



OLEKSII
MAZURENKO
FULL-STACK DEVELOPER

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

SKILLS

Core stack

Languages:
TypeScript, JavaScript

Markup & styles:
HTML, CSS/SCSS

Core:
React 19 · Next.js — App Router, RSC, SSR/SSG/ISR, Server Actions, Route Handlers, proxy (middleware)

Styling / UI:
TailwindCSS · shadcn / Base UI · Framer Motion · GSAP (simple animations) · Lexical · dnd-kit

State & data:
Zustand · SWR · TanStack Query · Zod · Prisma · MongoDB (Atlas) and PostgreSQL at CRUD level

Architecture:
modular structure for App Router · data schema design (Prisma)

SEO / i18n:
sitemaps, robots.txt, meta tags, JSON-LD, canonical, aria, semantic markup · next-intl · Lingui

Analytics:
Amplitude (event instrumentation) · own event-collection system · Faro (Grafana) · oxlint · Vercel

AI development:
Claude Code daily, agents with skills for routine work, MCP integrations

Integrations

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

Also worked with

Strapi 4/5 + plugins · build scripts and codegen · Playwright (visual regression) · pnpm · Turborepo · CI/CD · Solid.js · Svelte

HOW I WORK

1 · First remove the task, then speed it up

- Agent-run — the task never comes back to anyone: an agent runs it end to end (auto code review, analytics posted into Slack).
- Requester-run — the task never reaches me: the requester does it with AI (outreach, influencer platform).
- Team-run — a plugin the team changes on its own: CMS redirects owned by the SEO team.

2 · When I do it myself

- A recurring pattern — I write a skill for it first.
- Plan mode before code: scope and plan first, then the implementation — myself or with an agent, depending on the cost of a mistake.

3 · Pipeline

dev → build → unit tests + Playwright (agent-driven, read by me) → staging (my own hands-on pass) → QA → PR (agents with skills per project zone) → production → final check

Foundation: OOP · SOLID · DRY · KISS · YAGNI · CI/CD + a strict order of steps, described at: oleksiimazurenko.dev/pipeline

LANGUAGES

Ukrainian — native

English — A2: I read technical docs; speaking is improving daily

FULL-STACK DEVELOPER

REACT / NEXT.JS

Full-Stack Developer, 5+ years in React / Next.js; server side within Next.js (Server Actions, Prisma). I weigh tasks by business value and own my area end-to-end — from the decision to production. My first year at Promova I built the product by hand; after that I brought AI agents into the work: routine and part of the implementation I write with them, while the framing, the architecture, the review and the responsibility for production stay mine.

EXPERIENCE

Promova (promova.com) Apr 2025 – Aug 2026 · Remote

- Goal: keep the system of business funnels and public SEO pages fast and manageable for the teams; maintain the end-to-end web2app path and automate the routine away.
- Product work (first year — on my own). Migration to App Router and re-splitting components for the new Next.js model; Lingui; TanStack Query; query optimisation and fewer needless re-renders. Funnels, building sections, ongoing tasks and fixes.
- Web2app transition. Transit page between the ad and the App Store: intercepted the AppsFlyer Smart Script (OneLink) result with an accessor trap on window — redirect fired the instant it was written, no polling. Fallback timer and my own attribution assembly with a parameter whitelist.
- Analytics. Instrumented Amplitude events across the whole funnel — from the ad click to payment and the hand-off into the app; analysed exactly where users drop off.
- Payments. Integrated Solidgate and Stripe following the docs — wired up and taken to a working state.
- Common UI primitives moved to pure CSS. Initiated the research: modal, accordion, slider, fit-text — which of these work without JS. The approach became the basis of an open-source package (see React Patterns).
- Static pages for dynamic routes. Proposed a pre-deploy script that generates them. Solved the Turbopack issue: a page now pulls only its own CSS and JS chunks instead of everything.
- Moved A/B testing out of the Next layout into the proxy — so the page stays static.
- Analytics event queue. Between page load and Amplitude init a user could click and the event was lost. Proposed a queue that buffers events and replays them once analytics is ready.
- PR review routed by project zone (Claude agents). The agent detects which part of the system a change touches — performance, payments, analytics — and applies the matching review ruleset.
- Influencer outreach — automated preparation and sending of outreach emails instead of manual work.
- Yarn → pnpm (dependency quarantine, isolation, speed) · Redux → Zustand · Strapi plugins — defined the need, drove the work, tested it.
- Result: moved from executing tasks to proposing and driving solutions. Built processes where the content team and AI automation do the maximum on their own.

Southern Core (Outsourcing, NDA) Sep 2024 – Feb 2025 · Remote

- Goal: bring the projects to a modern stack and production quality — architecture, SEO, stability.
- Migrated Pages Router → App Router; re-architected onto Feature-Sliced Design.
- Error handling — added proper handling where there was none. UX — fixed rough edges, restored user-friendly interface logic. API request batching — minimised client↔server round-trips.
- Technical SEO and accessibility: robots.txt, sitemaps, meta tags, JSON-LD, canonical, image alt and title, aria. Localisation — implemented next-intl myself.
- Code review for junior developers — encapsulation, modularity, balanced component reuse.
- Infrastructure: Next.js (ISR) + Directus on OVH; deployed via SST + OpenNext on AWS Lambda; CRON automation; A/B via PostHog; CI/CD.
- Result: fast pages (ISR + batching), low infrastructure cost, content managed through Directus without a developer.

Landee (landee.app) Aug 2023 – Aug 2024 · Remote

- Goal: a personal-site builder (MVP) from scratch.
- Architecture. Planned the code structure and DB schema; MVP on Next.js (App Router), scalable FSD architecture.
- Block editor on Lexical (rich text: lists, tables, code) with drag-and-drop (dnd-kit, GSAP, shadcn/Radix).
- Data. MongoDB Atlas via Server Actions + Prisma; full CRUD; offline-resilient sync — a change queue with rollback. Own analytics system — collecting clicks and impressions, stored in MongoDB, custom charts.
- Publishing and onboarding. Page publishing (ISR), onboarding with templates; auth (Clerk). Localisation — implemented next-intl myself (uk/en).
- Server side. Cron routes on Vercel; Turborepo monorepo; third-party API integrations (background photos and video).
- Frontend and quality. TailwindCSS; GSAP animations (simple ones); UI/UX. Error handling; minimising useEffect — I know where reactivity is needed and where it is not. SEO: sitemaps, robots.txt, meta tags, JSON-LD.
- Result: shipped the builder MVP — editor, publishing, onboarding, analytics.

PERSONAL PROJECTS · oleksiimazurenko.dev

- Sortune — a YouTube Music library manager: AI agents connected over MCP classify genres and organise playlists by season and style. Designed and shipped.
- Flint — paid macOS app on sale: frees up disk space taken by cloud services (iCloud, Google Drive, OneDrive, Dropbox).
- React Patterns — open-source library on npm (@oleksiimazurenko/react-patterns) — grew out of production research: accordion, dialog, popover, slider, parallax, reveal, scroll-progress, sticky-shrink, fit-text, analytics. CSS-first, server-first, event delegation.
- oleksiimazurenko.dev — my own site on Solid.js: minimal JS, clean CSS, fast loads.

VOLUNTEERING

- Poludne (poludne.com) — a platform about Ukrainian literature: quotes, authors, a catalogue of works. Development in Svelte, in a team, on a volunteer basis.