Shiba Wang

s2259wan@uwaterloo.ca — GitHub — LinkedIn — shibawang.ca

Education

University of Waterloo Waterloo, ON

Bachelor of Mathematics Co-op, Computational Mathematics & Mathematical Finance (GPA: 3.72) *Expected: 2027*

- Relevant Coursework: Data Structures & Algorithms, Computational Mathematics, Optimization, Financial Mathematics

Professional Experience

University of Waterloo, Centre for Extended Learning Sept 2025 – Present

Software Engineer Co-op *Waterloo, ON*

- Built academic review portal for Admins and Faculty; designed permission system where edit access is based on user role and document status, enforced at both React and .NET API layers, consolidating 15+ access checks into single reusable module.
- Implemented auto-save handling concurrent edits across 10+ form sections: debounced user inputs, queued overlapping requests to prevent write conflicts, with automatic rollback on server validation failure.
- Reduced multi-step approval to single atomic operation that updates review status and unlocks 4 dependent documents; cache invalidation propagates state change to all active clients immediately without page refresh.
- Wrote 57 unit tests covering full 5-role × 6-status permission matrix; tests encode expected access rights for each combination and validate boundary conditions on status transitions, running as CI gate on every pull request.

Fountain Health Technologies Inc. Jan 2025 – Present

Founder & Software Engineer *Waterloo, ON*

- Architected real-time queue management platform for walk-in clinics serving three client types (patient kiosk, mobile tracker, staff dashboard); implemented WebSocket pub/sub with server-side tenant isolation to guarantee cross-clinic data security.
- Eliminated race conditions in queue updates by replacing application-side read-modify-write with a single atomic SQL `UPDATE`; guaranteed position consistency under concurrent check-ins without application-level locks.
- Reduced ghost connections from 15% to under 1% via 30s heartbeat timeout and session cleanup on disconnect; cut queue lookup latency from 800ms to 12ms by adding composite index—`EXPLAIN` confirmed Index Only Scan.
- Achieved 99.2% SMS delivery rate notifying patients of real-time queue position while pushing delivery status to clinic dashboard via WebSocket; enforced idempotency keys to prevent duplicate sends, integrated webhooks for failure tracking.

Nanjing Xitai Trading Co., Ltd. Jun 2024 – Aug 2024

Software Engineer Intern *Nanjing, China*

- Built ETL pipeline in Python processing daily sales CSVs; validated schema and data types with pandas, logged malformed rows for review, and persisted clean records to PostgreSQL, reducing manual data entry from 2 hours to 10 minutes.
- Developed React dashboard backed by FastAPI serving regional sales and inventory metrics via REST endpoints; deployed on internal server with cron-scheduled data refresh, replacing weekly Excel reports with self-service access.

Projects

Low-Latency Order Book & Matching Engine | ***C++17, CMake, Google Test*** GitHub

- Built price-time priority matching engine supporting AddOrder, CancelOrder, and ModifyOrder; orders organized by price level using std::map with doubly-linked list, achieving O(log P) insert, O(1) cancel, and 12M orders/sec throughput.
- Eliminated runtime heap allocation by pre-allocating 100K Order objects in memory pool; AddOrder averages 86ns, CancelOrder 34ns; best bid/ask retrieved in O(1) via std::map::begin() with bids sorted descending and asks ascending.
- Implemented matching logic that walks opposite book from best price, filling resting orders in FIFO sequence; supports partial fills with remaining quantity re-queued at original time priority.
- Validated with 15 unit tests covering matching logic, partial fills, self-trade prevention, and invalid input rejection; operations return typed OrderResult enum for error handling.

Low-Latency TCP Gateway | ***C++17, Linux epoll, Google Test*** GitHub

- Built single-threaded TCP server using epoll (edge-triggered) routing binary messages to Order Book; handles multiple concurrent clients with 8.2µs P50 round-trip latency and 79K msgs/sec throughput.
- Designed length-prefixed binary protocol for AddOrder/CancelOrder messages; implemented TCP framing to handle partial and coalesced reads, with input validation rejecting malformed data before processing.
- Validated with 17 unit tests covering protocol serialization, TCP framing, and connection handling; message parsing benchmarked at 50M ops/sec with zero heap allocation on hot path.

Skills

Languages: C/C++, Python, SQL, Java, TypeScript, JavaScript, C#, R, Bash
Systems: Linux, TCP/IP, Socket Programming, `epoll`, Memory Pools, Binary Protocols, WebSocket, Concurrency
Tools: Git, CMake, Google Test, Jest, `gdb`, `perf`, Valgrind, Docker, PostgreSQL, Redis, CI/CD
Frameworks: React, FastAPI, Node.js, ASP.NET Core, AWS, Nginx