Harsh Sanjaykumar Nagoriya

harshnagoriya@gmail.com — linkedin.com/in/harshnagoriya — harshnagoriya.github.io

Education

Master of Science in Computer Science   **3.94/4.0**
Arizona State University, Tempe, AZ   *Aug 2021 – May 2023*
Relevant Coursework: Cloud Computing, Data Processing at Scale, Software Testing/Quality Management

Bachelor of Technology in Information Technology   **8.14/10**
Dharmsinh Desai University, Nadiad, India   *Aug 2017 – May 2021*
Relevant Coursework: Data Structure, Algorithms, Databases, Advanced Java, Design Patterns, Distributed Computing

Technical Skills

Languages: Java, JavaScript, C/C++, Python, Bash, TypeScript, SQL, PL/SQL, C#, Prolog, Assembly Language
Database: MySQL, PostgreSQL, MariaDB, Oracle SQL Plus
Technologies/Frameworks: JSP, Servlets, JDBC, Flask, AJAX, Spring, Hibernate, Espresso, Appium, AWS Lambda
Developer Tools: Amazon Web Services, Jira, CircleCI, Netbeans, Eclipse, Visual Studio, JUnit, Git, Taiga, Scrum

Work Experience

Audere, Seattle, WA

Software Engineer, Backend   *Sep 2023 – Present*
- Collaborated with cross-functional teams to establish clear API specifications, ensuring the maintenance of 100% consistent and reliable interfaces across the board.
- Created and optimized more than 3 partner-facing API endpoints, enhancing system performance and delivering faster response times for business requirements.
- Designed and implemented Python scripts to automate the generation of database schema, resulting in 95% reduction in manual errors and saving over 15 hours of manual work per week.
- Maintained version-controlled Python scripts, enabling easy tracking of schema changes and promoting collaboration among 4+ team members.

SDET/QA 1   *Aug 2022 – Aug 2023*
- Demonstrated leadership abilities as the primary (lead) QA for Computer Vision and Android development, handling quality assurance activities and ensuring compliance with project deliverables and timelines for 3+ projects.
- Led and directed Android Espresso automation initiatives, establishing and executing robust test frameworks resulting in a 40% reduction in manual testing efforts and a 24% increase in test coverage.
- Created and executed over 200 mobile app test plans, ensuring high software quality and customer satisfaction.
- Collaborated with development teams to resolve 100+ bugs and improve software functionality.
- Monitored and reported on testing results, providing insights and recommendations to improve software quality, resulting in a 95% customer satisfaction rate.

Institute for Plasma Research, Gandhinagar, India

Project Intern   *Dec 2020 – Apr 2021*
- Enhanced the development of Ethereum blockchain by utilizing Solidity, JavaScript and Java and maintaining both client-side and server-side code. Complied with industry-standard software development procedures and crafted 5 UML diagrams to enhance comprehension and facilitate future maintenance.

Projects

Face Recognition as a Service ↗   *Jan 2022 – Mar 2022*
- Constructed a face recognition service based on Python, deployed on Amazon Web Services (AWS) EC2, incorporating the use of SQS and S3 for cost-efficient scaling.
- Automated load-balancing process with custom scripts, resulting in improved response time, reduced server downtime, and increased system reliability by 75%.
- Evaluated the scalability of the application by generating 1000 concurrent user requests, demonstrating its ability to handle high volumes of traffic efficiently.

Soccer Tournament Website ↗   *Sep 2021 – Dec 2021*
- Engineered a J2EE-powered Soccer Tournament Website for a diverse user base of 6 roles. Introduced innovative round-robin scheduling for 8 teams, ensuring equitable fixtures. Integrated real-time scores and live statistics, enriching the user experience with up-to-the-minute match insights.

Face Mask Detection System ↗   *Sep 2020 – Dec 2020*
- Developed a distributed application using ResNet-101 CNN, Amazon Lambda and IoT devices for real-time mask detection from live video recordings and improved IoT operations latency.
- Tested the AWS Lambda function with 500+ requests in 5 minutes, achieving 2s per response time.