SARFRAZNAWAZ SHAHNAWAZ

+1-623-999-2785 • sshahna2@asu.edu • github.com/informsarfu • linkedin.com/in/sarfraz-nawaz • informsarfu.github.io

PROFESSIONAL EXPERIENCE

Graduate Teaching Assistant – Arizona State University, Tempe, AZ **Aug 2023 - Dec 2024**

- Advised 300+ students on Java fundamentals, Data Structures, and Object-Oriented Programming (OOP) concepts through 10 hours of weekly Q&A and review sessions, ensuring clarity and understanding of core topics.
- Mentored students via InScribe and Ed Discussion boards, conducted one-on-one sessions, and provided project feedback, resulting in a 15% performance improvement and a 95% retention rate compared to previous batches.

Software Engineer - Tata Consultancy Services, Bangalore, KA **Feb 2021 – Jun 2022**

- Reclaimed over 40 hours per month of maintenance by automating Kafka cluster management with Kubernetes operator and integrating AWS Lambda, AWS Batch, and Control-M, freeing developers from routine tasks.
- Facilitated Change Data Capture (CDC) between on-prem Oracle and Kafka on AWS, processing up to 100k transactions per second by implementing a scalable data pipeline, ensuring continuous data flow and integrity.
- Achieved 25% data accuracy improvement by monitoring real-time Kafka data streams with Dynatrace Cloud, AWS CloudWatch, and Grafana, ensuring zero data loss during pod failures through automated failover and replication strategies.

EDUCATION

Master of Computer Engineering **Jan 2023 - Dec 2024**
Arizona State University, Tempe, AZ **4.0 CGPA**

Bachelor of Electronics and Communication Engineering **Aug 2016 - Aug 2020**
Crescent University, Chennai, TN **7.9 CGPA**

TECHNICAL SKILLS

Programming: Python, Java, JavaScript, Typescript, Kotlin, C++, SQL, HTML/Tailwind CSS, Shell (Linux), MATLAB
Technologies: Django, Flask, NodeJS, AWS, Angular, ReactJS, GraphQL, Maven, Gradle, jQuery, D3.JS
Softwares: Dynatrace, Kubernetes, Docker, Selenium, Postman, SoapUI, Snowflake, Figma, Git, Firebase, Jira
Databases: MongoDB (NoSQL), MySQL, Oracle, SQLite, Elasticsearch, PostgreSQL
AI Tools: Pandas, PyTorch, Scikit-Learn, NumPy, NLTK, OpenCV, TensorFlow
Other: Agile (Scrum and Kanban), SDLC, CICD, Backend, Full-Stack, DevOps, Performance/Unit Testing, OOP, Jira

ACADEMIC PROJECTS

FrameSense AWS: A multi-tier architecture on AWS for video frame extraction and image detection.

- Configured a serverless workflow with AWS Lambda and AWS S3 to extract video frames in real time using an ffmpeg script in a Docker container, reducing processing latency by 30% through containerization.
- Streamlined communication between the Web Tier (EC2) and App Tier (EC2) via Amazon SQS Queues to facilitate auto-scaling of distributed resources, ensuring high availability and efficient handling of spikes in loads.
- Deployed a face recognition system using facenet_pytorch on the App Tier EC2, integrated it with AWS CloudWatch to monitor performance and bottlenecks, achieving 90% accuracy and enhancing system reliability.

Guardian Angel: an Android application built on Django and SQLite, focusing on driver safety and entertainment.

- Spearheaded the design of a region-based music recommendation model with 15% improved accuracy using Collaborative Filtering and Alternating Least Squares algorithms.
- Engineered a YOLOv4-based real-time computer vision model for collision detection, which monitors accelerometer spikes and dashcam dark-pixel density to trigger API services for live tracking and updates.

Schedule Planner: A web development application built on the FEAN stack for organizing weekly agendas.

- Developed REST API endpoints with CRUD operations and middleware for request-response validation and authentication using NodeJs and ExpressJs, to ensure consistent data flow between frontend and backend systems.
- Coded frontend with Angular, Bootstrap, and Server-Side Events (SSE) to show live updates to the event calendar.

eBay Clone: Crafted an e-commerce website using the MERN stack to replicate product purchase workflows.

- Implemented backend API operations with NodeJs and ExpressJs middleware, and designed a React-based frontend, storing and serving JSON data from AWS S3, enabling scalability and reliability.
- Orchestrated MongoDB for storing user data, optimizing performance with schema indexing, consistent hashing, and aggregation pipelines to process complex queries at up to 1,000 records per second.