

Sultan Alzoghaibi

North Vancouver, BC (open to relocation) • Fluent in French • ssultan007@icloud.com • ca.linkedin.com/in/sultan-alzoghaibi/ • github.com/SultanAlzoghaibi • sultan-alzoghaibi.com

TECHNICAL SKILLS

- **Languages:** Java, Go, JavaScript, TypeScript, Python, SQL, HTML/CSS, YAML, XML, C++
- **Frameworks:** Spring Boot, React, Next.js, Tailwind CSS, JUnit, Node.js
- **APIs & Integrations:** RESTful APIs, gRPC, Web Sockets, GraphQL, Protocol Buffers
- **Cloud & Deployment:** Git, Terraform, GCP, GKE, Cloud SQL, AWS (ECS Fargate, EKS, EC2, RDS, CloudWatch), Elasticsearch
- **Infra & DevOps:** Docker, Kubernetes, PostgreSQL, MySQL, Redis, Linux, TCP/IP, NGINX, Bash, ngrok

EDUCATION

Bachelor of Science, Computer Science

Sep. 2022 – Present

University of Calgary, Alberta

- Science **Co-op Program** | Dean's List | **GPA: 3.96** (2024-25 Academic Year)
- Relevant Courses: InfoSec & Privacy; In progress (Computer Networks, Database Management Systems, Scalable Data Analytics)
- Award: 2nd place out of 43 Teams | Div. 2 | Alberta Collegiate Programming Competition 2024

PROJECTS

Leetcode challenge Discord Bot + Chrome extension | Go, MySQL, GKE, Terraform, JavaScript

Jul. 2025 – Present

- **With 13 active users**, built a Chrome extension and backend server that upon a successful LeetCode submissions, prompts users for notes and confidence scores, then logs data to a shared Google Sheet and MySQL database on Cloud SQL: enabling a discord bot to post daily tracking and leaderboards.
- Deployed on GKE using Terraform with YAML-defined test and prod environments, static **NGINX Ingress** IP address and self-healing pods. Performed in **production blue-green and rolling deployments**, ensuring near **24/7 uptime for real users**.
- Implemented cybersecurity protocols using **Terraform-defined private subnets**, strict origin validation and eliminating spoofed curl submissions by performing real-time **GraphQL lookups to LeetCode.com for verification**.
- **Links:** [Chrome extension](#) | [GitHub](#) | [FigJam DB schema's & system](#)

Distributed Slack Survey Platform | Spring Boot, AWS EKS, gRPC, PostgreSQL, Redis, C++, Terraform

May. 2025 – Jun. 2025

- Architected and deployed a Slack-integrated health check platform on AWS EKS, leveraging Spring Boot, Redis, and PostgreSQL to monitor team morale across multiple teams.
- Simulated Slack POST requests in a high-load test, averaging **2,430 RPS** across **109,480 requests** in 45s, with **95th percentile latency of 417ms** and **1.31% error rate**. Traffic was distributed across 30+ EKS and, auto-scaled via **HPA**.
- Implemented a **5-schema** PostgreSQL DB (hosted on RDS) and a Redis cache (hosted on ElastiCache), cutting repeated DB lookups and reducing latency from ~25ms to **~2ms (~92% faster)**.
- Cached Team scores are sent via **gRPC** to a C++ microservice with **~0.098ms avg** (*localhost, 100k request warmup)
- **Links:** [Draft System Design FigJam](#) | [GitHub](#) | [C++ microservice GitHub](#)

Horizontally Scaling Matchmaking Distributed System | Vanilla Java, Docker, AWS ECS Fargate

Feb. 2025 – Mar. 2025

- Deployed a matchmaking Server that creates and pairs players to a dedicated Docker GameServer container on **AWS ECS**, resulting in **0 coupling** between matchmaking and each gameplay session.
- Implemented **dynamic warm pooling** using a `Queue<IpAddress>` per game mode, reducing GameServer pairing time by **99.1%** (~46s → 0.4s). The system **automatically scales up or down pool size** based on the server load.
- Enabled concurrent matchmaking via **event-driven threads**, achieving **sub-950ms latency** under load.
- In GameServer and Player instances, added **WebSocket-based multi-threaded** chat feature on a separate port with real-time chat running in parallel with turn-based sequencing, while maintaining latency below **280ms**.
- **Links:** [2min demo video](#) | [GitHub](#) | [Portfolio site](#)

CLUB EXPERIENCE

Full-Stack Software Developer

Jan. 2025 – Apr. 2025

TechStart UCalgary (Club), Alberta | [Website](#) | [GitHub](#)

- Migrated from React with CSS to Next.js with Tailwind, leveraging SSR to boost SEO (**~25%**) and reducing load times (**~30%**).
- Built a **Stripe API** integration using **GET/POST RESTful API endpoints**, reducing average transaction time by **2 minutes** compared to Google Forms and enabling new item additions with **0 coding** via Stripe's dashboard.
- Collaborated in an Agile team of 5, applying Git best practices and performing code reviews to ensure 0 bugs were pushed to main, maintaining stable releases.
- Also Served for 8 months as VP UX/UI Design at TechStart; after being mentored by a UX/UI graduate, took initiative to create and lead workshops on user research, UI heuristics, and **Agile UX/UI** practices, receiving **83% positive feedback**.