Prasidh Aggarwal

Tempe, AZ | LinkedIn | paggar10@asu.edu | +1(929)6136103 | Portfolio

EDUCATION

MS in Computer Engineering - Computer Systems *Graduation Date: May 2024*
Arizona State University, Tempe, AZ *GPA: 4/4*
Experience: Teaching Assistant - MAT 117,142; Grader - MAT 242, 265, 267

BTech in Electronics and Communications Engineering *Graduation Date: August 2020*
Manipal Institute of Technology, India *GPA: 3.89/4*
Minor in Computational Mathematics (Graph Theory, Computational Probability, Time series)

SKILLS

Languages: Java, Python, C/C++, Bash, SQL, Javascript, MATLAB
Frameworks: Springboot, JUnit, Spring MVC, React, Node, D3.js
Tools: REST, AWS Lambda, AWS EC2, AWS SQS, DynamoDB, OpenStack, Linux, Git, Docker, K8s, Gradle, Cron
Core Competencies: OOPS, Microservices, Agile, Data Structures, Algorithms, Cloud Native development, CICD
Certifications: AWS Cloud Practitioner (Credential), Distributed Systems and Computing with Java, FCC Responsive Web Design

PROFESSIONAL EXPERIENCE

DC Software Engineer 1 **Bangalore, India**
Deloitte *Sept 2020 – Apr 2022*

- Collaborated with the product management teams to commence the SDLC cycle by designing 50+ UML diagrams, 250+ test cases using **JUnit**, **Mockito**, and Open-API contracts using **Swagger** for **REST** APIs.
- Designed, coded, and deployed 80+ RESTful APIs (ACH payments, credit card transactions, loan management, event notifications, transaction fraud detection, etc.) for the Banking Suite using Java **Springboot**, built using US banking cores - Mambu, Finxact, and Salesforce, which facilitated achieving four project MVPs.
- Accelerated the deployment of Microservices and Applications through streamlined pipelines using **JIRA**, **Concourse** CI, **Jenkins** CI, and **ArgoCD** for Banking Suite, which increased the development and deployment efficiency by 90%.
- Customized an integration between the **AWS Parameter store**, **AWS Secrets Manager**, and Springboot for the Banking Suite DevOps team, which enabled rolling restarts and automatic properties refresh for microservices, with 0 application downtime.
- Implemented an integration between **AWS MSK** and **Apache Kafka**, which commissioned an event-driven architecture for vendor notifications (specifically ACH and Credit cards), and reduced the API response times by 90%.
- Successfully launched a loan origination/management microservice, facilitating streamlined loan applications, refinancing, rescheduling, and amendments for 5000+ users with an 80% reduction in internal API calls.

Research and Development Intern **Manipal, India**
Manipal Institute of Technology *July 2020 – June 2022*

- Modeled and investigated the design of Ultra Reliable Index Modulation Schemes using **MATLAB**. Concluded using their BER v/s SNR graphs that Quadrature Shift keying provided 75% better performance over SSK and GSSK at the same spectral efficiency.
- Mapped various Space modulation techniques and reported that more the bits in the spatial domain (achieved through quadrature techniques), better the performance of the SMT.

ACADEMIC PROJECTS

Image Classification using IaaS | *Java, Python, AWS S3, AWS SQS, AWS CloudWatch, AWS EC2* | Link **Tempe, AZ**
Arizona State University *Spring 2023*

- Designed and implemented scalable AWS architecture for image classification using Python/Boto S3 and Java/AWS SDK, enabling efficient processing of over 100 image requests with sub-1 second average processing time.
- Enhanced system performance by optimizing auto-scaling policies, CloudWatch alarms, and implementing custom step-scaling based on m1-m2 metrics, reducing processing time by over 50% during testing.

Digit Classification using Offloading | *Java, Python, TensorFlow, NodeJS, Android Studio* | Link **Tempe, AZ**
Arizona State University *Fall 2022*

- Engineered mobile application for handwritten digit recognition using one master smartphone and four slave server devices, halving image processing time by 50% through distributed processing and optimized image segmentation.
- Orchestrated distributed ML models on 4 devices, one each for 4 quadrants of segmented image, achieving 75% faster predictions by training on subsets of MNIST and improving inter-device communication by 60%.

Other projects:

Responsive Websites | *HTML, CSS, JavaScript* | Link
Stack Hacking CTFs | *pwn, x86, Linux* | Link
Fibonacci Calculator | *PSOC, Embedded C* | Link
Automations | *Git, Python, AWS Lambda, Cron* | Link