# Junxian (Lyken) Lin

+1 (672) 208-2190 | Vancouver, BC | jinxl2361418252@gmail.com | Personal Website

## SUMMARY

Full-stack AI Software Engineer specializing in full-stack AI automation, agentic workflows, and production-grade LLM applications. Hands-on experience architecting multi-agent systems, integrating *FastAPI/Next.js* stacks. Combines UBC data science training with proven internship execution in building automated *ETL* pipelines and robust deployment architectures.

## EDUCATION

**Master of Data Science in Computational Linguistics, The University of British Columbia** Aug 2025 – Jul 2026
*Relevant: LLM Applications, RAG, Agentic AI, Machine Learning, Python* *Vancouver, Canada*

**Bachelor of Economics, South China Normal University** Sep 2020 – Jun 2025
*Focus: Econometrics, Time-Series Analysis, Linear Algebra* *Guangzhou, China*

## TECHNICAL SKILLS

| | |
|---|---|
| Programming | Python, TypeScript, SQL, R |
| Retrieval & Embeddings | Sentence-Transformers, Pinecone, ChromaDB, Embedding Pipelines, Vector Retrieval |
| LLM / Agents | RAG, Agentic Workflows, Tool Calling, Prompt Engineering, LLM Evaluation, Workflow Automation |
| Backend & Frontend | FastAPI, SQLAlchemy, REST APIs, Next.js, React, Tailwind CSS |
| Deployment & Hosting | Azure, Docker, Ollama Cloud, Vercel, Render, Supabase, Neon, Cloudflare |

## WORK EXPERIENCE

**Operations and Data Management Intern** Sep 2024 - Nov 2024
*CITIC Bank International*
- Mapped legacy transaction reporting processes, identified inefficiencies in legacy transaction reporting and replaced manual Excel workflows with automated Python scripts.
- Engineered data reconciliation pipelines to automatically cross-reference core banking outputs with risk management systems.

**Full-stack Application Development Intern** Apr 2024 - Aug 2024
*Frontop Digital Creative*
- Built end-to-end multimodal document pipeline with PaddleOCR (OCR, layout & chart extraction), text chunking & image embeddings; indexed image-text nodes into FAISS + graph DB for hybrid retrieval.
- Designed prompt templates & citation schemas; integrated re-ranking, deduplication & context assembly; packaged into production retrieval/QA APIs powering the core search experience.
- Enhanced multimodal QA frontend with Next.js: improved image-text answer rendering, citation highlighting & one-click source navigation for better traceability.

## RESEARCH & PROJECTS

**AI-Powered Capacity Management & RFP Optimization Platform @ CGI** **Apr 2026 – Jun 2026**
*Full-stack AI product* *Azure OpenAI, LLM, Fast API, Streamlit, React*
- Built an end-to-end **workflow automation pipeline** to replace a highly manual, document-heavy resource tracking process.
- Leveraged **Generative AI** and **Python** to parse hundreds of unstructured Word documents, extracting specific consultant skills.
- Developed a dynamic frontend capacity **dashboard** using **React** and Streamlit, allowing business stakeholders to filter and query extracted data instantly, **automating 80% of manual tasks**.
- Provided AI-powered suggestions on project assignment to support data-driven decisions.

**Data Copilot** **Jun 2026 - Nov 2026**
*AI Workflows & Automations* *Python, SQL, Type Script, AI Agent, Low/No-Code, React, Next.js, Fast API*
- Built a bronze → silver → gold pipeline on **BigQuery** with dbt and **Dagster**; a DQ gate blocks publication when a test fails.
- Kept NL2SQL to gold marts only, enforced in **MartSource** and again at the IAM layer; capped spend with a BigQuery dry-run byte estimate before every statement and a maximum_bytes_billed limit.
- Added DQ triage for failed tests: one evidence query per failure gives the counts, a likely cause, and a PR-level fix proposal.
- Gave semantic metrics a draft / confirmed / stale state machine and checked expressions before execution, so definitions stay auditable.
- Wired quality into CI: 164 pytest tests, dbt build --target ci, and a Next.js build, plus a daily GitHub Actions refresh and a /health page showing freshness and DQ status.
- Benchmarked on BIRD Mini-Dev (500 questions, 11 databases): Execution Accuracy 0.750 (375/500) with a 98.8% submission rate.

**Open-Source Contributor | Agenta-AI/agenta (~4.9k GitHub Stars)** **Sep 2026 - Oct 2026**
*Open-Source Contribution* *Python, TypeScript*
- Fixed a cancellation-handling bug in the agent runner to ensure user-initiated stops during resource acquisition are correctly reported as cancelled rather than failed.
- Updated unit tests to verify cancellation behavior while preserving existing error-handling semantics.
- **Merged** PR #7476 · TypeScript

**Data Validation & RAG Pipeline Architecture** Jan 2026 - Apr 2026
*Agentic RAG System | LLM, Tool Calling, Evaluation Framework* *Python, LLM, Sentence-Transformers*
- Architected a Retrieval-Augmented Generation system for secure automated information retrieval.
- Mitigated AI hallucination risks by engineering an 'Automated Validation Check' using LLM-as-a-Judge.
- Wrote Python scripts to automatically cross-reference generated answers with source texts, flagging outputs with <90% similarity scores for human review.
- On a versioned 10-query benchmark (8 answerable, 2 unanswerable), the retriever scored Recall@5 1.00, MRR 0.94 and nDCG@5 0.95 with 1.00 no-match accuracy, while end-to-end Ragas scores were Faithfulness 0.92 and Context Precision 0.89, at 1–2 tool calls and 3–5 s per query.