Vladislav Prozorov

Fullstack Developer

t.me/mscnever · vprozorovvlad@gmail.com · github.com/vladislavprozorov

Saint Petersburg, Russia · Open to remote · English: B2 (Upper-Intermediate)

SUMMARY

Fullstack developer building web products end-to-end: from PostgreSQL schema and APIs to Vue/React frontends and production deployment. Currently at Fortus (Vue 3, PHP 8, PostgreSQL); also the sole developer of a production Next.js website for a cruise travel agency.

WORK EXPERIENCE

Fullstack Developer

Fortus · Saint Petersburg *Mar 2026 – Present*

Web-based information systems with GIS functionality

- Owned full-cycle feature delivery: from requirements analysis with the business analyst and database design to backend, frontend and QA handoff
- Built Vue 3 + Vuetify interfaces: complex forms with validation and interactive Leaflet maps (WMS layers, object selection and drag-and-drop), letting users work with geodata without switching to third-party GIS tools
- Designed and implemented 100+ PostgreSQL stored functions (PL/pgSQL) powering the REST API
- Caught critical mismatches between specifications and the database schema early, preventing backend rework at the testing stage and saving up to 2 weeks of development
- Introduced test data generators that made manual testing 3x faster

Tech stack: Vue 3, Vuetify, TypeScript, Leaflet, GeoServer, Node.js, PHP 8, PostgreSQL (PL/pgSQL), REST API, Docker, Docker Compose, Git, Linux, Redmine

Fullstack Developer

Vontamona (cruise travel agency) · Part-time, sole developer *Sep 2025 – Present*

- Designed the Next.js frontend architecture (feature-based structure, module isolation) and implemented cruise selection as a finite state machine (FSM), eliminating impossible states in user flows
- Replaced a hardcoded catalogue with PostgreSQL: schema with CHECK constraints, versioned migrations (node-pg-migrate, advisory lock) and parameterized SQL search filters, with every row validated by Zod
- Built an admin panel on Next.js Server Actions so the business owner manages cruises without a developer: content updates went from a full CI/CD release cycle (~6 min, developer required) to under a minute with no deploy
- Shipped the database to production with zero downtime: two Docker Compose projects on a shared network, Caddy reverse proxy with basic auth on admin routes, migrations applied on deploy
- Set up CI/CD and the PR process, integrated Telegram Bot API and SMTP notifications for incoming requests; reached a Lighthouse SEO score of 100 with a mobile-first approach

Tech stack: Next.js 16, React 19, TypeScript, Tailwind CSS v4, Zod, React Hook Form, PostgreSQL 17, node-pg-migrate, Docker Compose, Caddy, GitHub Actions, Vitest

Live: vontamona.ru | **GitHub:** github.com/vladislavprozorov/full-vontamona-frontend

Fullstack Developer

Sber (Sberbank) · Fixed-term project: educational platform for ITMO and SPbU students *Oct 2025 – Dec 2025*

- Designed the system architecture (React + Node.js + PostgreSQL), supporting 300+ concurrent users
- Implemented secure authentication (JWT + OAuth2) with role-based access control (RBAC)
- Designed the REST API and data model for courses, users and learning progress
- Introduced Redis caching, cutting database load by 60% and reducing latency to under 150 ms
- Set up production infrastructure: Docker, Nginx, SSL (Let's Encrypt), CI/CD, automated backups

Tech stack: React, TypeScript, Tailwind CSS, Node.js, Express, Prisma, PostgreSQL, Redis, Docker, CI/CD

Live: projectstory.ru | **GitHub:** github.com/vladislavprozorov/edu-project

PROJECTS

Crypto Dashboard — real-time crypto market & portfolio tracker

Client-side app with live prices, portfolio P&L, candlestick charts with indicators, watchlist and price alerts

- Streamed live prices for the whole market over a single Binance WebSocket connection with resilient reconnection (exponential backoff with jitter, heartbeat watchdog, offline pause)
- Batched dozens of ticks per second into one UI update per second via an external store and useSyncExternalStore, avoiding re-renders on every message
- Moved indicator calculations (SMA, EMA, Bollinger Bands, RSI) to a Web Worker using zero-copy transferable buffers, with a main-thread fallback; live candles update in place without reloading the series
- Built portfolio accounting (average-cost P&L, partial sales, value history) as pure functions, stored in IndexedDB behind a repository interface with optimistic updates, rollback and cross-tab sync via BroadcastChannel
- Designed a data source cascade (CoinGecko → cached snapshot → Binance) to stay functional under API rate limits
- Covered with 98 Vitest unit and component tests plus Cypress E2E scenarios; CI/CD to GitHub Pages via GitHub Actions

Stack: React 19, TypeScript (strict), Vite, TanStack Query, TanStack Table, lightweight-charts, Recharts, WebSocket, Web Workers, IndexedDB, Tailwind CSS, Vitest, Testing Library, Cypress, GitHub Actions

Live: vladislavprozorov.github.io/crypto_dashboard | **GitHub:** github.com/vladislavprozorov/crypto_dashboard

component-forge — CLI tool for enforcing React architecture

- Implemented static import analysis (custom AST-like parser, no external dependencies) that detects FSD/Modular layer boundary violations and gives actionable hints for 14 layer pairs
- Designed a Handlebars-based code generation pipeline with slice-aware scaffolding (feature → ui+model+api, entity → model+api, widget → ui+model)
- Set up CI/CD (GitHub Actions) with a Node 20/22 matrix and automatic npm publishing on release tags

Stack: Node.js, TypeScript, Commander.js, Handlebars, chokidar, ora, jiti, Vitest, GitHub Actions

npm: @xanahlight/component-forge | **GitHub:** github.com/vladislavprozorov/component-forge

Interactive 3D Construction Site Visualizer

Web app for interactive 3D visualization of building construction status, controlled through a React UI

- Built an interactive 3D building visualizer with React, TypeScript and Three.js, separating the UI layer from the rendering engine
- Implemented floor status visualization (Completed / In Progress / Issue) with dynamic state updates
- Connected a mock API with typed data models; added lazy loading with shimmer skeletons

Stack: TypeScript, React, Three.js, Raycaster, Vite, GitHub Actions

Live: vladislavprozorov.github.io/sber-three-react | **GitHub:** github.com/vladislavprozorov/sber-three-react

EDUCATION

ITMO University — Expected 2027

MSc in Software Engineering and Computer Systems — *Saint Petersburg, Russia*

Astrakhan State University — 2025

BSc in Informatics and Computer Engineering (Digital Technologies), with Honours — *Astrakhan, Russia*

Thesis: Interactive platform for learning technical English (React, Node.js, PostgreSQL, Socket.IO)

TECHNICAL SKILLS

Languages: TypeScript, JavaScript, PHP 8, SQL (PL/pgSQL)

Frontend: Vue 3, Vuetify, React, Next.js, Angular, Redux Toolkit, TanStack Query, Tailwind CSS, Leaflet, Three.js, WebSocket, Web Workers, IndexedDB

Backend: Node.js, Express, NestJS, PHP, REST API, JWT, OAuth2, RBAC

Databases: PostgreSQL (stored functions, PL/pgSQL), Redis, MongoDB, Prisma, Sequelize, node-pg-migrate

DevOps & Tools: Docker, Docker Compose, Nginx, Caddy, Linux, CI/CD, GitHub Actions, Git, GeoServer, Redmine

Testing: Vitest, Cypress

Architecture & Practices: Feature-Sliced Design, Modular Architecture, Git Flow, Code Review, Agile

Spoken languages: Russian (native), English (B2, Upper-Intermediate)