

Jeffrey Yu

905-736-3398 | jeffrey.yu.w@gmail.com | linkedin.com/in/jeffrey-yu | github.com/JeffreyYu-Dev

EDUCATION

Western University

London, ON

Bachelor of Science in Computer Science

Expected Apr 2027

- **Relevant Coursework:** Software Design and Architecture, Object-Oriented Programming, Data Structures and Algorithms, Databases, Operating Systems, Computer Networks

SKILLS

Languages: Go, TypeScript, JavaScript, Java, Python, C++, C, SQL, HTML/CSS

Frameworks: React, Next.js, TanStack Start/Router, Hono, Tailwind CSS, JavaFX, Qt/QML, Wails

Databases: PostgreSQL, SQLite, Redis, Qdrant, Drizzle ORM, GORM

Developer Tools: Docker, Docker Compose, Git, Linux, Bun, llama.cpp, Claude Code, Cursor, ChatGPT

EXPERIENCE

Undergraduate Researcher

London, ON

Department of Computer Science, Western University

May 2026 – Aug 2026

- Implemented a two-tier LLM vehicle diagnostic pipeline from an IEEE Communications Standards Magazine paper, shipping a containerized TypeScript service that compresses ECU telemetry and driver profiles into few-KB queries able to survive bandwidth-limited satellite links
- Built retrieval-augmented generation over Qdrant vector search with attention-based query augmentation and reliability-weighted re-ranking, grounding severity-classified repair advisories in ingested manufacturer service documents
- Validated the pipeline against real EngineFaultDB sensor recordings rather than synthetic readings, replaying rich-mixture, lean-mixture, and sensor-fault scenarios end to end
- Instrumented per-request stage tracing streamed over WebSocket to an operator dashboard, and added graceful degradation at every external dependency so the pipeline stays inspectable when the vector store or cloud model fails

PROJECTS

Scribble, Real-Time Multiplayer Drawing Game | *Go, React, TypeScript, WebSockets*

- Built a real-time multiplayer game server in Go that runs each room as an isolated goroutine driven by a single command channel, eliminating lock contention across concurrent WebSocket connections, turn timers, and state broadcasts
- Designed a typed message protocol validated at the client boundary with Zod, and replayed canvas strokes from a server-side command log so mid-turn joiners render a board identical to every other player
- Covered the game engine with 1,600+ lines of Go tests and implemented transparent reconnection with exponential backoff and queued message replay

Cross-Platform File Sync Client | *Go, SQLite, GORM, Wails, React*

- Built a file synchronization engine in Go using FastCDC content-defined chunking over a rolling Gear hash, cutting upload volume through deduplicated delta transfers that ship only changed chunks
- Designed a platform-agnostic filesystem watcher API backed by per-OS implementations, building the macOS layer on native FSEvents with recursive tree watching, event coalescing, and drop detection that triggers full rescans

Collaborative Calendar | *C++, Qt/QML, TypeScript, Hono, PostgreSQL, Redis*

- Architected and implemented a full-stack collaborative calendar for a 2 person course project, owning the PostgreSQL schema, REST API, Qt/QML desktop client, and cloud deployment end to end
- Built a custom signup and authentication flow with JWT issuance, refresh, and Redis-backed session management, scoping multi-user event access through invite-code permission tiers

Virtual Pet Simulator | *Java, JavaFX, JSON*

- Led a 5 person team to on-time delivery by learning JavaFX ahead of the build, mentoring 4 teammates on the framework, and training the team on Git branching and merge conflict resolution
- Built the save-game persistence layer and exposed it as a simplified data-access abstraction, unblocking teammates from handling serialization and file state directly
- Introduced AI coding assistants into the team workflow to accelerate delivery across 5 contributors working against a one-week deadline