# Arihant Birani

Atlanta, GA 30308

**Phone:** (917) 545-8005 **Email:** abirani3@gatech.edu **LinkedIn:** linkedin.com/in/arihant-birani

## Education

**Georgia Institute of Technology** **August 2022 - May 2026**

*Bachelor of Science in Computer Science, Minor in Mathematics*

- **Relevant Coursework:** Data Structures & Algorithms, Operating Systems, Computer Networks, Databases, Computer Organization, Machine Learning, Artificial Intelligence, Linear Algebra, Computer Architecture

## Technical Skills

**Languages**: Python, Java, C/C++, JavaScript/TypeScript, SQL, Bash
**Frameworks**: React, Node.js/Express, FastAPI, REST/WebSockets, PyTorch, scikit-learn
**Databases/Cloud**: PostgreSQL, MySQL, MongoDB, Redis, AWS (S3, EC2, Lambda), Azure (ADLS, Application Insights)
**Tools & Testing**: Linux, Git, Docker, GitHub Actions (CI/CD), Terraform, Postman, JUnit, pytest, Logging/Monitoring

## Experience

**American International Group** **June 2025 - August 2025**

*Data Science Intern* *Atlanta, GA*

- Built an LLM-based agentic pipeline leveraging **Palantir Foundry**'s data integration layer to automate loss description generation and validation across diverse claim types, cutting adjuster review time by **85%+**.
- Evaluated 100+ document-level field errors by Anthropic text extraction model across 5+ unstructured formats (PDF, HTML, etc.), conducting root cause analysis to diagnose inaccurate model outputs.
- Designed a structured error taxonomy and conducted frequency analysis on model-ground truth mismatches to guide fine-tuning efforts, resulting in a **30%** reduction in critical field extraction failure rates across underwriting submissions.
- Used transformer-based sentence embeddings to match DUNS identifiers across 500+ unstructured policy and claims documents, improving retrieval accuracy by **18%**, while greatly reducing manual review workload.

**The Travelers Companies** **June 2024 - August 2024**

*Software Engineering Intern* *Hartford, CT*

- Constructed the Get-Loss-Consultation API Endpoint for the OmniAct Web Application, retrieving user records corresponding to User IDs passed in as query strings, writing 20+ test case scenarios to verify its functionality.
- Queried **10k+** User Sessions using U-SQL from the OmniAct Production Database and displayed **40+** significant metrics on the central DynaTrace Dashboard, presenting 20+ new user insights in weekly meetings with the product team.
- Collaborated with QA analysts to align API schema design with UI requirements for seamless integration in production.
- Conducted End-to-End API testing by chaining multiple requests together in order to validate complex user journeys.

## Projects

**Algorithmic Matching Engine** **June 2024 - August 2024**

- Built a full-stack system to match Travelers mentors and interns using the Gale–Shapley stable-matching algorithm with ranked preferences and soft constraints (location, skills, availability), reducing manual matching time by **90%**.
- Designed a normalized SQL schema (mentors, interns, preferences) and implemented REST endpoints for data ingest, validation, and match execution with structured error handling, processing **500+** profiles per run.
- Developed automated matching workflows and API test suites in Postman, generating match summaries, conflict reports, and audit logs for coordinators.
- Implemented a React dashboard to upload CSVs, trigger match runs, and review results with sortable tables, filters, and conflict-resolution tooling, enabling staff to complete workflows without engineering support.

**Quantum Sequence Modeling System** **January 2025 - May 2025**

- Architected a modular simulation framework for a quantum-inspired LSTM, separating data encoding, circuit construction, variational gate logic, and evaluation into independently testable components.
- Implemented custom computational primitives (variational blocks, entanglement layers, parameter-binding utilities) using Qiskit, optimizing for composability and maintainability over raw model performance.
- Designed a batching and caching pipeline to run large-scale circuit simulations efficiently on Aer, reducing redundant transpilation and enabling fully reproducible experiment runs across **100+** circuit configurations.
- Built abstractions and helper APIs to generate circuits programmatically based on feature dimensionality, gate depth, and qubit connectivity, mirroring classical configurable system design patterns.