# Noel Beraki

noelberaki03@gmail.com / 206-582-8707 / Seattle, WA, US / LinkedIn / GitHub

## Education

**University of Washington – Bothell,** GPA: 3.87 — Bothell, WA
*Bachelor of Science in Computer Science and Software Engineering* — *Expected June 2025*

**Relevant Courses:** Technical Writing; Fundamentals of Web Development; Software Engineering; Data Structures And Algorithms I, II; Database Systems; Design And Analysis of Algorithms

## Technical Skills

| | |
|---|---|
| **Programming Languages**: | C++, Java, SQL (Postgres), HTML, CSS, (Limited) JavaScript |
| **Frameworks**: | Spring Boot |
| **Developer Tools**: | Git, GitHub, Bitbucket, Visual Studio Code, IntelliJ, MacOS, Linux, Docker |

## Experience

**Software Development Engineer Intern** — June 2024 – September 2024
Amazon — Sunnyvale, CA

- On the AWS hyperplane team, working on region build automation and modeling the region build orchestration as a directed acyclic graph. Modeled DAG with yaml files providing team members with human-readable version of graph. Allowed team members to only need to code once and deploy to multiple regions.
- Provided automatic validation checks and publishing of a region build through a pipeline for CI/CD deployment.

## Projects

**One Piece RESTful API | GitHub** — Spring Boot, Java, SQL (Postgres)

- Developed a RESTful API utilizing Spring Boot to manage character data from the tv show One Piece.
- Integrated a PostgreSQL database through the JPA using Hibernate as the ORM framework.
- Implemented a set of endpoints for listing all characters; retrieving characters by ID, searching by name, and filtering by affiliation, which are columns of the connected database table.
- Implemented CRUD operations of creating, reading, updating, and deleting character records, using appropriate request mappings and status codes.

**Movies Inventory Tracking System | Bitbucket** — C++

- Developed an automated inventory tracking system for a movie rental store using the object-oriented design principle polymorphism and the factory design pattern to efficiently manage three movie types (Comedy, Drama, Classic), with more movie types able to be added without altering any pre-existing files.
- Implemented robust error handling mechanisms to gracefully manage invalid commands and notify users of any issues, ensuring smooth processing of borrowing, returning, and inventory management actions.
- Incorporated bonus features such as runit scripts to showcase the fully polymorphic design and a custom hashing function.

**Class Projects: Graph ADT, Enrollment System, SkipList, Maze Solver | Can only be shown in private** — C++

- Implemented a Graph data structure using an adjacency list. Algorithm's implemented include traversals such as DFS, BFS, Dijkstra's shortest path, Prims minimum spanning tree, and Kruskal's minimum spanning tree.
- An Enrollment System designed to manage student enrollment in courses across multiple universities. Worked with multiple classes sharing pointers, different types of containers, and sorting based on given criteria.
- Created a SkipList data structure that utilizes the advantages of a fixed sized array and a linked list. Provides search, insert, and add operations to be completed in O(log n) time.
- Implemented a maze solver using recursive backtracking. Maze represented as a string with 'X' for walls, '+' for visited indexes not a part of solved path, and '*' for visited indexes part of the path. Printed the attempted maze with a string that represented a solved path for the maze or an indication that the maze is unsolvable.