитектуру.

Это звучит спокойно и уверенно. Без “я пробовал”, “я изучал”, “я смотрел”.