

Dzmitry Masakouski

Senior Frontend Engineer

Warsaw, Poland (Remote — US & EU hours) · dzmitry.masakouski@gmail.com · +48 573 744 139 · [linkedin.com/in/dzmitry-masakouski](https://www.linkedin.com/in/dzmitry-masakouski)

SUMMARY

Frontend engineer with 5 years building production React and TypeScript applications for clients in the US, UK, Denmark, and Poland. Ships features end-to-end — from challenging the requirement and proposing the interface, through implementation and data-layer work, to release and defect triage. Recent experience spans a greenfield React/Vite SaaS platform and an adaptive-learning system serving 25M+ students.

KEY ACHIEVEMENTS

- Cut **~2 seconds** off application load time by eliminating unused JavaScript from the production bundle.
- Built and released major features for **ALEKS**, an adaptive learning platform used by **25M+ students** — several delivered predominantly solo, per written manager feedback.
- Built a **greenfield React + Vite + Mantine SaaS platform** close to from scratch, delivering most of its main product sections end-to-end.
- Delivered **10+ UX improvements** originating outside assigned tickets, several adopted by the design team.
- Delivered features across **4+ client products** in three countries; onboarded and trained **2 engineers**.
- Integrated **AI tooling (Claude, Cursor) into daily development**, and authored the coding rules that keep its output consistent with the codebase.

TECHNICAL SKILLS

Languages	TypeScript, JavaScript (ES6+), HTML5, CSS3, SQL
Frontend	React, React Hooks, Redux Toolkit, TanStack Query, Angular 2+, RxJS, NgRx
UI & Styling	Mantine, Material UI, Ant Design, Storybook, Sass/SCSS, LESS, design tokens, responsive/mobile-first, accessibility (familiar with WCAG/ARIA)
Build & Tooling	Vite, Webpack, Gulp, ESLint, Git, SVN
Backend & Data	REST, GraphQL, Node.js, MySQL, SQL query design, JSON pipelines, backend debugging, Salesforce integration
Testing	Jasmine, unit testing, QA collaboration
Design	Figma (incl. Figma MCP), UI/UX proposal & critique, design-system consistency
AI-assisted development	Claude, Cursor — daily use for implementation, SQL, and documentation; authored project AI coding rules
Practices	Agile/Scrum, code review, estimation & planning, legacy maintenance

EXPERIENCE

Intetics Inc. — Warsaw, Poland

Software Engineer · March 2024 – Present

Contract engineering for two long-running client products. Team of ~15.

Diocesan — React/TypeScript SaaS platform (*June 2025 – Present*) A platform serving many separate organizations from one application: a member-facing app plus an admin console for organization, user, and content management.

- Built the application close to from scratch on Vite, React, TypeScript, and Mantine alongside one senior engineer, establishing project structure and shared conventions.
- **Proposed the project's color and theming architecture** — storing the palette in `theme.ts` and consuming it through generated Mantine variables, rather than exporting CSS variables through a chain of intermediate files. Researched the framework documentation to make the case; the team adopted it, and it made adding new colors simpler and their source easier to trace when debugging.
- **Delivered most of the product's main sections end-to-end** — organization and administrator management, user roles, message and form templates, announcements, and user profiles — designing the forms, validation rules, and user-facing error and success states myself.
- Built a **template constructor** that let non-engineers edit message and form templates directly, removing a recurring engineering dependency. Authored the supporting documentation.
- Overhauled the **Announcements** module — replaced placeholder controls, repaired broken validation paths, and corrected mis-rendered error states.
- Owned **mobile-view alignment** across the product: audited every screen against design specs, fixed discrepancies, filed tickets for larger gaps, and proposed mobile-specific interaction improvements.
- Used **Claude and Cursor daily** for implementation, SQL, and documentation, and authored the project's AI coding rules so generated output stayed consistent with existing architecture and conventions.
- Resolved competing table interactions where checkbox selection, drag-and-drop reordering, and row-click each claimed the same gesture — surfaced the conflict and drove a design change.
- Shipped UX proposals adopted by the design team: duplicating the primary action above the fold for long content, explanatory helper text on ambiguous checkboxes, ordering new role options at the top of lists, contextual empty states in place of premature suggestions, and interface copy rewrites.
- Fixed lint violations and backfilled test coverage left by other contributors.
- Ran a daily defect-triage loop with QA and routed backend-owned issues to the API team.

React · TypeScript · Vite · Mantine · TanStack Query · Storybook · Figma · Cursor/AI tooling · Git · Jira

ALEKS (McGraw Hill) — adaptive learning platform (*March 2024 – June 2025*) AI-driven learning and assessment system used by 25M+ students across Math, Chemistry, Statistics, and Accounting. Large, mature codebase with substantial legacy surface area.

- Built a **JSON-to-database ingestion pipeline** parsing content files with arbitrary subsection nesting, validating file presence and structure, and inserting into the relational schema — reusing internal utilities where they fit and writing new logic where they didn't. An ISBN entry creates the full book record from the source file.
- Designed and built the **book import page**, proposing the interface and interaction model and iterating it through several rounds of stakeholder review.
- Solved the **section-linking algorithm** — computing shift offsets and parent IDs across a hierarchical table of contents — and built a parallel verification TOC to validate every field of the outgoing payload.
- Built an **AI-correlation import feature** with conditional rendering (hidden entirely when no source file exists), a confirmation step, and full correlation-table rebuild on execution.
- Delivered a **reporting feature** that grew from a single export button into a book-selection interface with dynamically generated columns and direct-download delivery once reports outgrew email attachment limits.
- **Migrated Salesforce queries from a proprietary in-house scripting language to JavaScript**, plus the dialog interface around them.
- **Proposed a database schema change** — one row per ISBN rather than array storage — which the team adopted.
- Participated in estimation and planning, cross-reviewed frontend pull requests, and ran feature demos.

JavaScript · HTML · CSS · MySQL · SQL · JSON · REST · Jasmine · legacy in-house frameworks · Git · Jira

Netcracker Technology

Software Engineer · June 2021 – December 2023

Three consecutive telecommunications client engagements.

Nuuday (Denmark) — customer service representative dashboard

- Developed frontend functionality for an analytics dashboard covering customer-satisfaction statistics, alerts and notifications, and CRM integration; integrated RESTful APIs with backend engineers.
- **Debugged and fixed a critical backend bug**, improving project performance and stability.
- Worked within a microservice architecture, troubleshooting issues across service boundaries.
- Configured Webpack for asset bundling; wrote unit tests; performed regular code reviews.

JavaScript · TypeScript · Angular 2 · NgRx · RxJS · Webpack · REST · Git

Netcracker core products & internal libraries

- Developed core library functionality — reusable components, shared UI, and common business logic — consumed by multiple downstream product teams.
- **Boosted application performance by ~2 seconds** by eliminating unused JavaScript from the bundle.
- **Onboarded and trained 2 new team members** into an established codebase.
- Implemented responsive design for cross-device compatibility; participated in estimation and planning; cross-reviewed frontend code.

JavaScript · React · Redux Toolkit · React Hooks · Ant Design · Webpack · RxJS · Sass/SCSS · Node.js

Telefónica UK (O2 UK) — self-service customer platform

- Developed, optimized, and maintained frontend functionality for a self-service platform covering account management, billing, usage monitoring, and service management.
- Refactored frequently repeated code into reusable components.
- Collaborated with the lead frontend developer, backend engineers, and QA in a high-performance Scrum environment.

JavaScript · TypeScript · Angular 2 · LESS · Gulp · RxJS · Material UI · Liferay CMS · SVN · Git

EDUCATION

Belarusian State University of Informatics and Radioelectronics — Minsk, Belarus B.S., Telecommunications · 2017 – 2021 · Specialization: Cybersecurity Engineering in Telecommunications

Continuing education — Database engine fundamentals; unit testing with Jasmine; Microsoft Generative AI for Beginners; web accessibility via internal training and Deque University ARIA resources.

LANGUAGES

Russian (native) · Belarusian (native) · English (B2, daily working language with US and EU teams) · Polish (A2)

ADDITIONAL

Based in Warsaw on a Polish temporary residence permit for business activity, with a registered sole proprietorship (JDG) — **immediately available for B2B / contractor engagements with no permit formalities**. Employment contract also possible with an employer-supported permit change. Comfortable overlapping with US Eastern and Pacific business hours.