Ryan Tan

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

Education

University of California, Berkeley Expected Dec. 2026
B.S. Electrical Engineering and Computer Science - GPA: 3.57/4.0 *Berkeley, CA*
Relevant Coursework: Structure of Computer Programs, Discrete Mathematics, Probability Theory, Data Structures, Object Oriented Programming, Linear Algebra, Computer Security*, Computer Networks*

Technical Skills

Languages: Python, TypeScript, C++, Java, JavaScript, C, HTML, CSS, SQL
Frameworks: React, Node.js, Express.js, Tailwind, Flask, Django
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 **Unity** 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 language model by integrating emotional stimuli into prompts to enhance user interactions, fine-tuning the model using a combination of transfer learning and hyperparameter optimization techniques.
- Evaluated the impact of emotional cues on NLP tasks, including sentiment analysis, measuring a **52%** improvement in user-friendliness.

Projects

K-Popify | *React, TypeScript, Flask, Python, Pandas, NumPy, Spotify Web API* | GitHub Link Dec. 2024 – Jan. 2025
- Built a web application to recommend K-Pop songs based on user-inputted tracks, integrating the **Spotify Web API** to fetch track features and utilizing **Flask** for a **RESTful** backend to deliver personalized song suggestions.
- 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.
- Reduced image load times from 2.5s to **600ms** by implementing **Amazon S3** bucket storage for album cover caching.

Binder | *React, TypeScript, Convex, Clerk, Python, Flask, OpenAI API* | GitHub Link Oct. 2024 – Oct. 2024
- Led the full-stack development of a collaborative studying platform for study group matching, leveraging **OpenAI API** and **Flask** to generate custom quizzes to assess user's skill sets, achieving **82%** accuracy study group matching.
- Improved predicting student's skill levels by **65%** by implementing **Bayesian Knowledge Tracing** algorithm that leverages individual student performance data from quizzes to identify complementary skill levels across multiple subject areas.
- Developed a scalable backend infrastructure using **Convex** and **Clerk**, enabling seamless user authentication while ensuring secure and efficient platform performance.

Skincare Tracker | *React, Node.js, Express.js, Tailwind, MongoDB* | GitHub Link June 2024 – Sep. 2024
- Built a responsive web app for managing skincare routines using **React** and **Tailwind CSS**, enabling users to track progress and develop consistency.
- Reduced API response times by **54%** by optimizing **MongoDB** queries and implementing **RESTful APIs** with **Express**.
- Implemented user authentication with **JSON Web Tokens (JWT)** and integrated API tests using **Supertest**, ensuring secure session management, data integrity, and reliable API endpoints.

Extracurriculars

Software Developer Sep. 2024 – Dec. 2024
UC Berkeley *Berkeley, CA*
- Developed an LLM-based recommendation system with a team of 7 achieving **82%** accuracy for **10,000+** queries by implementing a two-stage **Retrieval Augmented Generation** approach using **Python**, **PyTorch**, and **OpenAI API**.
- Streamlined data workflows with **Pandas** and **NumPy**, reducing computation time by **74%** and improving response efficiency for large-scale data processing.