Samuel Adeoye

https://www.linkedin.com/in/sam-ayo-adeoye/ • https://github.com/sam-ayo • samuelayomideadeoye@gmail.com

Education

Memorial University of Newfoundland St. John's NL**,** Canada
Bachelor of Science (Honors) in ***Computer Science*** September 2020 – December 2024
Cumulative GPA: ***3.88/4.0***, Major GPA: ***3.94/4.0***
Faculty of Science Dean's List (2022, 2023, 2024), MUNCSS member

Relevant Courses: Object-Oriented Programming, Database Systems, Data Structures and Algorithms, Algorithm Design and Analysis, Computer Networking, Machine Learning, Software Engineering, Operating Systems, Web Programming.

Highlights

- **Proficient Software Engineer:** Over 3 years of experience building clean intuitive user interfaces and deploying web applications, command line tools with a strong focus on scalability, security, and cloud integration.
- **Strong Backend Development:** Developed secure and performant backend systems using Python TypeScript, Java, and Rust with frameworks like Express, Spring, and Rocket.
- **Strong Frontend Development:** Developed production-ready web and mobile UIs using React, Flutter, Cloudscape (Design system), GPUI (Built by the Zed team), React Native, Tailwind CSS.

Technical Skills

Programming Languages	Typescript, Java, Python, C#, Javascript, Rust
Backend Frameworks	Express, FastAPI, Spring, Rocket, NestJs, Django, NestJS
Frontend Frameworks	HTML, CSS, React (Typescript), Flutter, Next Js, GPUI
UI Design	Cloudscape, Shadcn UI
Cloud Technologies	Microsoft Azure, AWS, Supabase, Appwrite, Vercel
Server	Nginx
Databases & Data storage	PostgresQL, MongoDB, MySQL, Supabase, Cosmos DB, Azure Blob Storage
Relevant Skills	Vim motions, Giit, Shell scripting, Bash, Azure functions, Langchain, Tesseract

Work Experience

Software Engineer - **Enaimco** Jan 2024 – Present

- Currently developing AI solutions to extract metadata and properties of subsea assets from unstructured engineering documents using Python and LangChain
- Implemented a video processing pipeline using Azure Functions with queue triggers (comparable to AWS Lambda with Amazon SQS triggers) and Azure Blob Storage (similar to Amazon S3), supporting the upload and handling of 500+ videos averaging 2.3GB each.
- Utilized Tesseract OCR for handling text extraction from video frames.
- Employed queue triggers to process each video frame asynchronously, to ensure reliability for video processing
- Enabled GIS location extraction and transcript generation from subsea engineering ROV videos
- Integrated full-text search functionality using MongoDB Full-Text Search and Lucene search indexes, significantly enhancing information retrieval capabilities within the application.

Co founder & Software Engineer **- Notable AI (https://www.notableai.ca)** December 2024 – Present

- Co-founded and built an AI meeting-assistant platform with 800+ users, 3 paying customers, 1000+ recordings, and 300+ AI-generated chats.
- Developed the cross-platform Electron desktop app using React for the renderer layer and Flutter mobile apps for IOS/Android, including real-time audio capture and transcription.

- Engineered backend services in Node.js/Typescript with PostgreSQL, supporting authentication, and AI integration.
- Built a unique meeting-detection engine using mic-activation events + focused-window PID analysis to reliably identify user meeting platforms and prompt automated note-taking
- Created AI-powered summarization and chat features using Vercel's AI SDK, enabling automated notes and conversational insights.

Software Engineer Intern - **Enaimco** May 2023 – Dec 2023

- Led the development of report generation using Large Language Models (LLMs) on the application
- Enhance server safety by implementing validation measures for request bodies to the company's API
- Took ownership over and managed the development of vector semantic search in the application
- Participated in code reviews, provided and received feedback, and suggested improvements to my team member's code changes.
- Worked in an agile team and collaborated with multiple stakeholders, including product managers and QA Engineers.

Software Engineer **- Data and Image Analysis Group (DIAG)** May 2022 – Present

- Led the development of a visualization tool for AI researchers to explore medical image metadata using Next.js, improving research workflow efficiency.
- Developed a Python-based data synchronization tool to seamlessly sync data between virtual machines.
- Configured and implemented NGINX for deploying in-house developed websites, ensuring optimal performance and reliability.
- Actively participated in code reviews on GitLab, providing valuable feedback and suggestions to enhance code quality and team collaboration.

Projects

Ragask (AI copilot for researchers)

Tech Stack: Python, LangChain, FastAPI, OpenAI GPT-4 API, NextJS, React, Typescript, MongoDB vector database, Beanie

- Currently building an AI assistant for researchers that leverages natural language processing to aid in literature reviews and data analysis.
- Implements advanced LLMs using LangChain and OpenAI's GPT-4 API to provide insights and summarize academic papers, aiming to streamline the research process.
- Implemented a RAG based approach to information retrieval from academic papers using MongoDB vector database as memory store
- Implemented the application logic using RESTful API using FastAPI
- Used Beanie ODM as the Object Document Mapper for interacting with MongoDB in a Object-Oriented programming style.
- Developing an interactive web interface with NextJs for user-friendly interactions.

YouTube to Spotify CLI

Tech Stack: Rust, Rocket (Rust Web Framework), YouTube API, Spotify API

- Developed a command-line interface tool using Rust and Rocket to create Spotify playlists mirroring the content of YouTube playlists.
- Interacted extensively with YouTube's playlistItems API and Spotify's authorize, search, playlist, and tracks APIs for seamless playlist synchronization.
- Implemented using Rocket OAuth2 with PCKE flow authentication for secure access to user accounts on both platforms.

RegPo (COURSE CHECKER)

Tech Stack: Rust, Sled (Embedded Database), Twilio API, Cron Jobs

- Created a Rust-based polling mechanism to monitor course availability for university registration.
- Utilized Sled (Similar to Python Shelve) as a local key-value store for data persistence
- Integrated Twilio API to send SMS notifications when desired courses become available.