Christian Cortez

Software Engineer · Backend, Systems & Research Computing

Madison, WI christiancortez.co@gmail.com ccortez.com linkedin.com/in/christian--cortez github.com/co-rtex

Education

University of Wisconsin–Madison Aug 2024 – May 2028 (expected)

B.S. in Computer Science, Data Science, and Statistics

Coursework: Algorithms, Operating Systems, Database Management Systems, Machine Organization, Software Engineering, Linear Algebra, Probability & Mathematical Statistics

Research experience

Software Engineering Research Assistant — LiDAR Phenotyping May 2026 – Present

UW–Madison / USDA Collaboration, Madison, WI

- Build Python and Bash pipelines on Linux that have processed 1,950+ scans (about 1 TB) across 50+ projects, cutting runtime from 60–90 to 20–35 minutes.
- Developed the validation and research-delivery layer: plot-to-genotype map checks, QC artifacts, byte-identical map snapshots for reproducible reruns, and auto-generated genotype-ranking plots for UW and external researchers.
- Created a FastAPI pipeline-configuration app with schema introspection, validation previews, and locked-field enforcement, so researchers change algorithm parameters without hand-editing YAML. Covered by 360+ pytest tests.
- Added automated quality checks that flagged 90–110 anomalous scans, validated against manual review with precision and recall, and wrote the pipeline documentation merged into the main branch.

Technical Computing Engineer Nov 2025 – May 2026

Space Science and Engineering Center, UW–Madison, Madison, WI

- Automated command-line and data-pipeline workflows and supported 15+ researchers.
- Raised first-pass pipeline completion to 90–95% and reduced manual run overhead by about 30%.

Industry experience

Software Developer Intern — Automation Team Jun 2026 – Aug 2026

HCSC / Blue Cross Blue Shield, Chicago, IL

- Cut manual check-record verification from 5–10 minutes to under one minute for 8–15 internal UAT users, addressing a recurring cross-team bottleneck.
- Added validation, timeouts, retries, cancellation, and fallback messaging so slow or failing upstream calls degrade clearly instead of silently.
- Used parameterized SQL queries and a composite index, and wrote 20–30 unit and integration tests that caught edge cases before production.
- Added logging that reduced diagnosis time by 50–70%.

Software Engineering Intern — Backend & AI Oct 2025 – Mar 2026

Phoenix-Aid / AccuraTrials, Madison, WI

- Built NestJS/TypeScript REST APIs and versioned PostgreSQL schemas that held p95 latency under 300 ms across 20,000+ staging requests.
- Built an asynchronous LLM extraction pipeline that drafts eCRF study builds from clinical-trial protocols, with Redis-backed queues, retries, and output validation, helping cut first-pass study setup from eight hours to 40 minutes.
- Measured output quality at 91% exact match against 1,200 human-reviewed protocol elements and 94% validity across 300 generated edit checks.
- Wrote 250+ automated tests (86% coverage) for the NestJS/PostgreSQL platform.

Systems projects

EngineX | C++20, CMake, GoogleTest, POSIX sockets Oct 2026 – present

- Built a price-time-priority matching engine as a single-sequencer state machine with O(1) cancels, partial fills, idempotent request IDs, and SHA-256 state hashing; local matching-core benchmarks reach 8.2M commands/s at 417 ns p99.9 latency.
- Added a CRC32C-framed write-ahead journal that acknowledges only after fdatasync, with deterministic replay (3.6M records/s locally), torn-tail repair, and corruption refusal, verified by fork crash-point and SIGKILL recovery tests.
- Built a multithreaded TCP order-entry server with bounded queues, group commit for durable mode, and gap-free snapshot resync (85.7K req/s in memory with 8 local clients); tested with 151 GoogleTest cases, 100-seed property tests against a reference model, and ASan/TSan CI.

TaskForge | Go, PostgreSQL, Docker, OpenTelemetry Feb 2026 – present

- Engineered crash-safe, at-least-once job processing with a PostgreSQL state machine, transactional outbox, idempotent submission, worker leases, heartbeats, stale-attempt fencing, backoff retries, and a dead-letter queue.
- Wrote 1,000+ Go tests, including integration and failure-recovery suites; a local 12-worker benchmark completed 10,000 of 10,000 jobs through 24 injected worker kills, with tuned settings reaching 209 ms p95 dispatch and 11 s recovery.

QuantLab | Python, FastAPI, PostgreSQL, Redis / RQ May 2026

- Built a Python research backend that ingests and indexes 1M+ OHLCV records for reproducible portfolio backtesting and risk analytics, including volatility, beta, Sharpe ratio, drawdown, value at risk, and expected shortfall.
- Ran 30-second to 10-minute analytics jobs on 4–8 Redis/RQ workers with idempotency hashes, retries, timeouts, and job-state tracking; validated the 18-endpoint FastAPI/PostgreSQL API with automated tests and GitHub Actions CI.

Leadership & activities

Vice President of Technology, Technical Lead for TradeGrid Apr 2026 – Present

FinTech Club at UW–Madison, Madison, WI

- Review pull requests for 20+ student developers, and added GitHub Actions CI and the onboarding guide.
- Wrote the matching specification (14 correctness invariants, 7 worked examples) that the team's engine and tests implement.

Activities: JPMorgan Chase Code for Good 2026, Volunteer CS & math tutor (50+ students), Alpha Kappa Psi, ColorStack

Technical skills

Languages	Python, Go, C++20, TypeScript, C#, SQL, Bash
Backend & data	FastAPI, NestJS, .NET / Entity Framework, PostgreSQL, SQL Server, Redis / RQ, pandas, NumPy
Reliability & testing	pytest, GoogleTest, Property-based testing, ASan / TSan, OpenTelemetry, GitHub Actions CI
Systems & tooling	Linux, Docker, Git, CMake, POSIX sockets, AWS