JUSTIN PARDO

☎ 908 764 0319 | ✉ justinpardo2423@gmail.com | in linkedin.com/in/justin-pardo

SKILLS

- Golang, Python, Java, JavaScript, Typescript, C/C++, React, Express, SQL, Docker, Linux
- MongoDB, PostgreSQL, NodeJS, Rest API, Postman, NumPy, Tailwind, HTML/CSS, Git/GitHub

WORK EXPERIENCE

Union College: *Machine Learning Researcher* June 2024– August 2024

- Collaborated with a team of researchers refining object detection models with applications in sports and lip detection, drastically improved model classification resulting in confidence rates rising from 20% to 95%
- Optimized the training process by transitioning from in-house solution to Google Colab and implementing video compression techniques achieving a 300% increase in efficiency in training with Roboflow dataset
- Collaborated with the lead researcher to meet project deadlines by establishing a sustainable Git workflow. Managed pull requests, resolved merging conflicts, and educated fellow researchers on standard Git practices

EDUCATION

Rutgers University || New Brunswick, NJ September 2024 – May 2026
B.S Computer Science

Union College || Cranford, NJ September 2022 – May 2024
A.S Computer Science

PROJECTS

Blog Aggregator | *Golang, PostgreSQL*

- Developed a robust RSS feed aggregator web server in Go, enabling users to add, follow, and unfollow feeds, providing a seamless experience for accessing content from various resources in one platform
- Integrated PostgreSQL for efficient data management, implementing database migrations to maintain data integrity and streamline updates across the backend system
- Implemented authentication via the chi package router only allowing authorized users to modify and delete feeds with graceful error handling when trying to interact with non-existing feeds or unauthorized routes

Interpreter | *Golang*

- Implemented core features such as lexical analyzer, parsing and evaluation using recursive descent to generate AST out of tokens delivering error logs for users
- Created robust unit tests for parsing system this allowed to catch mistakes early on in development resulting in 25% error reduction
- Optimized the interpreter by leveraging receivers and pointers for referencing and manipulating structs and interfaces, enhancing code reusability and performance.

Game Store | *React*

- Promoted reusability by using react router library outlet which eliminated the need to constantly trigger re-renders for persistent UI components when accesing various routes
- Improved loading times by up to 50% by caching Api calls to RAWG API allowing for reusability of data
- Developed an ASCII algorithm to derive consistent game prices, maintaining uniformity across different page sections despite varying Api requests