# Ryan Tan

510-513-2250 | ryntn@berkeley.edu | linkedin.com/in/ryntn | github.com/ryantanliner

## Education

**University of California, Berkeley** Expected May. 2026

*B.S. Electrical Engineering and Computer Science - GPA: 3.6/4.0* *Berkeley, CA*

- **Relevant Coursework:** Structure of Computer Programs, Computer Architecture, Discrete Mathematics, Probability Theory, Data Structures and Algorithms, Object Oriented Programming, Linear Algebra, Computer Security*, Computer Networks*

## Technical Skills

**Languages**: Python, TypeScript, Go, C++, Java, JavaScript, C, HTML, CSS, SQL
**Frameworks**: React, Flask, Node.js, Express.js, Django, Tailwind, Spring Boot
**Tools**: AWS, Docker, MongoDB, VS Code, PyCharm, Eclipse, Jupyter Notebook, Google Colab
**Libraries**: PyTorch, Scikit-learn, NumPy, Pandas, Matplotlib

## Experience

**Coding Instructor** Sep. 2023 – May 2024

*Code Ninjas* *Fremont, CA*

- Designed and taught programming lessons in **Python**, **JavaScript**, and **C#** to 30+ students aged 5–14, boosting coding proficiency and project completion through hands-on activities tailored to individual learning styles.
- Provided one-on-one and group instruction tailored to individual learning needs, boosting student engagement and retention through personalized teaching methods.

**Machine Learning Researcher** Sep. 2023 – Dec. 2023

*Algoverse AI Research Program* *Remote*

- Collaborated with a group of 4 students to explore the effects of implementing emotional cues when interacting with LLMs.
- Developed a custom LLM by integrating emotional stimuli into prompts to enhance user interactions, fine-tuning the model using a combination of transfer learning and hyperparameter optimization techniques, measuring an improvement by **83%**.

## Projects

**Secure File Sharing System** | *Go* Feb. 2025 – Present

- Built file sharing system from scratch using Go with an invitation scheme for file sharing and access revocation.
- Built a client application that allows users to authenticate, save, load, modify, and share files with controlled access.
- Secured encrypted and marshaled data using HMAC tags and digital signatures, guaranteeing authenticity and integrity.
- Created comprehensive test suite to validate system's functionality, including edge cases and data confidentiality.

**K-Popify** | *React, TypeScript, Flask, Python, Pandas, NumPy, Spotify Web API* Dec. 2024 – Jan. 2025

- Developed a web application that recommends K-Pop songs based on user-inputted tracks, integrating the **Spotify Web API** to fetch track features and using **Flask REST APIs** for backend services.
- Optimized song recommendation accuracy to **92%** across **275 K-Pop** songs without requiring data from users by implementing an algorithm utilizing **Euclidean Distance** to analyze 9 distinct audio characteristics.

**Binder** | *React, TypeScript, Convex, Clerk, Python, Flask, OpenAI API* Oct. 2024 – Oct. 2024

- Led the full-stack development of a collaborative study platform that matches users into study groups by analyzing skill sets based on their performance on quizzes generated by **OpenAI API**.
- Improved group matching accuracy by **91%** by implementing **Bayesian Knowledge Tracing** algorithm to analyze individual quiz performance data.
- Designed a scalable backend infrastructure with **Convex** and **Clerk** for secure user authentication and optimized performance for over 100 users.

**Skincare Tracker** | *React, JavaScript, Node.js, Express.js, Tailwind, MongoDB* June 2024 – Sep. 2024

- Built a responsive web app for managing skincare routines using **React and Node.js**, providing an intuitive interface for tracking product usage and schedules.
- Reduced API response times by **54%** by optimizing **MongoDB** queries and implementing **REST APIs** with **Express.js**.
- Enhanced security with **JWT authentication** and ensured reliable API endpoints by integrating automated tests using **Supertest**.

## Extracurriculars

**Software Developer** Sep. 2024 – Dec. 2024

*Open Project Berkeley* *Berkeley, CA*

- Implemented an LLM-based movie recommendation system with team of 5 using FlagEmbedding, Hugging Face, and PostgreSQL with pgvector, enabling efficient retrieval of similar movies based on user queries.
- Designed and optimized a vector search pipeline by storing 1024-dimensional embeddings in PostgreSQL and performing cosine similarity-based retrieval, reducing query latency and improving recommendation accuracy.