

OLIVIER JOHN NDJIKE NZIA

☎ (214) 392-7723 | ✉ Ndjikenzia.o@northeastern.edu | [in linkedin.com/in/ojn](https://www.linkedin.com/in/ojn) | [🌐 ojn.me](https://ojn.me)

EDUCATION

Northeastern University

Boston, MA

B.Sc. Computer Science | GPA: 3.6

Expected May 2025

- Relevant Coursework: Object-Oriented Design, Algorithms & Data Structures, Computer Systems, Networks & Distributed Systems, Web Development, Discrete Structures, Logic & Computation

TECHNICAL SKILLS

Languages: Java | TypeScript | Python | Go | C/C++ | SQL | GraphQL | Unix Shell | HTML/CSS

Frameworks/Technologies: PostgreSQL | Next.js | React | Express | Pinecone | Git | Bash/Zsh | Redis

EXPERIENCE

UKG - Software Engineer Co-Op

Jan 2024 - Present

- Worked within the **Source Control Management team** to automate migrations between multiple git instances
- Developed and maintained **Jenkins pipelines** to regulate company access groups and repository permissions

Generate Product Development Studio - Software Engineer

Student Activity Calendar [🔗](#)

Jan 2024 - Present

- Developing an application to facilitate **club management** and **student engagement** for the NEU student body
- Enabled **natural language search** through Pinecone vector embeddings to improve query performance
- Designed comprehensive **integration tests** to validate data flow and coherence between different backend systems

Compound [🔗](#)

Aug 2023 - Dec 2023

- Built Compound, a full-stack **financial literacy application** tailored to users' priorities and existing experiences
- Developed API endpoints using NextJS and NextAuth to transfer data from Prisma and PostgreSQL database
- Constructed E2E tests with JEST to simulate and ensure seamless user flows as well as secure data transfers

Khoury College of Computer Sciences - Computer Science TA

Sep 2022 - Dec 2023

- Taught Computer Systems and Discrete Structures concepts such as Assembly, virtual memory, and combinations
- Diagnosed and debugged code during weekly grading sessions to provide comprehensive feedback for 100+ students
- Offered individualized and group-led guidance to students during office hours and exam review sessions

PROJECTS

Taskmaster | PostgreSQL, Next.js, Node.js, Express, Redis

[GitHub](#) [🔗](#) | Jun 2023 - Aug 2023

- Engineered a full-stack issue and **project management interface** that enables users to create, assign, and track projects, tickets, and bugs using **Agile** and **SDLC** methodologies
- Designed a relational database normalized up to **3NF** using **PostgreSQL** to securely and efficiently store app data
- Implemented caching and cache invalidation through **Redis** to control traffic to PostgreSQL database, and improve app responsiveness while maintaining data integrity and validity
- Leveraged **Express** to build and optimize **REST API** routes allowing **CRUD** operations in PostgreSQL database
- Implemented authentication and role-based authorization by using hashes to enforce separate levels of privileges

Little Lens | ReactJs, JavaScript, GraphQL, Sanity.io

[Live Demo](#) [🔗](#) | Mar 2023 - Jun 2023

- Developed a responsive image sharing web application with search querying, commenting, and a live feed
- Designed schemas in a backend database using GraphQL and GROQ to send queries through the Sanity.io service
- Enabled authentication through **Google OAuth** to securely store user data and manage sessions for login/logout

Candid Conversations | JavaScript, Python, Sockets

[GitHub](#) [🔗](#) | Dec 2022 - Jan 2023

- Built a socket-enabled chat room application allowing for text communication between multiple clients over LAN
- Enforced **TLS** encryption protocol onto sockets to securely transfer data packets between server and clients
- Integrated an intuitive UI through JavaScript to streamline user interactions

Polished Pixels | Java, JUnit, Swing

[GitHub](#) [🔗](#) | Oct 2022 - Dec 2022

- Developed an image processing application with 11 different image filters using kernels and convolution matrices
- Implemented an intuitive CLI and a **Swing GUI** to enable interactions with the processor
- Streamlined conversions between different image formats to support seamless imports and exports
- Utilized **JUnit** and mock classes to test classes and interactions between model, view, and controller packages