

John Huang

Software Engineer

johnhuang105@gmail.com · (484) 358-6506 · linkedin.com/in/john-huang105 · github.com/Joh221105 · johnhuang-portfolio.netlify.app

EXPERIENCE

University of Washington GIX — Student Software Developer

June 2026 – Expected Dec 2026

Python, FastAPI, SQLAlchemy, OR-Tools (CP-SAT), PostgreSQL, ChromaDB, Gemini, Next.js, TypeScript, Alembic, pytest, GitHub Actions

- Led development of a capstone operations platform built for 150+ students and 140+ mentors across two university programs, replacing a manual spreadsheet workflow with 8 backend services (197 API endpoints) and a single web app; merged two diverging codebases into one and removed 46K lines of dead code with no loss of test coverage.
- Rebuilt the automated team-assignment engine on a constraint solver (Google OR-Tools), replacing a heuristic that timed out after 60 seconds with one that produces a provably best assignment in under 10 seconds and returns the same result on every run, ranked against the coordinator's stated priorities.
- Consolidated nine separate AI service integrations into one shared Python library used by all 8 services, eliminating three hidden bugs in retry handling that had been stalling unrelated requests and burning through API quota.
- Grew automated testing to 938 tests behind a 12-job continuous integration pipeline covering all services, and fixed test suites that had been writing to live student data on every run, taking the build from five failing jobs to one.
- Closed the platform's main privacy and abuse gaps: student names anonymized before any AI call and scrubbed from existing records, limits on uploads and search requests to cap both memory use and API spend, and exported spreadsheets sanitized against formula injection.

Secondsite — Software Engineering Intern (Computer Vision)

June 2026 – Present

Python, OpenCV, NumPy, SAM2, Docker & Docker Compose, AWS (EC2, EBS), OpenDroneMap/NodeODM, NVIDIA Jetson (balenaOS)

- Deployed an automated site-mapping pipeline end to end, turning live crane-mounted camera footage into georeferenced aerial maps across 11 coordinated Docker services; migrated it from an 8 GB onboard edge device to AWS (EC2 and EBS) after diagnosing a memory ceiling the hardware could not clear.
- Benchmarked 6 AWS server configurations against processing time and map quality to find the cheapest setup that still met accuracy targets, cutting cost per site map from \$11.80 to \$2.90 (75%) in exchange for 40 minutes of additional runtime, and documented the cost-versus-speed curve so the tradeoff can be re-decided per site.
- Replaced hand-tuned image masking with an AI segmentation model (SAM2), chosen by comparing the quality of the final maps each candidate produced rather than proxy scores; traced repeated reconstruction failures to frame-to-frame instability and fixed them with a per-camera voting step.
- Built GPS-driven frame selection over 48,598 daily position readings to recover the crane's rotation center, where the obvious approach was 9 m off; used it to deliver the project's first survey-aligned map (1 cm per pixel, 475 of 476 images placed) and cut a 5.7-hour processing stage down to minutes.
- Consolidated duplicated computer-vision code into a shared library used by both the simulation harness and the production scheduler, growing the test suite from 42 to 210 tests across a 16K-line codebase and adding coverage for a silent failure that had been discarding every survey reference point without an error.

PROJECTS

Thoth — RAG Knowledge Capture Platform

Python, ChromaDB, Claude API, Sentence-Transformers, RAGAS, LangSmith, GitHub Actions

- Built a proof-of-concept RAG system for T-Mobile that captures expert knowledge through guided interviews, routing every entry through a three-stage review and approval process and citing its source on every answer it gives.
- Raised fact recall on long documents from 33% to 100% (14 to 43 of 43 gold facts) by indexing raw source chunks alongside LLM-written summaries, and cut retrieval time from 406 ms to 80 ms per question by embedding each query once.
- Built an evaluation suite that scores retrieval and generation separately: an offline benchmark of 119 labeled questions (0.86 MRR on the held-out split) gates every pull request in CI, and RAGAS scores live answers at 0.95 faithfulness.
- Traced every query in LangSmith and cut p95 latency from 19.3 s to 6.9 s (50% to 99% of responses under 8 seconds) at 59% lower cost by routing classification to Claude Haiku and answers to Sonnet instead of one model for every step.

InboxIQ — Full-Stack Email Platform

Next.js, FastAPI, Supabase (PostgreSQL/RLS), Gmail API, Vercel + Railway

- Designed and shipped an email triage product to production (sign-in, Gmail sync, AI prioritization, dashboard), delivering all 11 launch requirements and clearing 6 critical blockers after midpoint review.
- Integrated Gmail sign-in with credentials stored server-side only and per-user data isolation enforced at the database level, and moved mail syncing and triage to background workers so the app stays responsive while mail is processed.
- Developed a multi-step AI triage engine (classification, summarization, data extraction) with output validation, automatic failover between providers, and an automated evaluation harness holding accuracy at 85% or better.

EDUCATION

M.S. Technology Innovation, University of Washington (Expected Mar 2027) · B.S. Computer Science, Western Governors University (2024) · B.S. Biochemistry, Lehigh University (2021)

TECHNICAL SKILLS

Languages: Python, TypeScript, JavaScript, Java, SQL, Swift

Frameworks/Libraries: FastAPI, Next.js, React, Node.js, SQLAlchemy, Pydantic, OR-Tools, OpenCV, NumPy, Pandas, PyTorch, Tailwind

Infrastructure/Tools: Docker & Docker Compose, AWS (EC2, EBS), GitHub Actions (CI/CD), PostgreSQL/Supabase, ChromaDB, Alembic, pytest, Vitest, Vercel, Railway, Git/GitHub, RAGAS, LangSmith

Concepts: REST API design, distributed systems, constraint optimization, data structures & algorithms, computer vision, RAG and LLM pipelines