RASHMI SHAW

Kolkata, India | shawrashmi7@gmail.com | +91-9051307659
GitHub | LinkedIn | LeetCode | Portfolio

EDUCATION

National Institute of Technology, Kurukshetra 2022 – 2026
B.Tech in Information Technology **CGPA: 8.0/10**

TECHNICAL SKILLS

Languages: Python, TypeScript, JavaScript, SQL
Computer Science: Data Structures, Algorithms, Object-Oriented Programming, Complexity Analysis, Problem Solving
Backend: FastAPI, Node.js, Express.js, REST APIs, WebSockets, Async Programming, SQLAlchemy
Distributed Systems: Apache Kafka, Redpanda, Redis, Event-Driven Architecture, Backpressure, Idempotency
Databases: PostgreSQL, MongoDB, Redis
AI Systems: LangGraph, LangChain, LLM Tool Calling, Agent Orchestration, Human-in-the-Loop
Infrastructure: Docker, Docker Compose, GitHub Actions, Linux, Prometheus, Grafana, OpenTelemetry
Testing: Pytest, Vitest, Integration Testing

PROJECTS

StreamAlpha – Real-Time Distributed Market Analytics Platform *GitHub*
Python | FastAPI | Kafka/Redpanda | WebSockets | Redis | PostgreSQL | Docker

- Designed and built an **event-driven distributed system** using Kafka/Redpanda for market-event streaming, FastAPI WebSockets for real-time client delivery, Redis for low-latency state, and PostgreSQL for persistent market aggregates.
- Implemented **per-client bounded asynchronous queues, symbol-based subscriptions, backpressure handling, and slow-client message eviction** to isolate overloaded consumers without blocking the entire real-time stream.
- Separated streaming and analytics workloads into independent services for event consumption, state updates, and OHLCV aggregation, improving scalability and fault isolation.
- Instrumented the system with **Prometheus metrics** for active WebSocket clients, delivered messages, dropped events, and queue depth; integrated Grafana/OpenTelemetry for operational visibility.
- Built automated tests and a configurable load-testing harness for evaluating latency and throughput behavior under increasing event rates.

Enterprise Agent Trust Platform *GitHub*
Python | FastAPI | LangGraph | PostgreSQL | Redis | SQLAlchemy | Docker

- Built a **9-node state-machine-based execution system** separating classification, planning, authorization, approval, execution, validation, recovery, audit generation, and completion.
- Implemented deterministic **RBAC, resource-ownership checks, policy validation, idempotent operations, and bounded retry/backoff** to improve reliability under failures and unsafe agent actions.
- Designed a **SHA-256 tamper-evident audit ledger** and human-in-the-loop approval workflow for sensitive operations.
- Created a controlled evaluation harness across **20 seeded synthetic scenarios**; guarded execution improved task success from **60% to 100%** and reduced unauthorized actions from **5% to 0%** within the benchmark.

RevenueGuard – Automated Revenue Recovery Platform *GitHub*
Python | FastAPI | LangGraph | SQLAlchemy | Experimentation | Next.js

- Built an event-driven backend system for detecting and prioritizing revenue leakage across payment failures, abandoned checkouts, cancellation risk, and refund workflows.
- Designed an **Expected Recovery Value** scoring pipeline using revenue at risk, recovery probability, and intervention cost to prioritize recovery opportunities.
- Implemented deterministic **control/treatment assignment** and holdout experimentation to measure incremental impact instead of attributing natural conversions to automated interventions.
- Added policy-controlled tool execution with discount caps, credit limits, contact-frequency constraints, and human approval for high-risk actions.

PROBLEM SOLVING

- Solved **500+ Data Structures and Algorithms problems** covering arrays, strings, linked lists, trees, graphs, dynamic programming, heaps, tries, binary search, greedy algorithms, backtracking, and sliding window.
- Strong foundation in **algorithm design, time/space complexity, data structures, and problem decomposition**.

LEADERSHIP

Core Member, Shiksha NGO

- Supported educational initiatives impacting **180+ underprivileged students** and coordinated **150+ volunteers** across scheduling, allocation, and program execution.