Sicheng Ouyang

Email: sicheng.ouyang@uwaterloo.ca | Tel: (647) 531-8488 | GitHub: carols12352 | LinkedIn: sicheng-ouyang | Portfolio: sicheng.dev

Education

University of Waterloo, Waterloo, ON *Sep 2025 – Present*

Bachelor of Software Engineering, Expected Apr 2030

Competitive Programming: USACO Gold Division (2024) · CCC Junior Distinction (2024)

Skills

Languages: Python, TypeScript, JavaScript, Java, C++, SQL

Frameworks & ML: React, Next.js, Vite, FastAPI, Flask, PyTorch, Hugging Face Transformers

Data & Backend: PostgreSQL, PostGIS, MySQL, Redis, SQLite, Supabase, WebSockets

Tools & Infrastructure: Docker, Linux, Git, GitHub Actions, PyInstaller, Vitest, Playwright

Experience

Mui Scientific

SQL · REST APIs · ERP Integrations · Next.js *May 2026 – Aug 2026*

Software Engineer Mississauga, ON

- Automated batch ERP-to-inventory synchronization with vendor REST APIs, custom parsers, and SQL workflows, enabling traceability across **10,000+** serial/batch records and cutting audit preparation from **8 hours** to **30 minutes**.
- Implemented validation rules and mismatch alerts in third-party inventory tooling, reducing routine manual checks across procurement, warehouse, and sales records by roughly **80%**.
- Modernized **30+** corporate pages with typed Next.js templates and a scrypt/HMAC-secured content API, enabling staff to schedule notices while achieving a **95+** Lighthouse performance score.

Tencent Music Entertainment Group

Python · Speech Synthesis · Audio ML *Jul 2024 – Aug 2024*

Machine Learning Engineer Shenzhen, China

- Evaluated 10 speech-generation models across 20+ controlled experiments using standardized Mel-spectrogram and F0 preprocessing, eliminating underperforming combinations and selecting the final singing-voice generation model.
- Optimized inference and vocal-simulation parameters through iterative tuning, raising an internal speaker consistency metric by 20% and MOS from **3.43** to **3.63**.

Advanced A-Level Study Website

WordPress · PHP · MySQL · Tencent Cloud *Sep 2023 – Aug 2025*

Co-Founder & Lead Developer Wuhan, China

- Co-founded and operated an A-Level learning platform by leading WordPress/PHP development, MySQL operations, and Tencent Cloud deployment, reaching **1,000+** peak daily visitors and supporting **200+** students.

Projects

The Cobalt Guide

Next.js · TypeScript · Supabase/PostGIS · MapLibre · Redis Demo GitHub

- Engineered viewport-bounded search across **58,559** locations with PostGIS GiST indexes and MapLibre clustering, reaching CDN-warm p95 latency of **65 ms** for maps and **45 ms** for search in 30-sample production benchmarks.
- Hardened the backend with transactional PostgreSQL RPCs, Supabase RLS, and Redis fallbacks to provide atomic updates, row-level authorization, and graceful cache degradation; verified by **270+** Vitest tests and live database/Playwright suites.

Real-Time Chat Platform

FastAPI · Next.js · WebSocket · PostgreSQL GitHub

- Implemented cursor-based message recovery with PostgreSQL before_id and since_id queries, enabling reconnecting clients to retrieve persisted messages missed during disconnections.
- Secured real-time delivery with JWT-authenticated WebSockets, conversation membership checks, per-user connection caps, and rate limits, blocking unverified and unauthorized clients.

AI-Powered Cross-Platform Task Manager

React/Vite · Flask · PyTorch/BERT · PyInstaller GitHub

- Fine-tuned multilingual BERT on **20,000** bilingual task commands using separate intent-classification and BIO slot-tagging heads, mapping **5** actions and **7** entity types into structured JSON task updates.
- Collapsed a **2-process** web stack into **one** self-contained tray app for macOS and Windows by embedding the Vite UI and Flask/Werkzeug server in PyInstaller, with cached PyTorch inference and explicit model warm/unload controls.

Neural Network Playground

FastAPI · PyTorch · React Flow · Recharts Demo GitHub

- Converted React Flow JSON graphs into dynamic PyTorch models through FastAPI, validating layer order, dimensions, and activations before training across **five** datasets.
- Implemented cursor-based REST polling for incremental epoch metrics and rendered updates with Recharts every 1.5 seconds, enabling live monitoring and pause, resume, and stop controls.