

# YAHYA ABOUELMAGD

[Link to my website with all my contact info, links and other projects!](#)

## Skills

---

**Languages :** Python, Java, C/C++, Kotlin, SwiftUI, PHP, HTML/CSS, NodeJs, ReactJs, SQL, Ladder-Logic, ARM64 Assembly, Verilog

**Development Skills:** Data Structures and Algorithms, Web Development, Object-Oriented Programming, CI/CD, Embedded System Development, Remote System Development, Networking, Digital Design, Debugging, and Version Control

## Education

---

**University of British Columbia**

**September 2021 – May 2026**

*Bachelor of Applied Science in Computer Engineering, Deans list 2022*

*Vancouver, BC*

## Work Experience

---

**Broadcom Ltd.**

**Jan 2025 – Aug 2025**

*Software Engineer Co-op*

*AWS, Spring, Jenkins, Java, Microservices, Docker, Gradle, K8s*

- Designed a static analysis workflow to **refactor 600+** deprecated Java APIs across the Layer7 Gateway; integrated into **Jenkins** CI/CD to block regressions and ensure compliance with modern Java security standards.
- Hardened service reliability by redesigning error-handling flows in **Spring** microservices, producing clearer diagnostics and preventing silent failures in production deployments.
- Integrated **Black Duck** scans into **Jenkins** pipelines to automate dependency checks, patch CVEs, and generate compliance reports for security teams.
- Replaced legacy PowerMockito tests with secure Mockito-inline practices, eliminating reliance on unsafe bytecode manipulation and increasing test stability.

**DELL Technologies**

**July 2024 – August 2024**

*Intern*

- Participated in designing and implementing a new **TCP** communication protocol suited for DELL's PowerEdge Servers.
- Specifically worked on developing an acknowledgment (ACK) mechanism and timeout handling.
- Led the testing and shadowed the rollout of the protocol, boasting a **35%** decrease in packet loss and network traffic.

## Capstone Project

---

**Agentic, AI Data Scientist** | *Machine-Learning, Numpy, RESTful API, NodeJS, Docker, AWS, MongoDB*

- Automated **data validation and preprocessing workflows**, reducing manual preparation effort by **40%** and improving speed and accuracy of retail analytics.
- Architected a scalable, cloud-deployed pipeline of four agents — **Data Cleaning, Preprocessing, Feature Engineering, and Hypothesis Testing** — using a Node.js/TypeScript backend on **AWS**.
- Developed ML models for **classification, regression, and forecasting** and integrated **LLM-powered sub-agents** to generate plain-language explanations, enhancing interpretability and usability of insights.

## Projects

---

**Dormdash** | *Kotlin, RESTful API, NodeJS, Docker, AWS, MongoDB*

- An Uber-like app that lets students schedule requests for movers to pick up, store, and return their belongings running on a **Dockerized NodeJS** server.
- Integrated Google OAuth, Maps, and Stripe and **REST** endpoints, for seamless auth, scheduling, and payments.
- Designed **NodeJs** backend and Android MVVM frontend to support up to **400** concurrent users and **200** daily jobs, gaining core experience in building scalable, microservice-oriented applications.
- Followed an **event driven architecture**, allowing students to receive real-time updates on their orders.

**Video Game Library RDBMS (CPSC 304), UBC** | *SQL, React, Oracle, Github*

- Designed a **Relational Database Management** system, managing a video game library.
- Enhanced data model complexity by defining **one-to-one and one-to-many relationships**, such as distinguishing recreational and professional players, and linking companies to game series.
- Built a **ReactJS** front-end, enabling users to remove or update entries by sending **SQL-Injected** requests.
- Implemented **dynamic** database querying features for real-time data updates and tailored user information retrieval.

**Text Messaging Application** | *Full-Stack Development, RESTful API, NodeJS, MongoDB, WebSocket, JavaScript, Ollama*

- Designed a web application enabling users to create profiles and chat rooms.
- Implemented live chat functionalities and room updates via **AJAX** requests and **WebSocket** communication.
- Followed the **MVC** architecture for dynamic client-side rendering of chat and profile pages.
- Set up a **MongoDB** database with collections to store and manage chat messages and user information.
- Developed a Data Access Object (DAO) layer to read/write conversations and user information. Including robust user **authentication** and authorization mechanisms
- Built and deployed a **NodeJS** server, with **RESTAPI** endpoints to allow for profile creation, messaging, etc.
- Hosted a language AI model locally and provided users with an advanced autocorrect feature.