

# Sultan Alzoghaibi

(825) 964-0164 • North Vancouver, BC • [ssultan007@icloud.com](mailto:ssultan007@icloud.com) • [ca.linkedin.com/in/sultan-alzoghaibi/](https://ca.linkedin.com/in/sultan-alzoghaibi/) • [github.com/SultanAlzoghaibi](https://github.com/SultanAlzoghaibi) • [sultan-alzoghaibi.com](https://sultan-alzoghaibi.com)

## TECHNICAL SKILLS

- **Languages:** Java, TypeScript, JavaScript, Python, C++, SQL, HTML/CSS, JSON, XML, C
- **Frameworks:** Spring Boot, React, Next.js, Tailwind CSS, JUnit, Node.js, Motion
- **APIs & Integrations:** RESTful APIs, gRPC, WebSocket, Slack API, Stripe API, OpenAI API, Protocol Buffers
- **Libraries & Tools:** Git, Playwright, Figma, Bash, Visual Studio Code, IntelliJ IDEA, Clion, PyCharm, pgAdmin
- **Cloud & Infra:** Docker, Kubernetes, AWS ECS Fargate, EC2, PostgreSQL, TCP/IP, Redis, Linux, CloudWatch, 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: Programming in Python, Object-Oriented Programming in Java, Data Structures and Algorithms, Intro to Software Engineering, Info Security & Privacy
- Award: 2nd place out of 43 Teams | Div. 2 | Alberta Collegiate Programming Competition 2024

## PROJECTS

### Horizontally Scaling Matchmaking Distributed System | Java, JavaFX, Docker, AWS ECS Fargate, CloudWatch, AWS SDK

- Deployed a SearchServer 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.
- Connected players using a **multi-threaded** HashMap<gameMode, List<Player>> and event-driven threads, enabling **concurrent matchmaking** with **sub-950ms latency** under load.
- In GameServer and Player instances, **multi-threaded** a 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](#)

### Distributed Slack Survey Platform (In Progress) | Java, Spring Boot, gRPC, PostgreSQL, Redis, C++, Kubernetes, AWS EKS

- Building a scalable Slack survey platform for weekly team health tracking and detecting dysfunctional teams: Slack API → load balancer → Spring Boot server → gRPC C++ microservice → PostgreSQL and Redis (caching). Using **Kubernetes** and ngrok for local testing, with scalable deployment planned on AWS EKS. **Expected completion: June 2025**
- Integrated Slack slash commands via **REST API** into a **Spring Boot server**, routing commands to threaded handlers and interacting with a **relational PostgreSQL database** of **5 linked schemas** (monitored via pgAdmin).
- Implemented **Redis caching** to eliminate redundant PostgreSQL lookups, reducing latency from ~25ms to **~2ms (~92% faster)** (monitored via Redis Insight).
- Cached Team scores are then sent via **gRPC** to a highly optimized **C++ microservice** that computes a variance-adjusted average and returns the result in **~0.098ms** (\*localhost, 100k request warmup).
- Set up **4 Kubernetes pods** via **Minikube** using custom YAML configs to run PostgreSQL, Redis, Spring Boot, and a C++ gRPC microservice for local testing and debugging.
- **Links:** [3min dev-vlog](#) | [System Design FigJam](#) | [GitHub](#) | [C++ microservice GitHub](#)

### Pacman with Smart Pathing Ghost AI | Java, JavaFx

- Led a team in developing Pac-Man clone in **2 weeks**, writing **~85 %** of the code and building an Object-Oriented architecture with **O(1)** ghost pathfinding and a synchronized GUI-grid backend, achieving a grade of **102.3%**.
- **Links:** [demo video](#) | [FigJam Planning](#) | [GitLab](#) | [Portfolio site](#)

## EXPERIENCE

### Full-Stack Software Developer

Jan. 2025 – Apr. 2025

TechStart UCalgary (Club), Calgary, 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 clear communication to ensure 0 bugs were pushed to main, maintaining stable releases.
- 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**.