Nikita Lobanov

Full Stack Engineer
nikitaalobanovv@gmail.com | linkedin.com/in/nikitalobanov | nikitalobanov.com | github.com/nikitalobanov12

EDUCATION

British Columbia Institute of Technology Bachelors of Science in Computer Science, Expected April 2026
Concentration: Full-Stack Web Development *Vancouver, BC*
Relevant Courses: Data Structures & Algorithms, Web Database Technologies, Operating Systems, Object-Oriented Programming, JavaScript Frameworks and Server, Mathematics for Computing, Agile Software Development, Web Administration

EXPERIENCE

Seaspan Corp **May. 2024 -Sep. 2024**
Software Development Intern *Vancouver, BC*

- Designed and optimized **Express.js REST APIs** to facilitate data exchange between Oracle ERP Cloud and Excel; Processing **1,400+ files per month** & removing the need for manual data entry for accounts payable records.
- Developed data visualization dashboards in **React** for key vessel performance indicators such as speed-over-ground, specific fuel oil consumption, and engine load, enabling precise monitoring and optimization across Seaspan's fleet of **227 containerships**
- Served as a point of contact for employee support during the global **July 2024 CrowdStrike outage**, working closely with the IT Support team & guiding **80+ employees** through remediation steps and confirming full functionality of their machines.
- Monitored **Oracle database** performance and conducted routine health checks on Linux/Windows servers and REST APIs, ensuring **99.9% uptime** by identifying and addressing potential bottlenecks.

PROJECTS

DayFlow | React, React Router, Tauri, PostgreSQL, GitHub Actions, Node.js, Stripe, Google VertexAI **GitHub | Production Deployment**

- Built cross-platform task management app using React frontend & Tauri platform for native desktop compilation with a shared codebase between web and desktop versions. Grew from 0 to 300+ daily active users over 2 months.
- Integrated Google Gemini AI API to parse natural language task descriptions like "plan mom's birthday party next week" into structured tasks with deadlines, priorities, and subtasks. Simplifying the task creation process for users.
- Identified slow database queries using PostgreSQL EXPLAIN ANALYZE and added B-tree indexes on user_id, due_date, and status columns. Optimized JOIN queries between tasks and projects tables. Cut average query time from 800ms to 240ms.
- Built Stripe payment integration system with webhook handlers for payment processing and trial management. Set up GitHub Actions CI/CD pipeline that runs tests, builds artifacts, and deploys to production on commit or merge to main.

Circles | Next.js, Prisma PostgreSQL, Redis, Vercel, Tailwind CSS **GitHub | Production Deployment**

- Built a social platform for private community groups with photo sharing, activity feeds, and member permissions. Led team of 3 developers using 1-week sprints, daily standups, and Git feature branch workflow with code reviews done by me.
- Worked with 5 designers to create Figma mockups and convert them into pixel-perfect React components. Built responsive layouts that matched designs across desktop, tablet, and mobile breakpoints.
- Fixed N+1 query problems where loading user feeds was making 200+ individual database calls to fetch post authors, comments, and likes. Implemented Prisma eager loading with include statements and restructured queries to use joins. Reduced query count from 200+ to 3 queries per page load.

WriteShare | AWS Services, TRPC, Web Sockets, Docker, Redis, NextAuth.js **GitHub| Production Deployment**

- Built Google Docs-style collaborative markdown editor where 10+ users can edit simultaneously without conflicts. Integrated Liveblocks WebSocket connections with Yjs's CRDT algorithm to merge concurrent edits and maintain document consistency.
- Implemented Redis caching layer that stores frequently accessed documents in memory while PostgreSQL handles persistent storage. Used Redis pub/sub for real-time notifications and background workers to sync cache to database every 30 seconds. Reduced document load times from 2.3s to 150ms.
- Debugged infinite save loop where React useState updates triggered CRDT sync events, which triggered more useState updates. Fixed by moving document state management entirely into Liveblocks provider and removing React state dependencies for editor content.
- Containerized application with Docker multi-stage builds and deployed on AWS ECS Fargate with auto-scaling groups. Used RDS PostgreSQL for data persistence and ElastiCache Redis clusters for caching and session management.

SKILLS

Languages: JavaScript, TypeScript, SQL, HTML, CSS, Golang
Frontend: React, Next.js, Tailwind CSS, React Router
Backend: Node.js, Gin, Express.js, NestJS, REST APIs, gRPC, tRPC
Databases: PostgreSQL, Redis, Oracle Database, Prisma ORM
Cloud & DevOps: AWS (ECS, ECR, RDS, S3, VPC, ElastiCache), Docker, GitHub Actions, Vercel
Tools & Methodologies: Git, GitHub, Jira, Agile/Scrum, CI/CD, Database Optimization, Performance Monitoring