Ridham Patel

ridham412work@gmail.com • LinkedIn • GitHub • ridhamcodes.com

Education

University of Southern California — Los Angeles, CA
M.S. Computer Science — Expected May 2028

Arizona State University, Ira A. Fulton Schools of Engineering — Tempe, AZ
B.S. Computer Science — May 2026
GPA: 3.89/4.00 Dean's List

Vellore Institute of Technology (VIT) — Vellore, India
B.Tech. Computer Science and Engineering (Data Science) — August 2021 – August 2022
CGPA: 8.87/10

Completed first semester; audited advanced CS coursework (non-credit) while preparing for international transfer to Arizona State University.

Professional Experience

Research Aide — August 2025 – December 2025
School of Computing and Augmented Intelligence – Arizona State University — Tempe, AZ

- Trained and benchmarked computer vision and LLM-based models for biomedical imaging pipelines, comparing performance against established baselines across multiple architectures.
- Integrated large language model-based methods into biomedical processing pipelines and systematically assessed output quality, identifying failure modes and improvement opportunities.
- Reviewed 3–4 research papers weekly and authored technical summaries to inform model selection, experimental direction, and future research priorities.

Research Assistant — August 2024 – May 2026
Biodesign Department – Arizona State University — Tempe, AZ

- Developed a Rust–Python interface using Maturin to enable Python orchestration of high-performance simulation components, bridging ergonomic scripting with near-native compute speed.
- Implemented a full-featured web platform with Django and JavaScript for configuring, launching, visualizing, and interacting with simulation parameters in real time.
- Extended the core simulation framework to support spatial diffusion modeling, improving biological fidelity while refactoring code structure for long-term maintainability.

Machine Learning Intern — June 2024 – August 2024
WDWIL – Magik Kraft — Ahmedabad, India

- Built a full-stack system using Django and JavaScript to support real-time defect detection workflows, serving as the operational interface between ML models and production data streams.
- Deployed end-to-end training and inference pipelines on AWS (EC2, Docker, Kubernetes) with GPU-enabled instances, enabling scalable model training and low-latency production inference.
- Implemented model serving and evaluation pipelines with TensorFlow and PyTorch, integrating unit tests and CI/CD to ensure production-grade reliability and reproducibility across development and production environments.

Live Technical Projects

Snap2Plan View project

Next.js, TypeScript, Supabase, PostgreSQL

- Built a full-stack SaaS task management platform using Next.js 16 App Router, Supabase (PostgreSQL + RLS), and TypeScript with CI/CD via GitHub and automated Vercel deployment, integrating Claude and Gemini Vision APIs to extract structured tasks from handwritten notes, photos, and voice input in under 10 seconds.
- Architected a real-time Kanban board with drag-and-drop (dnd-kit) and Supabase Realtime WebSocket subscriptions, Google Calendar OAuth 2.0, and an automated email reminder pipeline (Brevo SMTP + Vercel cron) delivering deadline notifications at 72h/24h/2h intervals.

HF Market Microstructure Signal Platform View project

C++20, Python, FastAPI, React

- Built a C++ engine benchmarked at 1M LOB snapshots/sec computing OFI, VPIN, Kyle's λ, realized volatility, and Hawkes-process intensity.
- Streamed analytics through FastAPI/WebSockets and validated numerical components with Python and Catch2, including Avellaneda–Stoikov and Cartea–Jaimungal models.

Lock-Free Limit Order Book View project

C++20, React, TypeScript

- Engineered a price-time priority matching engine via lock-free MPSC/SPSC queues (Vyukov/Lamport), achieving 2.6M+ orders/sec with sub-microsecond P99 latency and zero heap allocation on the hot path.
- Eliminated cross-core false sharing using `alignas(64)` cache-line isolation on all concurrent structs, reducing thread contention across MPSC producer threads.
- Parsed NASDAQ TotalView-ITCH 5.0 binary protocol to replay real exchange data, adding pre-trade risk controls and validating IOC/FOK/GTC order semantics against production-grade market feeds.
- Built a React/TypeScript trading dashboard streaming live order book depth, fill tape, VWAP, spread, and book imbalance at 60 Hz over a self-implemented RFC 6455 WebSocket server.

Technical Projects (Past)

Reinforcement Learning Market-Making Agent View project
Python, PyTorch, Plotly Dash

- Trained a PPO agent to quote BTC-USD bid/ask spreads from Coinbase Advanced Trade REST API data, achieving 2.4× higher mean episode PnL and 8% lower inventory exposure vs. a TWAP baseline across 100 out-of-sample episodes.
- Designed an Avellaneda-Stoikov reward function $r = \text{spread_pnl} - \gamma q^2 \sigma^2$ with a 20-dimensional microstructure state vector including Kyle's λ, realized volatility, OFI, and depth slope.
- Derived dynamic inventory skewing from Kyle (1985) and Avellaneda-Stoikov (2008), enabling asymmetric bid/ask offsets replicating the closed-form reservation price $r^* = m - q\gamma\sigma^2(T-t)$.
- Built a Bloomberg-style Plotly Dash dashboard (500 ms refresh) with real-time order book depth, agent quote overlays, and dual-axis PnL/inventory time series.

Statistical Arbitrage Research Platform View project
Python, Streamlit

- Built a production-grade pairs trading backtester using Engle-Granger & Johansen cointegration with event-driven dollar-neutral simulation and look-ahead-free Z-score signal generation verified via unit tests.
- Implemented a Kalman filter state-space model for dynamic hedge ratio estimation, replacing static OLS and eliminating stale beta during regime changes such as the 2020 COVID volatility spike.
- Constructed a risk-parity multi-pair portfolio engine with CVaR/ES (Basel III), HAC Sharpe (Newey-West), Omega Ratio, and 2,000-path Monte Carlo across 6 historical stress scenarios including GFC and 2022 rate shock.

PageCraft AI View project
TypeScript, React, Express

- Developed a prompt-to-code tool generating full-stack apps via GPT APIs using WebContainers for fully in-browser execution, cutting developer setup time by ∼60% with no remote builds.
- Built a React-based IDE with Monaco Editor, live preview, and XML-driven file generation executing 100+ shell-level commands per session, supporting real-time edits and iterative follow-up prompts.

Skills

Languages: C++20, Python, Rust, Java, C, C#, JavaScript, TypeScript, SQL, HTML, CSS

Frameworks & Libraries: React, Next.js, Redux, Django, Node.js, Express, FastAPI, PyTorch, TensorFlow, NumPy, Pandas, SciPy, Statsmodels, Scikit-learn, OpenCV, Streamlit, Gymnasium

Systems & DevOps: Linux, Docker, Kubernetes, AWS (EC2, S3), CI/CD, Git, Supabase, PostgreSQL, REST APIs, WebSockets, Multithreading, Lock-free Data Structures

Quantitative: Statistical Arbitrage, Market Microstructure, Reinforcement Learning, Time Series Analysis, Stochastic Calculus, Risk Management