Alexander Kozik

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

Education

Cornell University — Ithaca NY
B.A. Computer Science, B.A. Mathematics | Grade: 4.024/4 — *Aug 2022 – Present*

Experience

PL/SWE Research Intern (REU) — May 2024 - Present
Carnegie Mellon University - Software and Societal Systems Department — *Pittsburgh, PA*

- Working with Dr. Jonathan Aldrich on improving SASyLF, a proof assistant specialized for programming languages.
- Used Java to implement a polymorphic module system (functors), improving modularity and enabling code reuse.
- Designed and implemented deep cloning and substitution algorithms for an AST hierarchy with over 100 node types.

Full Stack Software Engineer — Aug 2023 – Present
CMSX - A course management platform used by over 8,000 students each semester at Cornell University — *Ithaca, NY*

- Built API endpoints and serializer with Java and TS for fetching data about students and courses.
- Migrated the FE from global state (Redux) to component-level state (lazily fetching students' data using APIs).
- Implemented a parser for CSVs and UI (JSP) enabling professors to grant extensions via file upload.

TA - Analysis of Algorithms, Data Structures and Functional Programming — Aug 2023 – Present
Cornell Bowers CIS — *Ithaca, NY*

- Leading recitations with 30 students at a time, helping refine course material and debug code (OCaml).
- Mentoring 2 groups of 3-5 students as they complete a cumulative final project for the course.
- Grading projects and exams and answering 100+ questions on EdStem about projects and course content.

Projects

LambdaScript - Custom Functional Programming Language | *TypeScript, OCaml, Jest, OUnit, LaTeX*

- Built-in data types: Int, Float, Bool, String, Tuple, List, etc.
- Pattern matching, lambdas, currying, conditionals, let expressions, recursive definitions, etc.
- Elegant type system with parametric polymorphism, custom ADTs, and higher-kinded types.
- Linear-time type-inference algorithm that generates the type of any expression and the kind of any type.
- Test suite utilizing functors for code reuse with 13,000+ unit tests to verify correctness.

AlgoSandbox - Algorithm Visualizer | *React.js, TypeScript, SASS, Vite*

- Features 10+ popular algorithms and data structures: merge sort, insertion sort, stack, queue, etc.
- Interactive pages detailing the time complexity, space complexity, and implementations of the algorithms.
- Animated sandboxes utilize animations to demonstrate how algorithms unfold in real time.
- Elegantly designed home page features a tag-based search engine for easy navigation.

HabitStack - Habit Tracking Application | *React.js, TypeScript, JavaScript. SASS, Express.js, Firebase, Vite*

- Interactive calendar for tracking daily tasks coupled with dashboard widgets that display user information.
- Users can create personalized plans for maintaining healthy habits and breaking bad ones.
- Chat functionality between users and secure authentication using Firebase and Express.js.
- A page with essential habit-changing knowledge I've discovered from reading personal development books.

CritterWorld - Evolving Artificial Life Simulator | *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 creates 6 types of modifications in critter programs, which models genetic mutations.
- Test suite using JUnit to ensure the correctness of the application.

Technical Skills

Languages: Python, Java, JavaScript, TypeScript, C/C++, OCaml, Haskell, RISC-V Assembly
Frontend: React.js, JavaScript, TypeScript, HTML/CSS, SASS, TailwindCSS, JSP, Electron.js, Next.js, Redux
Backend: Express.js, Flask, Firebase, MySQL, Node.js
Other: PyTorch, SymPy, NumPy, LaTeX, JavaFX, Data Structures and Algorithms, Software Testing/TDD, REST API, Git, Linux, JavaCC

Applicable Courses

Analysis of Algorithms, Honors Object Oriented Programming and Data Structures, Systems Organization and Programming, Data Structures and Functional Programming, Discrete Structures, Programming Languages and Logics, Machine Learning, Probability Theory, Independent Reading and Research, Linear Algebra, Multivariable Calculus, Introductory Macroeconomics