



АЛЕКСЕЙ
МАЗУРЕНКО
FULL-STACK DEVELOPER

EMAIL newiqa@gmail.com
ТЕЛ. +380 63 588 3160
САЙТ oleksiimazurenko.dev
GITHUB github.com/oleksiimazurenko
LINKEDIN linkedin.com/in/oleksii-mazurenko
TELEGRAM @oleksii_mazurenko
ФОРМАТ Remote

НАВЫКИ

Основной стек

Языки:
TypeScript, JavaScript
Разметка и стили:
HTML, CSS/SCSS
Ядро:
React 19 · Next.js — App Router, RSC, SSR/SSG/ISR, Server Actions, Route Handlers, proxy (middleware)
Стили / UI:
TailwindCSS · shadcn / Base UI · Framer Motion · GSAP (простые анимации) · Lexical · dnd-kit
Состояние и данные:
Zustand · SWR · TanStack Query · Zod · Prisma · MongoDB (Atlas) и PostgreSQL на уровне CRUD
Архитектура:
модульная структура под App Router · проектирование схемы данных (Prisma)
SEO / i18n:
sitemaps, robots.txt, метатеги, JSON-LD, canonical, aria, семантическая разметка · next-intl · Lingui
Аналитика:
Amplitude (навешивание событий) · собственная система сбора событий · Faro (Grafana) · oxiint · Vercel
AI-разработка:
Claude Code ежедневно, агенты со скиллами под рутину, MCP-интеграции

Интеграции

Solidgate · Stripe · Better Auth · Firebase Auth · GrowthBook · PostHog · Strapi CMS · Payload CMS · AWS (Lambda, SST + OpenNext) · Hetzner · OVH

Также работал с

Strapi 4/5 + плагины · билд-скрипты и кодоген · Playwright (визуальная регрессия) · pnpm · Turborepo · CI/CD · Solid.js · Svelte

КАК Я РАБОТАЮ

1 · Сначала убрать задачу, потом ускорять

- Делает агент — задача больше не приходит ни к кому, агент выполняет её сама: авторевью кода, аналитика сама падает в Slack.
- Делает заказчик — задача не доходит до меня, заказчик выполняет её вместе с AI: аутрич, платформа инфлюенсеров.
- Делает команда — плагин, через который команда меняет сама: редиректы в CMS для SEO-команды.

2 · Когда делаю сам

- Рутинный паттерн — сначала пишу под него скилл.
- Plan mode перед кодом: сначала границы задачи и план, потом реализация — сам или с агентом, в зависимости от цены ошибки.

3 · Конвейер

dev → build → юнит-тесты + Playwright (агент, результат читаю я) → стейдж (мой ручной проход) → тестировщик → PR (агенты со скиллами по зонам) → прод → финальная проверка

Основа: OOP · SOLID · DRY · KISS · YAGNI · CI/CD + строгая последовательность шагов, описанная здесь: oleksiimazurenko.dev/pipeline

ЯЗЫКИ

Украинский — родной

Русский — свободно

English — A2: техническую документацию читаю, разговорный подтягиваю ежедневно

FULL-STACK DEVELOPER

REACT / NEXT.JS

Full-Stack разработчик, 5+ лет в React / Next.js; серверная часть — в пределах Next.js (Server Actions, Prisma). Оцениваю задачи по бизнес-ценности и веду свою часть — от решения до продакшена. Первый год в Promova делал продукт руками, дальше добавил в работу AI-агентов: рутину и часть реализации пишу с ними, а постановка, архитектура, ревью и ответственность за прод остаются моими.

ОПЫТ РАБОТЫ

Promova (promova.com)

Апр 2025 — Авг 2026 · Remote

- Цель: держать систему бизнес-воронки и публичных SEO-страниц быстрой и управляемой для команд; поддерживать web2app-путь и автоматизировать рутину.
- Продуктовая работа (первый год — сам). Миграция на App Router, Lingui, TanStack Query, оптимизация запросов и лишних ререндеров. Воронки, вёрстка секций, текущие задачи и фиксы.
- Web2app-переход. Страница-транзит между рекламой и App Store: перехватывал результат AppsFlyer Smart Script (OneLink) через assessor-ловушку на window — редирект в момент записи, без опроса, плюс фолбэк-таймер и собственная сборка атрибуции.
- Аналитика. События Amplitude по всей воронке — от входа с рекламы до оплаты и перехода в приложение; анализ, где именно отваливаются пользователи.
- Платежи. Интеграция Solidgate и Stripe по документации — подключил и довёл до рабочего состояния.
- Частые UI-сущности на чистом CSS. Инициировал ресёрч: модалка, аккордеон, слайдер, fit-text — что из этого можно без JS. Подход лёг в основу openсорс-пакета (см. React Patterns).
- Статические страницы для динамических роутов. Скрипт, генерирующий их перед деплоем. Решил проблему Turbopack: страница тянет только свои чанки CSS и JS, а не всё подряд.
- A/B-тесты из layout в прокси — чтобы страница оставалась статической.
- Очередь событий аналитики. Между загрузкой страницы и инициализацией Amplitude юзер успевал кликнуть — событие терялось. Предложил очередь, которая накапливает события и воспроизводит после инициализации.
- Ревью PR по зонам проекта (агенты Claude). Агент определяет, какой части системы касается изменение — оптимизация, платежи, аналитика — и применяет соответствующий набор правил проверки.
- Аутрич инфлюенсерам — автоматизированная подготовка и рассылка писем вместо ручной работы.
- Yarn → pnpm (карантин, изоляция пакетов, скорость) · Redux → Zustand · Strapi-плагины — определял потребность, вёл реализацию, тестировал.
- Результат: от выполнения задач перешёл к постановке и ведению решений. Выстроил процессы, где контент-команда и AI-автоматизация делают максимум самостоятельно.

Southern Core (Outsource, NDA)

Сен 2024 — Фев 2025 · Remote

- Цель: привести проекты к современному стеку и продакшн-качеству — архитектура, SEO, стабильность.
- Миграция Pages Router → App Router; архитектура на Feature-Sliced Design.
- Обработка ошибок — добавил адекватный хендлинг там, где его не было. UX — исправлял недостатки, восстанавливал логику понятного интерфейса. API Request Batching — минимизация запросов клиент↔сервер.
- Техническое SEO и доступность: robots.txt, sitemaps, метатеги, JSON-LD, canonical, alt и title, aria. Локализация — сам внедрял next-intl.
- Code review для младших разработчиков — инкапсуляция сущностей, модульность, баланс переиспользования компонентов.
- Инфраструктура: Next.js (ISR) + Directus на OVH; деплой через SST + OpenNext на AWS Lambda; CRON-автоматизация; A/B через PostHog; CI/CD.
- Результат: быстрые страницы (ISR + batching), низкая стоимость инфраструктуры, контент управляется через Directus без разработчика.

Landee (landee.app)

Авг 2023 — Авг 2024 · Remote

- Цель: конструктор персональных сайтов (MVP) с нуля.
- Архитектура. Планировал структуру кода и схему БД; MVP на Next.js (App Router), масштабируемая FSD-архитектура.
- Блочный редактор на Lexical (rich-text: списки, таблицы, код) с drag-and-drop (dnd-kit, GSAP, shadcn/Radix).
- Данные. MongoDB Atlas через Server Actions + Prisma; полный CRUD; офлайн-устойчивая синхронизация — очередь изменений с откатом. Собственная аналитическая система — сбор кликов и показов, свои графики.
- Публикация и онбординг. Публикация страниц (ISR), онбординг с шаблонами; аутентификация (Clerk). Локализация — сам внедрял next-intl (uk/en).
- Серверная часть. Cron-роуты на Vercel; монорепо на Turborepo; интеграция со сторонними API (фоновые фото и видео).
- Фронтенд и качество. TailwindCSS; анимации GSAP (простой сложности); UI/UX. Обработка ошибок; минимизация useEffect. SEO: sitemaps, robots.txt, метатеги, JSON-LD.
- Результат: запущенный MVP конструктора — редактор, публикация, онбординг, аналитика.

ЛИЧНЫЕ ПРОЕКТЫ · oleksiimazurenko.dev

- Sortune — менеджер библиотеки YouTube Music: AI-агенты через MCP классифицируют жанры и сами раскладывают плейлисты по сезону и стилю. Спроектировал и запустил.
- Flint — платное macOS-приложение в продаже: освобождает место на диске от облачных сервисов (iCloud, Google Drive, OneDrive, Dropbox).
- React Patterns — openсорс-библиотека в npm (@oleksiimazurenko/react-patterns): accordion, dialog, popover, slider, parallax, reveal, scroll-progress, sticky-shrink, fit-text, analytics. CSS-first, server-first, делегирование событий.
- oleksiimazurenko.dev — собственный сайт на Solid.js: минимум JS, чистый CSS, быстрая загрузка.

ВОЛОНТЁРСТВО

- Poludne (poludne.com) — платформа об украинской литературе: цитаты, авторы, каталог произведений. Разработка на Svelte, в команде, на волонёрских началах.