

BACKEND / FULLSTACK

Fullstack-разработчик. Сильная сторона в backend.

Пишу на Python (Django REST Framework) и TypeScript (NestJS, React, Next.js), данные храню в PostgreSQL. Последний год в блокчейне: токен и часть биржи на Ethereum и BNB, маркет-мейкер на Solana. Встраивал LLM в продукты. В одиночку сделал мультитенантный SaaS и музейный экспонат с компьютерным зрением.

локация

Грозный, клиенты по миру

опыт

~6 лет, самозанятый с 08.2020

формат

штат, контракт или проект · удалённо

контакт

Telegram @mdikiy

6 лет в разработке	14 запущенных проектов	1 000 000 ₽ грант «Студенческий стартап»
------------------------------	----------------------------------	--

ОПЫТ И СТЕК

10.2025 — 07.2026	REM EcoTech, Норвегия backend-разработчик Выпустил токен на Ethereum, писал часть биржи на Ethereum и BNB, сделал сайт компании. Node.js · TypeScript · Ethereum · BNB Chain
12.2021 — 10.2025	ЧГУ им. А. А. Кадырова fullstack-разработчик Сайты университета, квизы и тесты, внутренние сервисы и автоматизация работы с данными. Интерактивные экспонаты для музея. Django REST Framework · React + TypeScript · PostgreSQL · Docker
backend	Python / Django REST Framework, Celery, Node.js / NestJS, PostgreSQL, Redis
frontend	TypeScript, React, Next.js, MobX
web3	Ethereum, BNB Chain, Solana
ai	интеграции с LLM, AI-агенты
инфраструктура	Docker, Linux, Git, Vercel
прочее	компьютерное зрение, OpenCV, AprilTag

ПРОЕКТЫ

01	Edalink За 15 минут у заведения появляется онлайн-меню с приёмом заказов по ссылке из профиля. Мультитенантный SaaS: изоляция данных арендаторов, автовыдача поддомена с провижинингом SSL, беспарольный вход по email, подписки с пробным периодом и деградацией функциональности при неоплате, лимиты заказов по тарифу, конструктор темы, стоп-лист и промокоды в реальном времени, генерация заказа в deeplink мессенджера. стек: Next.js · TypeScript · мультитенантность · биллинг · роль: единственный разработчик · статус: запущен
02	Маркет-мейкер на Solana Торговая система на Solana: тысячи кошельков и сделки на 1 млн \$ меньше чем за 5 минут. Генерация и управление кошельками пачками до 10 000. Выпуск токенов, торговые алгоритмы, снайпинг. Очередь транзакций с приоритетом и контролем срока жизни blockhash, пул RPC-провайдеров с переключением по задержке. Идемпотентность по ключу операции и сверка с цепочкой, чтобы ни одна операция не прошла дважды. Лимиты на объём и убыток, аварийный стоп. стек: NestJS · TypeScript · React · Solana · роль: бэкэнд и панель · статус: закрытая платформа

- 03 **Хехагхо**
Онлайн-школа подготовки к ЕГЭ, где заявка приходит менеджеру уже с результатами теста.
Движок тестов: хранение вопросов, вариантов и весов, подсчёт прогнозного балла по разделам. Каталог преподавателей с записью на занятия, роли и личные кабинеты, захват лида с передачей менеджеру, пиковая нагрузка в сезон набора.
стек: Django · React · роль: fullstack · статус: запущен
-
- 04 **Langify**
Платформа, где клиенты ищут лингвистов по специализации.
React SPA: каталог лингвистов с поиском по специализации и профили, которые лингвисты заполняют сами.
стек: React · SPA · роль: fullstack · статус: запущен
-
- 05 **Проекционная книга**
Стоит в музее Чеченского государственного университета им. А. А. Кадырова. Посетитель берёт книгу и листает, объяснять ничего не нужно.
Совмещение трёх систем координат: камера, проектор и движущаяся книга. Маркеры AprilTag на разворотах дают страницу и положение плоскости, гомография из плоскости страницы в кадр проектора рисует контент заранее искажённым. Калибровка камера–проектор один раз при установке. Позиция сглаживается по времени с порогом на резкое движение, при перекрытии маркеров держится последняя валидная, смена разворота — по фиксации на нескольких кадрах. Автозапуск, сторожевой процесс, режим ожидания.
стек: Python · OpenCV · AprilTag · роль: разработка экспоната · статус: в экспозиции
-
- 06 **Автоматизация процессов ЧГУ**
Четыре года делал для университета сайты, квизы и внутренние сервисы.
Сложнее всего было то, что работающий процесс нельзя останавливать. Импорт исторических таблиц, где одна сущность записана тремя способами, нормализация, ручное разрешение конфликтов, параллельная работа со старым порядком. Права проверяются на уровне выборки: видимость зависит от подразделения, должности и периода. Единый бэкенд на DRF, несколько фронтендов под разные аудитории, общая аутентификация, Docker.
стек: DRF · React · PostgreSQL · Docker · роль: fullstack · статус: внутренние системы
-
- 07 **Юнидеск**
Победитель федерального конкурса «Студенческий стартап», грант 1 000 000 ₽.
Модульная электронная среда для вузов, в разработке. Под проект зарегистрировано ООО «Нодус».
стек: Django REST Framework · React · PostgreSQL · роль: основатель, разработка · статус: в разработке
-

КАК Я ПРИНИМАЮ РЕШЕНИЯ

- 01 Начинаю со схемы данных. Фреймворк выбираю потом, под неё.
- 02 Заранее решаю, что будет при сбое и повторном запросе. В маркет-мейкере одна продублированная транзакция стоила бы реальных денег.
- 03 Права доступа проверяю в запросе к базе, а не прячу кнопку в интерфейсе.
- 04 Новую систему запускаю рядом со старой и выключаю старую, когда люди привыкнут. Так переводил процессы в университете.
-