Samandar Odilboyev

Tashkent, Uzbekistan • samandarodilboyev@gmail.com • linkedin.com/in/samandar-zarif

SUMMARY

Full-stack developer with **3+ years of commercial experience** (React/TypeScript, NestJS/Prisma/PostgreSQL), specializing in data-heavy tables and admin/permission systems. Cut a table's edit-mode switch from ~10s to under 1s. Also built AI content-generation UI and background workers. Earlier frontend work (2+ years) on real-time systems: traffic control of a capital city.

SKILLS

Languages: TypeScript, JavaScript, HTML5, CSS3, SQL

Frontend: React, Redux, TanStack Query, Formik, Yup, MUI, Tailwind CSS, SASS

Backend: NestJS, Node.js, Supabase, Prisma, REST API design, WebSocket

Databases: PostgreSQL

Tests: Jest, pgTAP (Supabase)

Tools & DevOps: Git, GitHub, GitLab, Jira, Bitbucket, Docker (basic)

Practices: Scrum/Agile, code review, real-time systems (WebSocket, WebRTC), Role-based access control (RBAC)

AI Tools: Cursor, Claude Code

EXPERIENCE

Full-stack Developer — Sigma Software (Remote, Full-time) *- May 2025 – Oct 2026*

MVP media-asset generation platform (AI content generation) for a German automotive client — 17 months

- **Built a bulk-edit table view** for structured item lists with mixed field types (text, date, boolean, key-value), backed by server-side filtering and sorting on JSON fields in PostgreSQL.
- **Reduced edit-mode toggle latency from ~10s to <1s** for large data grids (20–30 rows by 20+ columns with complex dialog-based inputs).
- Led implementation of a **permission-model migration** from per-user rows to group-based access: built the group schema (Prisma/PostgreSQL), APIs, and admin UI; analyzed the existing guard and interceptor logic to scope the cutover and split the remaining work into detailed handoff tickets.
- Developed the UI for AI-driven content generation, persisting in-progress generation state across localStorage, sessionStorage, and Redux.
- Built background workers that surface process-level errors across projects through dashboard indicators.
- Shipped ongoing admin features and fixes, working with PMs, designers, and the client's engineers to scope requirements.

Frontend Developer — BurgutSoft, CityNet (On-site) *- Mar 2024 – Apr 2025*

Traffic Control & Construction Monitoring Systems — *13 months*

- **Sole frontend developer** on the team: built the front-end for **a real-time Traffic Control System,** integrating live sensor and video camera data streams to help operators monitor and optimize traffic flow.
- Developed a Construction Monitoring platform with live camera streaming via WebSocket and WebRTC for real-time visualization.
- Developed a Clock-In application to automate and streamline daily employee attendance tracking within the construction department.

Frontend Developer — Koinot (Remote, Part-time) *- Nov 2021 – May 2023*

Admin dashboards/panels *— 19 months*

Designed and developed a Ticket Management System for a football stadium with integration to external devices.

Developed admin panels for a news website and a multi-feature Telegram bot, improving content management and user interaction.

PROJECTS

GradeBook — multi-tenant grading app for schools (Personal project, 2026)
https://gradebook-project.vercel.app · demo: head@kodlab.test (pswd: demo+password)

React, TypeScript, Supabase, PostgreSQL, Row Level Security (RLS), pgTAP, GitHub Actions

- Designed role-based access control (RBAC) for each school: a permission matrix of roles, actions and scopes (own records, assigned courses, whole school), enforced in PostgreSQL with row-level security rather than only in the UI.
- Tested every role and refused action with 220+ pgTAP database tests, run before each commit and in the CI/CD pipeline; a script of 83 cross-role access attempts is fully refused.
- Built a spreadsheet-style grade sheet: keyboard entry, paste from Excel/Sheets, undo, background saving with rollback when the database refuses a change, and month closing with audited corrections.

Note2Quiz — AI quiz generator (Telegram bot) (Personal project, 2026)
NestJS, TypeScript, PostgreSQL, Prisma, Google Gemini API, Telegraf, i18n (uz/ru/en)

- Built a Telegram bot that turns a photo or PDF of study material into a multiple-choice quiz in under a minute, using schema-constrained structured output (JSON schema, not free-text parsing) to guarantee valid question/option/answer shapes from the LLM.
- Added a pre-flight AI "triage" pass that screens each upload for educational relevance and prompt-injection attempts before any credits are charged, plus retry-with-backoff on transient Gemini errors and automatic credit refunds on failure or when fewer questions were produced than paid for.
- Designed a credit-based monetization model (per-quiz cost + page-based reading fee) with native Telegram Stars in-app purchases.
- Shipped full uz/ru/en localization and DOCX export of generated quizzes, backed by 24 Jest test suites covering the Gemini, payment, and queue-processing logic.

EDUCATION

Bachelor in Management of Information Security @ IIAU - International Islamic Academy of Uzbekistan
Part-time (~8 weeks a year), 4th year out of 5.