Shazer Ali

📞 +92-3165494937 ✉ 24100053@lums.edu.pk 🔗 https://www.linkedin.com/in/shazerali/ 🌐 https://shaz-hash.github.io/

EDUCATION

Lahore University of Management and Sciences Aug. 2020 – June 2024

Bachelor of Science in Computer Science *SCGPA/Percentage: 3.88/4.00 , CGPA/Percentage: 3.81/4.00*

RESEARCH EXPERIENCE

Research Associate June 2024 – Present

Network Lab & Blockchain Lab *LUMS*

- Worked on enabling real-time mobility support for edge applications in 5G.
- Creating Access Control Policies Pipeline over Ethereum 2.0.

Undergraduate Research Assistant Sept 2023 – May 2024

Network Lab *LUMS*

- Worked on optimising Android Kernel's MMU for federated learning environments.

Undergraduate Research Assistant May 2022 – Sept 2022

Blockchain Lab *LUMS*

- Explored potential attacks on existing Reputation models deployed over Ethereum 2.0

RESEARCH PROJECTS

Enabling Real-Time Mobility Support for Edge Applications in 5G June 2022 – Sept. 2024

Advisor: Dr. Zafar Ayyub Qazi *Network Research Group @ LUMS*

- **Research Goal:** To provide dynamic stateful mobile applications with lower latency on the network edge as users move across different edge sites. We aim to achieve this goal by proactively migrating the edge application's state before the event of handover.
- **My Contributions:**
  * Developed a two-step App State Synchronization by extending Redis with LFU-based monitoring, enabling proactive synchronization using early mobility predictions.
  * Modified Redis source code in C to implement asynchronous key migration with pThreads, achieving $1.7\times$ faster state migration without blocking the main event loop.
  * Developed a mobility handler module with APIs to receive call from the BS predictor, initiating proactive state synchronization over Redis.
  * Simulated state distribution on edge applications such as CarMap and EMP to analyze trade-offs during 5G handovers.
  * Reduced blocking state migration time by at least $2.5\times$, enhancing edge application performance during handovers.
- ***Under review by ACM-MobiCom 2025***

Exploring Federated Learning on Low-end Devices Jan. 2023 – April 2024

Advisors: Dr. Zafar Ayyub Qazi & Dr. Ihsan Ayyub Qazi *Networks Research Group @ LUMS*

- **Research Goal:** To analyze and mitigate the impact of federated learning on resource-constrained mobile devices by optimizing memory management to improve user experience and inclusivity.
- **My Contributions:**
  * Modified the FLOWER framework to enable background FL training on Android devices using WorkManager, allowing user interaction during training; contributions integrated into FLOWER's official codebase.
  * Implemented kernel-level memory logging via both App and tools such as ADB & Perfetto to analyze Android's memory management, identifying that KSWAPD evicted pages from foreground apps during FL training, increasing page faults by 20% and degrading user experience (TTID).
  * Developed a custom tool to calculate Working and Resident Set Size (WSS & RSS) for FL, revealing many unreferenced memory pages that could be safely evicted.
  * Created a custom Linux kernel to make the Kernel MMU FL-aware, reducing memory pressure on foreground apps and improving TTID by at least $1.1\times$ on low-end devices with minimal impact on FL performance.

Blockchain-Based AI Model Governance and Policy Compliance System   June 2024 – Present
Advisors: Dr. Basit Shafiq & Dr. Jaideep Vaidya

- *Blockchain Research Group @ LUMS & Management Science and Information Systems @ Rutgers University*
- **Research Goal:** To develop a system allowing users to create or access AI models through natural language queries, ensuring policy compliance via smart contracts on the blockchain.
- **My Contributions:**
  * Integrated OpenAI's LLM APIs to convert natural language queries into accurate BPMN composite models using Retrieval-Augmented Generation and few-shot learning techniques.
  * Implemented policy compliance mechanisms using smart contracts on Ethereum, utilizing ECDSA signatures and SHA-256 hashing to securely verify user credentials against XACML business policies.
  * Established a pipeline with asynchronous listeners for Ethereum events using WEB3 APIs, enabling real-time updates and synchronization between clients, servers, and the blockchain in a REACT and NODE.JS architecture.

Reputation Model on Ethereum 2.0   May 2022 – September 2022
Advisors: Dr. Zartash Afzal Uzmi & Dr. Basit Shaifiq   *Blockchain Research Group @ LUMS*

- Created a new reputation quantification protocol in smart contracts over Ethereum 2.0 to implement fairness in the reputation system.
- Tested the functionality of the reputation quantification model using Brownie and performed custom simulations of collusion, Sybil, and re-entrance attacks in Python. *Report Link*

OPEN SOURCE PROJECTS CONTRIBUTION

Flower: A Friendly Federated Learning Framework   August 2023 – September 2023
Flower Lab — CaMLSys LAB @ Cambridge   *github.com/adap/flower*

- Utilized Android Jetpack's WorkManager library to implement background training tasks, enabling federated learning processes to run reliably even when the app is in the background or the device is resource-constrained.
- Integrated **TensorFlow Lite** with Java to deploy a robust federated learning system on Android devices, facilitating efficient on-device model training on the CIFAR-100 dataset.
- Refactored the Android example within the Flower framework using modern Java technologies (e.g., **Android Jetpack Libraries**) to enhance user control over training parameters and improve the overall user experience.
- Implemented resilient training mechanisms by managing background threads and services, ensuring continuous model training that is robust to interruptions from network issues or device power constraints.

SOFTWARE DEVELOPMENT EXPERIENCE

GradAssist - ActualAIz   September 2024 – Present
gradassist.actualaiz.com   *Advisors: Dr. Ihsan Ayyub Qazi, Dr. Zafar Ayyub Qazi, Dr. Agha Ali Raza*

- Developed an LLM-powered chatbot to provide personalized educational assistance to graduate students, enhancing learning experiences through AI-driven interactions.
- Implemented a caching layer using Vercel's Redis KV to reduce load on SQL databases and decrease response times, optimizing backend performance.
- Utilized RAG with multi-vector querying and metadata filtering to minimize hallucinations and deliver accurate, personalized responses.

TEACHING EXPERIENCE

CS 3812: Introduction to Blockchain (Spring 2024)   Head Teaching Assistant
Course Instructors : Dr. Zartash Afzal Uzmi, Dr. Basit Shafiq, and Dr. Naveed-ul-Hassan

- **Optimized Existing Proof of Work Assignment :** Enhanced Python-based Bitcoin blockchain simulation to reduce block submission latency on the main chain over the Linux Servers on LUMS Network.
- Created Quizzes and Assessments on Blockchain System Architecture to test students for their theoretical concepts on different distributed consensus Protocols such as Proof of Work and Proof of Stake
- Coordinated TA team tasks based on course outlines, designed quizzes, led programming assignments, conducted tutorials, held office hours, and graded assessments.

CS 582: Distributed Systems (Fall 2023)   Teaching Assistant
Course Instructor : Dr. Zafar Ayyub Qazi

- **Developed New Test Cases:** Implemented additional **Go-Lang** test cases to validate students' **RAFT** implementations, ensuring correct leader elections after previous leader failures.
- Held weekly office hours, conducted assignment tutorials, and answer student Queries on Slack

CS 100: Introduction to Programming (Summer 2023)　　　　　　　　　　　Teaching Assistant

Course Instructor : Dr. Waqar Ahmed
- Developed C++ assignments and quizzes to strengthen students' understanding of memory management with pointers and translating real-life problems into programmable solutions.

Awards and Honors

- Graduated undergrad with an **Award of High Distinction** (2024)
- **Top 10%** in the Computer Science batch of 160+ students, LUMS, Class of 2024
- Placed on **Dean's Honor List** for 2020-2024
- **ACE of SPADES**, single-handedly designed and developed the entire website, earning recognition as the individual with the most significant contribution to the society's workload of hosting 1000 participants

Course Development Projects

- **Distributed, Fault-Tolerant Key-Value Store**　　　　　　　　　　　　　　*Go-Lang*
  Implemented a distributed key-value store using the **Raft consensus algorithm** for leader election and log replication, ensuring fault tolerance and high availability. *Github*
- **Animal Welfare Society Webpage**　　　　　　　　　　　*Node.js, React.js, MySQL, Git*
  Developed an interactive web interface using [**Web Technologies**], enhancing user experience for reporting animal injuries among LUMS students and streamlining the submission process.
- **Peer-to-Peer File Sharing System**　　　　　　　　　　　　　　　　　　*Python*
  Built a P2P file sharing system utilizing **Consistent Hashing** and **Chord's finger tables**; assessed robustness by simulating node failures and network partitions to ensure reliability. *Github*
- **Reliable Chat Application on Custom Transport Protocol**　　　　　　　　　　*Python*
  Created a chat application over a custom **TCP-like protocol** built on **UDP sockets**, supporting concurrent clients with reliable messaging and file transfer despite user failures. *Github*
- **UNIX-like File System**　　　　　　　　　　　　　　　　　　　　　　　*C*
  Implemented a UNIX-inspired file system using similar block structures; leveraged **C libraries** for memory management and abstracted low-level operations for file handling.
- **User-Level Threading Library**　　　　　　　　　　　　　　　　　　　　*C*
  Developed a threading library with application-level context switching using **register-based PCB storage**; implemented custom locks to prevent race conditions and ensure thread safety. *Github*
- **Command-line Shell**　　　　　　　　　　　　　　　　　　　　　　　　*C*
  Built a fully functional Linux-like shell supporting complex command **pipelining** and **chaining**, enabling users to execute multiple commands efficiently. *Github*
- **PCB File Storage**　　　　　　　　　　　　　　　　　　　　　　　　　*C*
  Developed a file storage system simulating Process Control Blocks (PCBs) for efficient storage and retrieval operations. *Github*

Technologies

Languages: C, C++, Golang, JavaScript, Java, Python, TypeScript, Bash, Solidity
Frameworks, Libraries, & Tools: TensorFlow, TensorFlow Lite, Redis (Python), Memcached (Python), Scikit-Learn, Next.js, React.js, Redux, React Context API, Express.js, Node.js, Embedded JavaScript (EJS)
Databases & Tools: Redis, Memcached, Teradata SQL, Hadoop, MySQL, MariaDB, MongoDB
Cloud & Deployment Tools: Amazon Web Services (AWS) EC2, Apache, Linux (Arch, Ubuntu, Debian)