Andrey Karmanov

Honours Computer Science and Business Administration 1B

andrey@karmanov.ca | 289-772-8378 | linkedin.com/in/andrey-karmanov | andreykarmanov.github.io

Education

2021 – 2026 Waterloo, Canada
Honours Co-op Computer Science and Business Administration,
University of Waterloo, Wilfrid Laurier University

Skills

Languages
Python - JavaScript - Java

Tools and Libraries
Git - Google CP - MongoDB - React.js - Node.js - Flask

Work Experience

09/2021 – present
Lifeguard, *University of Waterloo*
Selected from pool of candidates to ensure safety of swimmers of all ages and abilities.

08/2020 – 09/2020
Full Stack Developer Co-op, *Mei Naggapan*

- Worked with team to plan website - Segarage.org ↗
- Reworked site to utilize bootstrap and templates
- Reworked search results and filters with Elasticsearch
- Edited MySql database to optimize for searching and user experience

Projects

Maze Pathfinding Visualization, *Javascript, React, Algorithms*

- Implemented depth first search, **A Star**, and **Dijkstra's** Search algorithms to solve randomly generated mazes of various sizes.
- Algorithms were implemented without external libraries.
- Visualized the algorithms with an **interactive JavaScript React app**, which allows users to change the size, shape, and search algorithm of the maze.

Self-Teaching through Reading and Practice, *Data Structures, Algorithms*
Over quarantine I began, and finished all relevant practice problems and chapters of textbook
Data Structures and Algorithms in Python by Roberto Tamassia and Michael T. Goodrich.
I have applied this knowledge to create other projects, and sink deeper into the field of computer science.

Online News Reliability Estimator, *Google CP, Flask, HTML, JavaScript, TensorFlow*
Worked with a University of Waterloo Professor to create an extension that classifies news articles as reliable or misleading.

- Created a machine learning model from online datasets.
- Hosted the model on personal machine, and then Google CP.
- Created browser extension that anonymously submitted sites to model to inform about reliability.

Sorting Visualizer, *JavaScript, React, Data Structures, Algorithms*

- Implemented **heap** sort, **quick** sort, **merge** sort, etc.
- Algorithms were implemented without use of external libraries, with my **own heap class** for heapsort.
- Visualized the algorithms with an **interactive JavaScript React app,** viewable on personal site.