

Aryan Ali Khan

Rawalpindi, Pakistan | +92 345 7380005 | aryan.a.minhas@gmail.com | [Portfolio](#) | [LinkedIn](#) | [GitHub](#)

Skills

Programming Languages	C, C++, Java, Python, SQL, Assembly Language, POSIX
Frameworks & Tools	JavaFX, LangChain4j, Ollama, Apache POI, Apache PDFBox, pgvector, Docker, Maven, CMake, pthreads (POSIX)
Core Competencies	Data Structures & Algorithms, Object-Oriented Programming (OOP), Problem Analysis, Low-Level Logic Optimization
Technical Interests	Software Architecture, Full-Stack Desktop Development, Low-Level System Optimization, Algorithmic Problem Solving, AI-Augmented Systems, Concurrent & Distributed Systems

Education

National University of Computer and Emerging Sciences (FAST-NUCES) | Islamabad, Pakistan

Bachelor of Science in Computer Science 2024-Present

- **Current CGPA: 3.08**
- **Relevant Coursework:** Software Design & Architecture (SDA), Data Structures, Database Systems, Object-Oriented Programming, Operating Systems, COAL.

Technical Projects

Smart City Management System | C++, Custom Data Structures, Algorithms

- **Architected** a modular Smart City ecosystem simulating Islamabad's transport, medical, and housing infrastructure by manually implementing core structures without relying on the C++ Standard Template Library (STL).
- **Engineered** a high-performance routing network using Adjacency List Graphs and implemented Dijkstra's Algorithm to compute shortest-path transport queries.
- **Designed** an N-ary Tree to manage complex administrative hierarchies (Sector-to-House) and built a separate-chaining Hash Table for instant, collision-resistant retrieval of citizen and medical records.

AVAL Systems – Financial Reconciliation Engine | Java, JavaFX, PostgreSQL, pgvector, LangChain4j, Ollama, Docker

- **Engineered** a hybrid matching engine combining rule-based logic with semantic AI vectorization, embedding transactions via a local Ollama LLM for Cosine Similarity search in pgvector.
- **Architected** a data pipeline for Excel/PDF parsing (Apache POI/PDFBox), schema standardization, probabilistic match generation, and human-in-the-loop approvals.
- **Applied** GRASP and GoF design patterns across a multi-layer architecture; containerized the infrastructure and Ollama inference server via Docker Compose.

CityMind – Urban AI Simulation | Python, Flask, SocketIO, scikit-learn

- **Designed** an AI simulation solving 5 challenges using CSP (AC-3, MRV, LCV), MST + Max-Flow, Parallel Tempering, D* Lite + Genetic Algorithms, and K-Means + Decision Trees.
- **Implemented** all graph algorithms and data structures manually without external libraries; integrated a Flask/SocketIO backend with a real-time PIXIJS frontend.

ChronoRift – Multi-Process OS Game Engine | C++, POSIX, pthreads, Docker

- **Architected** a multi-process RPG across 3 Linux processes (Arbiter, HIP, ASP) communicating exclusively via POSIX Shared Memory and unnamed semaphores for IPC.
- **Implemented** a thread-per-NPC concurrency model with a stamina scheduler and a POSIX signal (SIGUSR1) mechanic for asynchronous thread suspension and full state recovery.
- **Engineered** a wait-for graph deadlock detector and a contiguous memory allocator for inventory swapping; containerized the application in Docker.