

Nikita Lobanov

Canadian Citizen | nikitaalobanov@gmail.com | [linkedin.com/in/nikitalobanov](https://www.linkedin.com/in/nikitalobanov) | nikitalobanov.com | github.com/nikitalobanov12

EDUCATION

British Columbia Institute of Technology | Vancouver, BC

Expected Graduation: **May 2026**

Bachelor of Science in Computer Science, Full-Stack Web Development Specialization

Relevant Courses: Data Structures & Algorithms, Web Database Technologies, Operating Systems, Object-Oriented Programming, JavaScript Frameworks and Server, Mathematics for Computing, Web Administration, Unix Tools and Scripting, Agile Development.

EXPERIENCE

Seaspan Corp: Full-Stack Development Intern | Vancouver, BC

May 2024 - Aug 2024

- Developed Spring Boot REST APIs to facilitate data exchange between Oracle ERP Cloud and Excel, processing 1,400+ files per month & **eliminating manual data entry for accounts payable, saving 160+ hours monthly.**
- Delivered 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.
- Coordinated employee support during the global July 2024 CrowdStrike outage, working closely with the IT Support team & guiding 80+ employees through remediation steps. **Achieved 100% system recovery and full functionality across all machines.**
- Monitored Oracle database performance and conducted routine health checks on Linux/Windows servers and Spring Boot & Express.js REST APIs, making sure to address potential bottlenecks related to unoptimized database queries & resource usage.

PROJECTS

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

- Architected cross-platform task management app using **React** frontend & **Tauri** platform for native desktop compilation with a shared codebase between web and desktop versions. **Scaled from 0 to 300+ daily active users in 8 weeks.**
- Engineered **Google Gemini AI API** integration to parse natural language task descriptions like "plan mom's birthday party next week" into structured tasks with deadlines, priorities, and subtasks.
- Optimized slow database queries using **PostgreSQL EXPLAIN ANALYZE** and **B-tree indexes** on frequently read columns. Restructured JOIN queries between tasks and projects tables, **improving average query performance from 800ms to 240ms.**
- Orchestrated **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

- Spearheaded development of a Full-stack 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. **Delivered 100% of sprint commitments across 8 iterations.**
- Collaborated with 5 designers to create Figma mockups and convert them into pixel-perfect React components. **Achieved responsive layouts matching designs across desktop, tablet, and mobile breakpoints with zero layout inconsistencies.**
- Eliminated 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 by 98.5% from 200+ to 3 queries per page load.**

WriteShare: AWS Services, tRPC, WebSockets, Docker, Redis, NextAuth.js

GitHub | Production Deployment

- Developed Google Docs-style collaborative markdown editor where **25+** users can edit simultaneously without conflicts. Integrated Liveblocks **WebSocket connections** with Yjs's **CRDT algorithm** to automatically merge edits and maintain consistency.
- Deployed **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. **Accelerated document load times by 1,433% from 2.3s to 150ms.**
- Resolved 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 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, Golang, C++, Java, Python, Lua, Bash

Frontend: HTML, CSS, React, Next.js, Tailwind CSS, React Router, Shadcn, Sass, EJS, Angular

Backend: Node.js, Gin, Express.js, NestJS, REST APIs, gRPC, tRPC, GraphQL, Chi, Spring Boot, Flask, Microservices

Databases: PostgreSQL, Redis, Oracle Database, Prisma ORM, MongoDB, Drizzle

Cloud & DevOps: AWS (ECS, ECR, RDS, S3, VPC, ElastiCache), Docker, GitHub Actions, Vercel, Linux, Railway, Cloudflare

Tools & Methodologies: Git, GitHub, Jira, Agile/Scrum, CI/CD, Database Optimization, Performance Monitoring, CDN, SEO