Niharika Bhasin

133 Bayridge Ave, Brooklyn, NY 11220

646-358-5226 nb4048@nyu.edu Portfolio linkedin.com/in/niharika-bhasin github.com/niharikabhasin

Education

New York University **Sep 2024 – May 2026**
Masters in Computer Science *Brooklyn, NY*

SRM Institute of Science and Technology **Sep 2020 – May 2024**
Bachelors of Technology in Computer Science Engineering *Delhi, India*

Technical Skills

Languages & Platforms: Python, C/C++, Java, JavaScript, SQL, Linux, Embedded Platforms
Geospatial & Algorithms: Computational Geometry, Spatial Indexing (R-Tree), Routing Systems, GIS Data, Mapbox
Systems & Architecture: Distributed Systems, Microservices, Event-Driven Architecture, Concurrency Control
Cloud & Infrastructure: AWS (Lambda, DynamoDB, EC2, S3), Docker, Kubernetes, Terraform, Jenkins, Git
Data, Backend & Streaming: FastAPI, Django, Node.js, REST APIs, WebSockets, PostgreSQL, PostGIS, Redis, Kafka

Experience

Graduate Assistant, Electrical & Computer Engineering Department **June 2025 – May 2026**
New York University *Brooklyn, NY*

- Engineered a high-performance backend infrastructure platform using **FastAPI, PostgreSQL, Redis, and Python** to ingest and process real-time **device telemetry**, automating scheduling and access coordination across **150+ concurrent users and distributed lab systems**.
- Designed and optimized low-latency **REST APIs** and concurrency-aware backend workflows on **Linux servers**, leveraging indexing, caching, and asynchronous task execution to **reduce peak-time data latency by 45%** for mission-critical operations.
- Built **automated Python data pipelines** and event-processing systems to extract, synchronize, and analyze **large-scale operational metadata**, improving overall system throughput by **35%** and ensuring high-fidelity tracking.
- Developed and drove **end-to-end feature implementations** from core server infrastructure to client access points, maintaining **99.99% data integrity** and robust audit logging for safety-critical hardware environments.

Software Engineering Intern **June 2025 – August 2025**
Toricent Labs *New York City, USA*

- Architected a highly scalable, high-performance serverless backend on **AWS (Lambda, API Gateway, DynamoDB)** to handle large-scale data routing, engineering for workloads exceeding **1M+ monthly API invocations**.
- Spearheaded a zero-downtime infrastructure migration to **AWS Cloud** using **Terraform (IaC)**, designing resilient server infrastructure that slashed operational expenditure by **30%** and optimized resource utilization.
- Developed **50+ production-grade microservices** in **Node.js and Python**, implementing complete end-to-end features with both client and server footprints for asynchronous data streams and enterprise integrations.
- Optimized data retrieval and system throughput by implementing **distributed caching via Redis (DAX)**, reducing P99 API latency by **25%** under peak algorithmic stress tests.
- Maintained **99.9% system uptime** by designing data-driven, event-driven workflows and full-stack observability utilizing **AWS CloudWatch**, structured logging, and real-time alerting systems.

Projects

SafeZone — *Geospatial Navigation & Routing Engine (Django REST, PostGIS, Mapbox)*

- Engineered a geospatial routing engine using **PostGIS and Python** to compute optimized navigation paths via **computational geometry and spatial indexing**, cutting routing latency by **40%**.
- Built event-driven backend workflows for customer-facing navigation, enabling real-time routing updates and asynchronous streaming telemetry ingestion across distributed **AWS** infrastructure.

CodeSync — *Real-Time Distributed System & Stream Processing (WebSockets, AWS, Docker)*

- Engineered a low-latency distributed synchronization engine using **WebSockets and AWS SQS** to process high-throughput telemetry streams with sub-**50ms** propagation delay.
- Built scalable, resource-constrained **Linux execution environments** using containerized runtimes on **AWS EC2** with asynchronous event-processing pipelines for high-concurrency workloads.

Ticket Finder — *High-Concurrency Distributed Platform (FastAPI, Redis, PostgreSQL)*

- Designed a high-performance backend using **FastAPI, Redis, and PostgreSQL** supporting **10k+ concurrent requests** with sub-**100ms** response latency under peak algorithmic workloads.
- Implemented distributed locking, Saga transaction patterns, and concurrency-aware backend infrastructure to ensure fault tolerance and strict data quality at scale.