

# Jo Jojo

661-593-9466 | [jo\\_jojo@brown.edu](mailto:jo_jojo@brown.edu) | [linkedin.com/in/jo-jojo](https://www.linkedin.com/in/jo-jojo) | [github.com/JoJojo1256](https://github.com/JoJojo1256)

## EDUCATION

### Brown University

May. 2027 (Expected)

*Bachelor of Science in Math-Computer Science; GPA:3.9/4.0*

*Providence, RI*

**Relevant Courses:** Data Structures and Algorithms, Foundations of Systems, Computer Vision, Design and Analysis of Advanced Algorithms, Abstract Algebra, Operating Systems, Applied Cryptography, Graph Theory

**Summary:** Math-CS student at Brown University with full-stack development, OS design, and AI-workflow experience.

## EXPERIENCE

### Software Engineer

Sep. 2024 – Feb. 2025

*Kyron Medical*

*Providence, RI*

- Developed a scalable claims automation system using **Node.js**, **Python**, and Availity's API, enabling secure, HIPAA-compliant processing of EDI transactions for up to 500 patients concurrently
- Increased claim verification speed by 30% by optimizing backend workflows with asynchronous processing
- Implemented an AI-powered denial handling bot using **LLaMA 2-13B** and **AWS Lambda**, automating communication between insurers and hospitals to resolve claims faster

### Software Engineer Intern

May 2024 – Sep. 2024

*Pleio Inc.*

*New York, NY*

- Produced custom **React** web applications integrated with Pleio's **Microsoft SQL Server** and **Salesforce CRM**, automating data entry and workflows, significantly reducing manual effort and streamlining day-to-day operations
- Integrated **Microsoft Azure** for cloud storage, improving data flow efficiency, and for centralized user permissions management, enhancing security with role-based access control and multi-factor authentication
- Fine-tuned a Random Forest Machine Learning Model using **TensorFlow** in Python, optimizing hyperparameters (`n_estimators`, `max_depth`) with Grid Search to predict employee retention at 92% accuracy

## PROJECTS

### Weenix Operating System | *C, Linux, GDB*

Jan. 2025 - Present

- Implemented process management system, including process creation, scheduling, and context switching between multiple user processes and kernel threads in a Unix-like environment
- Designed and implemented a preemptive, priority-based thread scheduler, managing multiple threads and processes to ensure efficient CPU time distribution, with support for process states such as running, blocked, and ready
- Created kernel-level device drivers to interface with hardware, implementing low-level input/output (I/O) operations for virtual devices and hardware peripherals
- Engineered the Virtual File System (VFS) abstraction layer, enabling the OS to support multiple file systems by defining a unified interface for file system operations like `read()`, `write()`, and `open()`
- Developed a directory management system for SF5S, implementing `mkdir()`, `rmdir()`, `opendir()`, and `readdir()` system calls to enable hierarchical directory structures and navigation within the file system
- Designed and implemented virtual memory management (VM) subsystem, including page table setup, page fault handling, and demand paging to provide efficient memory utilization and process isolation

### Pneumonia Detector | *Python, TensorFlow, React, Node.js, Express.js, scikit-image, scikit-learn*

Dec. 2024

- Developed a binary pneumonia detection classifier using TensorFlow and a VGG16 model, achieving 95% accuracy in classifying chest X-rays from a Kaggle dataset as positive or negative for pneumonia
- Added dropout layers in CNN architecture, reducing overfitting by 20% as measured by validation accuracy
- Generated saliency maps with TensorFlow's gradient-based methods, visualizing focus areas in model predictions

### KickIt! | *React, Chakra UI, Google API, Node.js, Express.js*

Apr. 2024

- Led the development of KickIt!, a FullStack web app using **Google APIs** for real-time geolocation and activity discovery to enable users to find recreational activities within a 5-mile radius
- Leveraged **RESTful HTTP requests/JSON parsing** to retrieve real-time restaurant data, processing over 5000 API requests daily and reducing data retrieval latency by 30%
- Constructed a user-friendly FullStack application using **React.js** and Chakra UI for the front-end, and **Express.js** for the back-end, enabling session management and supporting over 1000 concurrent users

## TECHNICAL SKILLS

**C/C++, JavaScript/HTML/CSS, Typescript, React.js, Express.js, SQL, Python, TensorFlow, Azure, Git, Github, AWS, Docker**