ALEXANDER KOZIK

Software Engineer

linkedin.com/in/alex-kozik • github.com/LambdaAK • alexkozik.com • alex.kozik@yahoo.com • 215-264-2104 • Ithaca, NY

Education

Cornell University — Ithaca, NY

B.A. Computer Science, B.A. Mathematics | GPA: **3.96** / 4.0 — 08/2022 - Present

Experience

Computer Science Course Management System X (CMSX) @ Cornell University — Ithaca, NY

Full Stack Software Engineer — 08/2023 - Present

Technologies used: **Java, JSP, React.js, TypeScript, MySQL**

- CMSX is a course management platform in use by over **8,000 students** across **100 different courses**. Handling a vast and intricate codebase with **100,000 lines of code** and over **20 years of history**. Significantly enhanced professors' user experience, adding the ability to grant deadline extensions using a CSV. Currently working on improving the **responsiveness** of the mobile UI written in React.

Cornell Bowers CIS — Ithaca, NY

Teaching Assistant for CS 3110 - Data Structures and Functional Programming — 08/2023 - Present

- Facilitating **office hours** to help students debug their code (OCaml) and **refine course content**. **Grading** projects and exams. **Mentoring** two groups of students as they complete a cumulative final project for the course. **Answering questions** on Ed Discussion about projects and course content.

Staples — Warrington, PA

Retail Sales Associate — 07/2021 - 09/2021

- Operated as a cashier and supplies stocker. Demonstrated the ability to work quickly and efficiently.

Projects

LambdaScript - Custom Programming Language

https://github.com/LambdaAK/LambdaScript

A functional programming language inspired by Haskell and OCaml.

Technologies used: **OCaml, LaTeX**

- Basic and compound **data types**: int, float, bool, str, unit, list, vector, et cetera. **Functional constructs**: conditionals, lambdas, let expressions, currying, pattern matching, custom infix operators, list comprehension. **Type inference algorithm** uses a **type equation generator** and a **unification algorithm** to infer types. A **REPL** allowing a user to type expressions and receive their value and type. **Rigorous test suite** utilizing **functors** for code reuse. 20+ page long document detailing the **syntax** and **semantics** of the programming language.

AlgoSandbox - Algorithm and Data Structure Visualizer

https://github.com/LambdaAK/AlgoSandbox

A powerful tool designed to help people grasp complex algorithms and data structures through visual representation

Technologies used: **TypeScript, React.js, SCSS, Vite**

- Features many popular algorithms and data structures: merge sort, insertion sort, stack, queue, ... Pages detailing the **time complexity**, **space complexity**, and **implementations** of the algorithms. **Animated sandboxes** utilize visual effects to demonstrate how algorithms unfold in real-time. Elegantly-designed home page features a **search engine** for easy navigation.

HabitStack - Web Application

https://github.com/LambdaAK/HabitStack

A sleek and intuitive web application designed to support individuals in building healthy habits and breaking bad ones.

Technologies used: **React.js, TypeScript, JavaScript, SCSS, Express.js, Firebase, Vite**

- **Interactive calendar** for tracking daily tasks coupled with **dashboard widgets** that display user information. Create **personalized plans** for maintaining healthy habits and breaking bad ones. **Chat functionality** between users. **Secure authentication** using Firebase and data transfer using an Express.js **backend.** A page with important **habit-changing knowledge** I've discovered from reading personal development books.

CritterWorld - Artificial Life Simulator

A simulator for critters that fight to survive and reproduce. My final project for CS 2112 at Cornell, completed in a group of 3.

Technologies used: **Java, JavaFX, SceneBuilder, Gradle, JUnit**

- **Parser** and **Interpreter** for a programming language that controls the critters. **Graphical user interface** that shows how the critters move around the map and interact. **Fault injector** that converts valid parse trees into alternative, yet still valid structures. Created a **test suite** using JUnit to ensure the correctness of the application.

Skills

Frontend: React.js • JavaScript • TypeScript • HTML/CSS • SASS • TailwindCSS • JSP

Backend: Express.js • Flask • Firebase • MySQL

Languages: Python • Java • JavaScript • TypeScript • C/C++ • OCaml • Haskell • RISC-V Assembly

Other: SymPy • PyTorch • LaTeX • JavaFX • Russian

Applicable Courses

Analysis of Algorithms, Honors Object Oriented Programming and Data Structures, Systems Organization and Programming, Data Structures and Functional Programming, Discrete Structures, Linear Algebra, Multivariable Calculus, Introductory Macroeconomics