Tristan Mitsuji ODonnell Software Engineer

Software Engineer focused on backend systems and data infrastructure. Experienced in building DAG-driven execution engines, structured ingestion pipelines, and reproducible analytics workflows. Strong emphasis on clean system boundaries, data integrity, and scalable processing. Seeking backend or data engineering internship opportunities.

CONTACT

Los Angeles, CA

tristanodonnell100@gmail.com
818-439-9662

linkedin.com/in/tristan-odonnell-dev
github.com/TristanODonnell
tristanodonnellportfolio.com

TECHNICAL SKILLS

Languages
Python • SQL
TypeScript • JavaScript
C++ • C#

Backend & Systems
FastAPI • SQLAlchemy • Alembic
REST APIs • Background Workers
State Machines • Idempotent Systems
DAG Orchestration • Retry Semantics
Transaction Management

Databases & Data
PostgreSQL • Data Modeling
Migrations
Pandas • NumPy • Parquet

Machine Learning
Scikit-learn • Random Forest

Frontend
React • HTML/CSS

Testing & Development
Testing (Pytest) • Git
Environment Configuration

Game Engines & Systems
Unreal Engine 5 (Blueprints + C++)
Unity (C#)

COURSEWORK

Data Structures
Discrete Structures
Computer Architecture & Assembly
Microcontrollers & Embedded

EDUCATION

Bachelor of Science – Applied Computer Science
University of Colorado Boulder | 2026 – 2028

Bachelor of Science – Business Administration
California State Polytechnic University, Pomona | 2021

CERTIFICATIONS

Unity Developer Bootcamp
Offered via University of California, Irvine Continuing Education | 2021

PROJECTS

US Equities Trading Platform | Backend Architecture & Data Infrastructure | 2026 - Present
Python, PostgreSQL, SQLAlchemy, Parquet, Alembic, Docker

- Defined canonical trading **system contracts** (MarketBar, OrderIntent, BrokerOrder, Fill, RunManifest) with **strict invariants** and **idempotent execution semantics.**
- Architected **system-of-record storage layer** using **PostgreSQL** for transactional state and **Parquet**

Prediction Market Autonomous Trading Framework | *Backend Architecture & Data Pipeline* | 2025 – 2026
Python, FastAPI, PostgreSQL, SQLAlchemy, Alembic, React, TypeScript

- Architected a **modular, multi-stage backend system** spanning ingestion, factor scoring, decision gating, and execution, with strict separation between strategy logic and capital safety.
- Designed long-running **dual-loop worker orchestration** with idempotent execution guarantees, state reconciliation, and duplicate-trade prevention.
- Implemented **risk-gated execution layer** with paper/live modes, kill switches, per-market exposure caps, and daily trade limits to prevent unintended capital exposure.
- Engineered persistent state layer using **PostgreSQL with SQLAlchemy ORM** to manage positions, executions, and PnL reconciliation between in-memory and stored state.
- Built **config-driven strategy framework** and **monitoring API/dashboard** for observability and runtime tuning.

Config-Driven DAG Execution Engine | *Workflow Orchestration & Runtime Architecture* | 2026
Python, YAML, DAG Compilation, Retry Semantics, Run-State Persistence

- Architected a compile-then-execute **DAG workflow engine** that transforms declarative YAML pipeline specifications into validated execution plans with **deterministic ordering.**
- Implemented **dependency graph validation**, cycle detection, and **topological scheduling** prior to runtime execution.
- Designed **retry-aware runtime** with per-step state machine and structured run-state persistence (run_manifest.json).
- Built **artifact tracking** and **step registry abstraction** to decouple orchestration logic from execution logic and enable reproducibility.

Algorithm Selection Meta-Learning System | *ML Experimentation & Model Evaluation* | 2026
Python, Scikit-learn, NumPy, Random Forest, Feature Engineering

- Designed supervised **meta-learning system** to predict optimal graph search algorithm (BFS/DFS/A*) from structural maze features.
- Generated labeled dataset (144 mazes) via **oracle benchmarking** across solver performance metrics (path length, nodes expanded, runtime).
- Engineered **topology-aware feature set** (dead-end ratio, reachable ratio, shortest path length, open-space density) to encode search-space complexity.
- Trained and evaluated **Random Forest (200 trees)** achieving 100% test accuracy and zero average regret; introduced regret-based evaluation to measure real decision cost beyond accuracy.

PROFESSIONAL EXPERIENCE

Project Associate – Global Early Careers / Talent Brand
Activision Blizzard, Irvine, CA | 2022 – 2024

- Supported national internship campaigns that drove 3.3M site visits and 3K+ applications across 835+ schools through coordinated branding and outreach initiatives.