Nicholas Bowden

bowdenn@oregonstate.edu 360-460-4260
https://www.linkedin.com/in/nicholas-bowden/ www.nicholasbowden.com

Summary

Insightful computer science student driven by an endless curiosity to see what's around the next corner. Seeking a web development internship. Well-versed in the ecosystem while quick to adopt and leverage new technologies that arise. Prepared with the skills to overcome problems anywhere in stack, having built multiple full stack applications using React and Node. Comfortable using both SQL and NoSQL databases.

Education

Oregon State University - **BS in Computer Science** 2020 - Ongoing
GPA: 3.9, Post-baccalaureate program

Eastern Washington University - **BA in Music** 2010-2015
GPA: 3.6, Dean's list for 6 quarters.

Skills

Languages: Javascript, Python, SQL

Technical Skills: software engineering, frontend development, backend development, databases, data structures, algorithms, web scraping, git/github

Soft Skills: time management, customer service, leadership, written and verbal communication, collaboration, introspection, empathy

Relevant Coursework & Project

CS 161/162 – Introduction to Computer Science
CS 225 – Discrete structures in Computer Science
CS 261 – Data Structures
CS 271 – Computer Architecture and Assembly Language
CS 290 – Web Development
CS 340 – Introduction to Databases

Blog List – full-stack single page blog list application designed with React, Node.js, and MongoDB hosted on Heroku. Logged in users can post blog links on a public wall for others to see and like.

- Backend built with Node.js and Express while using Mongoose to communicate with MongoDB.
- Frontend built with React, Styled Components, and Axios.
- End-to-End testing done using Cypress with help from Jest, and Supertest for the unit tests.

CrowdFlow – SQL database project with web-based UI utilizing a RESTful architecture. Simulates the CrowdFlow Ticketing System admin interface for managing the database.

- Full schema, ERD, and project proposal extensively reviewed by peers throughout all stages.
- Multiple pages all with CRUD functionality dynamically updated from the database.
- Displays variety of database relationships with use of foreign keys and constraints

DFS/BFS/Dijkstra's – Implementations of the depth first search, breadth first search, and Dijkstra's search algorithms for graphs.

- Written in Python and follows best practice guidelines.
- Functionality implemented for both directed and undirected graphs.
- Employs adjacency lists and matrices to find valid paths, connected nodes, cycles, etc.