

Jinho Bae

📞 619-820-7352 ✉️ jinho.baej@gmail.com 🌐 github.com/jinhobh

Education

Cornell University

Expected May 2028

Bachelor of Science in Computer Science

Ithaca, NY

- **Coursework:** Computer System Organization, Machine Learning, Data Structures & Algorithms, Discrete Structures, Linear Algebra, Differential Equations

Technical Skills

Languages: Python, Java, C, Assembly (RISC-V)

Technologies: SQL/PostgreSQL, FastAPI, PyTorch, Docker, CI/CD(Github Actions), Linux (Ubuntu), bash

Experience

Crow (crowescrow.io) – Pre-seed Startup

May 2026 – Present

Software Engineer Intern

Boston, MA

- Built invite link feature for Crow's Deal Room, allowing users to setup escrow payments and send an invite to their counterparty via secure link without needing their wallet addresses.
- Implemented deal-room and invite models, APIs, and backend state transitions for invite creation, limited previews, expiration/revocation handling, and counterparty wallet attachment.
- Added invite-acceptance safeguards using hashed invite tokens, duplicate-accept handling, one counterparty per deal constraints, transactional updates, and activity records to prevent reused links and inconsistent deal states under concurrent conditions.

Computational Mechanics & Scientific AI Lab (CoMSAIL) - JXWang

Aug 2025 – Present

Undergraduate Researcher

Ithaca, NY

- Built a Python simulation system that models how a wildfire spreads across a 2D landscape and trains a drone agent to decide which regions to observe next to most efficiently reduce uncertainty.
- Converted wildfire time-series data into a solved finite-difference simulator using 2D heat equation and Navier Stokes to model heat diffusion and wind-driven movement.
- Designing the drone decision loop so the agent receives only limited local measurements of the fire, chooses where to fly next, and learns to collect observations that better reconstruct the full fire map under a limited flight budget.

Projects

TraceGraph | Python, stdlib, tomllib, pytest, argparse

- Built a static dependency analyzer for Python that models import dependencies as a directed graph, correctly resolving relative, dynamic, and TYPE_CHECKING imports to enable repository-scale dependency analysis.
- Implemented iterative Tarjan SCC detection with import-binding analysis to distinguish benign dependency cycles from those capable of raising ImportError, uncovering a type-checking misclassification affecting 41 imports in Flask
- Designed a graph-based test impact analysis engine over reverse dependencies and validated it against coverage.py on major open source software like Flask (491 tests, 24 first-party modules) and Requests (635 tests, 19 first-party modules), achieving 100% recall with 8–34% test suite reduction.

HookIt – Webhook Delivery Platform | Python, FastAPI, PostgreSQL, SQLAlchemy, Alembic, fly.io, Docker

- Built a webhook delivery service with at least once delivery guarantees; failed deliveries retry on an exponential backoff schedule (10s → 1hr cap) for up to 6 attempts, then dead-letter for one click manual redrive, so no event is ever silently dropped.
- Implemented concurrent worker scheduling and idempotent event ingestion using row-level locking and transactional deduplication, ensuring duplicate requests and competing workers resolve to exactly one persisted event and one delivery attempt.
- Secured the pipeline end-to-end: HMAC-SHA256 signing with timestamped replay protection, SHA-256-hashed API keys (256-bit entropy), and Fernet-encrypted webhook secrets at rest and verified correctness with 298 tests exercising the API, worker, and a real Postgres instance together in CI on every PR.