

YAHYA ABOUELMAGD

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

Skills

Languages : Python, Java, C/C++, SwiftUI, PHP, HTML/CSS, NodeJs, ReactJs, SQL, Ladder-Logic, ARM64 Assembly, Verilog
Development Skills: Data Structures and Algorithms, Web Development, Object-Oriented Programming, 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

Vancouver, BC

Work Experience

Terraforma Systems Inc.

September 2024 – Present (Expected: 4 months)

Intern

Pytorch, Java, PHP, AWS, JavaScript, SQL, Networking

- Built and tested a webhook-invoked Jenkins pipeline. Automating 75% of the CI/CD process.
- Exposed REST-ful endpoints to receive JSON-formatted sensor data and images from machines.
- Developed templates and visualized received data on them through jquery.
- Speeding up database communication for users' web tools by implementing caching strategies, reducing query load.
- Utilizing the Ewon Java Toolkit to develop the 'Smart Lifter', enabling remote tag management and event log viewing.
- Deploying and integrating a machine-learning model running on flask trained to classify materials.

DELL Technologies

July 2024 – August 2024 (2-months)

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.

Projects

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.
- Implemented an MVC pattern 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
- Protected the server from Cross-Site Scripting (XSS) attacks by sanitizing messages.
- Implemented a NodeJS server, with RESTful API server-side endpoints to allow for profile creation, messaging, etc.
- Hosted a language AI model locally and provided users with an advanced autocorrect feature.

Video Game Library DBMS (CPSC 304), UBC | *OracleSQL, PHP, Oracle, Github*

- Designed a relational database to describe a video game library with tables like Player, Game Series, Tournament.
- 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.
- Designed and implemented a PHP front-end, enabling users to remove or update entries by invoking OracleSQL calls following their interaction.
- Implemented dynamic database querying features for real-time data updates and tailored user information retrieval.

Large-scale Twitter Proxy Service | *Back-End Development, Apache Tomcat, AWS, Java, IntelliJ, JUnit, Git/Github*

- Designed a thread-safe service able to proxy up-to a 1000 requests to Twitter concurrently, using their API SDK.
- Setup an Apache Tomcat server to expose API endpoints for users and manage threads.
- Implemented the twitter action proxy layer responsible for forwarding requests to Twitter.
- Load-tested the application on a remote EC2 instance on AWS to ensure it can handle the expected load.
- Achieved 95% code coverage using a suite of unit tests, evaluating server capabilities at different capacities.

Other Programming Experience

Networking

- In-depth knowledge of each of the OSI model, helping me build and test - on WireShark - a TCP-like reliable data transfer instance, with less than 10% packet loss and congestion control
- Designed, implemented, and tested scalable distance-vector and link-state routing protocols.

Embedded Systems(C/C++)Concurrency, Synchronization, Virtual memory, System calls, and File systems.

- Built a small bash shell, able to run Unix commands, and control processes running in the foreground and background.
- Enhanced Harvard's OS161 with synchronized multi-threading and multiprocessing capabilities, filesystem and process system calls, and memory virtualization. Optimizing physical memory allocation and making it able to withstand 2000 concurrent processes.