

# NICOLAS GONZALEZ

Boston, MA | 305-502-2397 | [niko04.gonzalez@gmail.com](mailto:niko04.gonzalez@gmail.com) | [linkedin.com/in/gonzi09/](https://www.linkedin.com/in/gonzi09/) | [github.com/Gonzi09](https://github.com/Gonzi09)

## Education

### Northeastern University

Bachelor of Science in Computer Science

Sept 2025 – May 2027

Boston, MA

**Relevant Coursework:** *Object-Oriented Design, Data Structures and Algorithms, Programming Languages, Software Engineering, Linear Algebra, Data Science*

### Miami Dade College (The Honors College)

Associate of Arts in Computer Science

Aug 2023 – May 2025

Miami, FL

## Technical Skills

**Programming Languages:** Python, C++, Java, JavaScript, SQL

**Core:** Data Structures & Algorithms, Object-Oriented Programming, RESTful APIs, Unit Testing, Debugging, Backend

**Frameworks:** React, Flask, Node.js, TensorFlow, PyTorch, Scikit-learn

**Tools:** Git, Linux, Supabase, Vercel, Jupyter Notebook, Power BI, Figma

## Work Experience

### NovaBike Cali

Software Engineer

Nov 2025 – Jan 2026

Remote

- Architected and deployed full-stack e-commerce platform (React, TypeScript, Express, PostgreSQL) supporting 60+ SKUs with dynamic filtering, and indexed query optimization for low-latency API responses
- Implemented RESTful backend APIs with Prisma ORM and relational schema design, reducing average response latency by 35% through indexed queries, optimized joins, and request validation
- Designed structured audit logging system capturing IP, state transitions, and user metadata, improving debugging resolution time by 50% and enabling traceable production incident analysis

### Air Force Research Laboratory

Software Engineer Intern

May 2025 – Aug 2025

Rome, NY

- Built multi-modal ML classification pipeline (Python, PyTorch) processing 50K+ LiDAR, RF, and EO inputs, achieving 98.7% accuracy and reducing false positives by 18% through feature engineering and model tuning
- Optimized YOLO + DeepSORT tracking system in PyTorch, increasing multi-object tracking accuracy by 12% while sustaining 28 FPS real-time GPU inference under constrained hardware
- Designed scalable data preprocessing pipeline (Pandas/NumPy) handling 2M+ records with automated validation and schema checks, reducing downstream model training errors by 30%
- Benchmarked feature-based ML models against Topological Data Analysis approaches, improving classification stability across noisy sensor inputs by 9% through robustness evaluation

### NASA Marshall Space Center

Software Engineer Intern

May 2024 – Aug 2024

Huntsville, AL

- Engineered asynchronous Python data pipeline (Pandas, NumPy) processing 100GB+ operational records, reducing manual reporting workload by 40% and enabling automated near real-time analytics
- Optimized ETL transformations on 100K+ SharePoint records, reducing execution time from 60+ minutes to 1 minute (98% improvement) through refactored data transformations
- Designed semantic data model for PowerBI dashboards, enabling sub-5-second query latency through optimized schema design and efficient aggregation logic

## Projects

### DrSmile Web Page | *React, Firebase, Tailwind, RESTful APIs*

- Designed optimized Firestore data schema and integrated external REST APIs (Google Maps, WhatsApp), reducing dashboard load latency by 30% and improving data consistency across patient records
- Architected and deployed full-stack patient management system using React and Firebase (Authentication, Firestore), implementing role-based access control and validated CRUD operations, reducing manual scheduling workflow by 35%

### Urban Network Connectivity & Vulnerability Analysis | *Python, OSMnx, NetworkX, SciPy*

- Engineered large-scale urban graph analysis pipeline using OSMnx to model 200K+ road-network nodes, applying topology-driven DFS algorithms to detect articulation points and structural connectivity vulnerabilities
- Implemented spectral graph analysis via Laplacian eigenvalues (algebraic connectivity) using sparse matrix representations, quantifying network robustness and optimizing computation time by 40% on city-scale graphs