Ryan Soe

+1 5103641905 | ryansoe26@gmail.com | linkedin/ryansoe | github/ryansoe

EDUCATION

University of California, San Diego Sept. 2024 - Jun. 2026
B.S.. in Computer Science GPA: 3.95/4.0

De Anza College Sept. 2022 - Jun. 2024
A.A. in Computer Science GPA: 3.87/4.0
Relevant Coursework: Data Structures, Algorithms, Object Oriented Programming, Systems Programming

TECHNICAL SKILLS

Front End: Javascript, ReactJS, HTML5, CSS3
Back End: Python, C/C++, Java, NodeJS, Express, Flask, TensorFlow
Development Tools: Mongo, Git, Bash, Jupyter

PROJECTS

Twitter Clone | Mongo, Express, ReactJS, NodeJS (MERN Stack) Oct. 2024

- Engineered a feature-rich social media application with over **10** key features, including user authentication, post creation, and real-time notifications, showcasing end-to-end full-stack development expertise.
- Developed reusable front-end components using React and optimized data fetching and caching with React Query, improving API response times by **40%** and enhancing user engagement.
- Integrated secure back-end functionalities, including password hashing, JWT-based authentication, and scalable APIs using Node.js and Express, processing over **500** user interactions in testing scenarios.

Memory Allocator | C, Bash, Git Oct. 2024

- Designed a custom memory allocator in C to dynamically manage a simulated heap of **4,096** bytes, implementing both malloc and free functions to handle allocation and deallocation requests efficiently.
- Implemented a **best-fit allocation strategy** that reduced heap fragmentation and ensured memory block alignment to 16-byte boundaries, complying with industry standards.
- Delivered a library capable of managing **dozens** of concurrent allocations while providing robust functionality like coalescing adjacent free blocks and maintaining metadata for each memory block.

Chess Engine | C/C++, Git May 2024

- Built a chess engine using C/C++, implementing Minimax and Alpha-Beta pruning algorithms to simulate optimal strategies by evaluating up to **15** layers of potential game moves and counter-moves.
- Processed and analyzed up to **2,000** possible moves per second, significantly enhancing performance by reducing unnecessary computations through over **50%** effective pruning techniques.
- Designed robust and scalable data structures to represent chessboard states, representing standard 64-square boards with **1,000+** configurations, ensuring flexibility for future integration with artificial intelligence enhancements or graphical user interface development.

Image Classification Model | Python, TensorFlow, Jupyter July 2024

- Designed and implemented a machine learning pipeline using a convolutional neural network to classify **60,000** images across **10** classes, achieving **85%** accuracy through data augmentation, dropout, and transfer learning.
- Optimized the model's performance with hyperparameter tuning and leveraged GPU acceleration to reduce training time by **40%,** showcasing its scalability and applicability in artificial intelligence-driven tasks for large datasets.

EXPERIENCE

Software Developer Cupertino, CA
DeAnza Hacks Sept. 2023 - Oct. 2023

- Collaborated in a cross-functional team of **5** to design and develop a web application improving accessibility to a digital library of **500+** books through screen reader compatibility.