# Hannan Ali

Lahore, Pakistan | i.hannanalimirza@gmail.com | Website | GitHub | LinkedIn

## Education

**COMSATS University Islamabad, Lahore Campus** — **Lahore, Pakistan**
*B.S. in Software Engineering* — *Completed Jan. 2026*

- **Undergraduate Thesis:** *Reducing False Positives in Automated Web Vulnerability Scanning* (team lead, three-person team).
- **Systems Coursework:** Operating Systems, Computer Architecture, Distributed Systems, Computer Networks, Algorithms, Data Structures, Software Security, Database Systems, Probability & Statistics.

## Research Interests

High-performance computing, computer architecture, and operating systems, with emphasis on parallel and heterogeneous execution, runtime and distributed systems, and the experimental methodology used to evaluate them: benchmarking, scalability and latency analysis, cache locality, concurrency, and the security properties of systems software.

## Research Experience

**Undergraduate Researcher — Performance Evaluation of Execution Architectures** — **Lahore, Pakistan**
*COMSATS University Islamabad | Advisor-supervised independent study* — *2025 – 2026*

- Investigated how execution strategy governs the throughput and scalability of a security-scanning workload, treating sequential, multithreaded, and multiprocess configurations as the independent variable in a controlled experiment.
- Designed the experimental methodology: repeated trials under fixed workloads, instrumented measurement of wall-clock latency, CPU utilization, resident memory, and throughput, and scaling sweeps over concurrency levels.
- Analyzed results with non-parametric hypothesis testing (Kruskal–Wallis, Mann–Whitney U) to establish statistical significance and to localize bottlenecks arising from process-level parallelism and I/O contention.
- Implemented a reusable benchmarking harness automating experiment orchestration, result aggregation, and statistical analysis, so that the evaluation is reproducible across workloads and machines.

**Undergraduate Thesis Researcher — Reducing False Positives in Web Vulnerability Scanning** — **Lahore, Pakistan**
*COMSATS University Islamabad | Team lead, three-person team* — *2025*

- Motivation: automated web security assessment is limited less by detection coverage than by false-positive alert triage, which displaces cost from detection to manual triage.
- Designed an end-to-end vulnerability analysis pipeline combining active scanning (OWASP ZAP, Nikto) with passive crawling behind a modular Django service interface.
- Developed a weakly supervised confidence-estimation approach (Random Forest) over scanner behavioral and contextual features, suppressing 29.5% of alerts across 94,256 findings while preserving high-severity findings.
- Led experimental design, evaluation protocol, and written reporting for a three-person team.

## Publications

**Hannan Ali**. "Performance and Scalability Evaluation of the Nuclei Vulnerability Scanner Under Different Execution Architectures." *Manuscript under review.* — *2026*

- Characterizes the performance envelope of a widely used template-driven vulnerability scanner across sequential, multithreaded, and parallel-process execution strategies.
- Contributes an empirical scalability analysis over execution time, CPU utilization, memory consumption, and throughput, with statistically grounded comparisons between strategies, and an open benchmarking framework for reproducible evaluation of scanning workloads.
- **Status:** manuscript complete and under advisor review ahead of conference submission.

## Selected Research Projects

**Low-Latency Order Book for NASDAQ ITCH Market Data (C++)** — **2026**

- Designed and implemented cache-efficient data structures for an ITCH-protocol order book operating on a high-rate message stream, where per-message latency, not aggregate throughput, is the binding constraint.
- Explored memory-aligned layouts, struct packing, and data-oriented organization to improve cache locality and reduce pointer chasing on the hot path.
- Benchmarked layout and algorithmic variants against one another to attribute latency reductions to specific architectural effects rather than to incidental code changes.

**Sensor-Driven Embedded Monitoring System (ESP32, C++)** **2025**

- Designed an event-driven system on a resource-constrained microcontroller, with sensing, local decision logic, and network notification sharing a single execution budget.
- Analyzed and tuned compute, networking, and notification behavior under variable workloads to bound responsiveness and energy cost.

## Industry Experience

**Software Engineer** **Lahore, Pakistan**

*Utilexa Labs* *Jan. 2025 – Present*

- Implement backend services and APIs in Python and C++, with responsibility for latency, throughput, and resource behavior of the services in production.
- Analyze and tune server-side performance; work across the deployment path from design through containerized release.

**Freelance Software Engineer** **Remote**

*Independent* *Jan. 2023 – Dec. 2024*

- Designed and implemented backend architectures and REST APIs for international clients (Python, Django) — data model, request handling, and deployment configuration — with accompanying client applications in React and Next.js.

## Technical Skills

| | |
|---|---|
| **Programming Languages** | C, C++, Python, Bash, SQL, JavaScript |
| **Systems Programming** | Linux, POSIX APIs, memory management, concurrency and synchronization, containerization (Docker) |
| **Performance Engineering** | Benchmarking methodology, latency and throughput measurement, cache-aware data layout, profiling-driven optimization |
| **Parallel Computing** | Multithreaded and multiprocess execution models, scalability analysis |
| **Machine Learning** | Supervised classification, feature engineering, model evaluation (scikit-learn) |
| **Statistical Methods** | Non-parametric hypothesis testing (Kruskal–Wallis, Mann–Whitney U), experimental design |
| **Security Tools** | OWASP ZAP, Burp Suite, Nmap, Nuclei, Scrapy |
| **Embedded Systems** | ESP32, Arduino, sensor interfacing |