**Стек и статус:**

* **Front-end:** React + TypeScript (Vite), Tailwind, shadcn/ui.
* **Back-end (API):** Python FastAPI.
* **Воркер/бот:** Node.js (Jupiter/Jito интеграции, анти-rug).
* **Очереди/кэш:** Redis (BullMQ).
* **БД:** PostgreSQL (метаданные/логи), для dev можно SQLite.
* **Интеграции:** Groq (LLM), Apify (акторы Twitter/Telegram), QuickNode/Helius (Solana RPC/Webhooks).
* **Запуск:** Docker + Docker Compose; есть скрипты start.sh / stop.sh / restart.sh / update.sh / logs.sh.
* **ENV:** используем .env по шаблону .env.example.

**Хостинг/сервер:**

* **VPS:** Ubuntu 22.04 LTS, **4 vCPU / 8 GB RAM / 80+ GB SSD**, статический IP.
* **Порты:** 80/443 (через Nginx/Caddy), внутренние 3000 (UI), 8000 (API), 6379 (Redis), 5432 (Postgres).
* **Нужно установить:** Docker 24+, Docker Compose v2, Nginx или Caddy, Certbot (либо auto-TLS Caddy).

**Ключи/переменные окружения (пример):**

GROQ\_API\_KEY=\*\*\*

APIFY\_TOKEN=\*\*\*

QUICKNODE\_HTTP=\*\*\*

HELIUS\_API\_KEY=\*\*\*

JITO\_TIP=\*\*\* # при необходимости

JUPITER\_BASE\_URL=https://quote-api.jup.ag

REDIS\_URL=redis://redis:6379

DATABASE\_URL=postgresql://user:pass@postgres:5432/alpha2

SNAPSHOT\_INTERVAL\_SECONDS=60

NODE\_ENV=production

BASE\_URL=https://app.мойдомен.com

**Задачи фрилансера:**

1. Проверить 3 модуля (ИИ-центр, парсер/акторы, веб-интерфейс),
2. Сшить их через текущие REST/события (UI → API → воркер),
3. Подключить Groq, Apify, QuickNode/Helius (ключи дадим),
4. Собрать Docker-ом, развернуть на VPS, настроить Nginx/Caddy + HTTPS,
5. Протестировать e2e: триггеры парсера → mint-адрес → быстрые security-чеки → кнопка Buy (симуляция) → логи,
6. Передать краткую инструкцию и шаблон .env.

**Что сдать на выходе:**

* docker-compose.yml + актуальные Dockerfile
* рабочие start.sh / stop.sh / restart.sh / update.sh / logs.sh
* README.md (деплой, ENV, порты)
* .env.example
* конфиг Nginx/Caddy с TLS