Alexander Kozik

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

Education

Cornell University Ithaca NY

B.A. Computer Science, B.A. Mathematics *Aug 2022 – May 2026*

- **Relevant Coursework:** Analysis of Algorithms, Machine Learning, OOP and Data Structures (Honors), Systems Organization, Systems Programming, Functional Programming, Programming Languages and Logics, Discrete Structures, Probability Theory, Advanced Linear Algebra, Multivariable Calculus, Real Analysis, Macroeconomics
- **GPA: 4.024/4.0** Dean's List, Pursuing CS Honors, ML, PL, & SWE Specialization

Experience

Full Stack Software Engineer Aug 2023 – Present

Cornell University CMSX *Ithaca, NY*

- Maintain Cornell's official CS course management system, used by **10,000+ students**, ensuring efficiency and reliability.
- Lead a team of developers in rewriting our **JSP** website in **React**, modernizing the platform and enhancing responsiveness.
- Migrated CMSX from **Redux** to a component-level state system, creating a scalable foundation for future development.
- Designed 10+ **APIs** and serializable data structures, allowing secure data transfer between the backend and frontend.
- Implemented a parser for CSVs in **Java** and UI in **JSP**, enabling professors to grant assignment extensions via file upload.

Software Engineer Intern May 2024 - Aug 2024

Carnegie Mellon University - Software and Societal Systems Department *Pittsburgh, PA*

- Used **Java** to implement a polymorphic module system in SASyLF, a proof assistant, drastically improving code reusability.
- Leveraged **JavaCC**, a parser generator, to implement a recursive descent parser for an extension to SASyLF's syntax.
- Engineered a highly efficient **deep cloning algorithm** for a class hierarchy with **100+ classes**, utilizing a **caching mechanism** to prevent redundant cloning and infinite loops, ensuring reliable duplication of abstract syntax trees.
- Developed a **substitution algorithm** designed to switch out parts of an abstract syntax tree with other nodes.
- Created a modular testing framework using **Python** to automate unit tests and integration tests, ensuring code reliability.

TA - Analysis of Algorithms & Data Structures and Functional Programming Aug 2023 – Present

Cornell Computing and Information Science *Ithaca, NY*

- Recipient of the **CS Course Staff Award** for providing exceptional service to students (awarded to less than 10% of TAs).
- Lead presentation-style recitations, giving clear and engaging explanations of course content to **30+ students** per week.
- Hold office hours to help students grasp rigorous proofs about **greedy**, **dynamic programming**, and **graph algorithms**.
- Assist in debugging code using **Python**, **Java**, **C++**, and **OCaml**, and evaluating projects, problem sets, and exams.
- Mentored 5 groups of 3-5 students as they completed final projects, providing technical insights and problem-solving advice.

Projects

SimpleML - Machine Learning Library | *Rust*

- Developed a **math library** with support for operations including PCA, SVD, differentiation, and matrix inversion.
- Implemented a **stochastic gradient descent optimizer** to enhance training efficiency and optimize functions.
- Utilized linear algebra, calculus, and probability theory to design and optimize **powerful ML models** that allow a user to make predictions from data. Models include neural networks, clustering algorithms, regression, and support vector machine.

LambdaScript - Interpreter for a Custom Functional Programming Language | *TypeScript, OCaml, Jest, OUnit*

- Engineered a recursive algorithm for executing programs, yielding a **50% performance boost** over Python.
- Leveraged **TypeScript types** and **ADTs** to design abstract syntax trees that model programs with low runtime overhead.
- Implemented an algorithm that computes the type of any valid expression in linear time, ensuring efficient type inference.
- Built a **recursive descent parser** supporting advanced types, expressions, and patterns that provides helpful feedback.

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

- Implemented a secure login system with **OAuth2** authentication, allowing users to stay logged in across sessions.
- Used **React** to implement a monthly calendar that displays a user's goals and tasks, simplifying progress management.
- Developed **TypeScript** APIs to integrate with a database, facilitating seamless **frontend** and **backend** communication.
- Engineered a **backend server** that interacts with a realtime database, allowing users to save plans for achieving goals.
- Leveraged the APIs and a database to develop live chat functionality, enabling low-latency communication between users.

CritterWorld - Evolving Artificial Life Simulator | *Java, JavaFX, SceneBuilder, Gradle, JUnit*

- Used **JavaFX** and **multithreading** to design a GUI that visualizes the interactions and evolution of organisms over time.
- Implemented an abstract syntax tree parser and interpreter for a programming language that models virtual organisms, encoding behavioral instructions and random genetic mutations; implemented A* and Dijkstra's shortest path algorithms.

Technical Skills

Languages: Python, JavaScript, TypeScript, Java, HTML/CSS, C/C++, Rust, SQL, Haskell, OCaml, Assembly
Frontend: React, Electron.js, Next.js, Redux, TailwindCSS, SASS, JSP, JavaFX, Framer Motion
Backend: Express.js, Flask, Firebase
Dev Tools and Libraries: PyTorch, SymPy, NumPy, AWS, Docker, Kubernetes, REST API, Linux, Git