ARYAN MURUGESH

+1 (515) 337-9991 | murugesharyan@gmail.com | San Jose, CA, USA | linkedin.com/in/aryan-murugesh-2a1154217

SKILLS

Skills: JavaScript, Blockchain, Cryptography, React.js, Git, Data Structures & Algorithms, Cloud, Google Analytics, Google Tag Manager, Springboot, Python, HTML/CSS, MySQL, Scikit-learn, FEMU, QEMU, Pandas, REST APIs, VS Code

EDUCATION

Iowa State University **August 2024 - May 2026**
Master's, Computer Engineering

PES University,Bengaluru,KA,India **January 2020 - May 2024**
Bachelor's, Electronics and Communication

PROFESSIONAL EXPERIENCE

Scaler Academy **Bengaluru, KA, India**
Software intern *January 2024 - March 2024*

- Launched automated lead nurturing sequences within HubSpot using Google Apps Script, resulting in 6.5 Lakhs ($7.8K) in revenue and shortened sales cycle by 2 days, improving team efficiency.
- Constructed interactive lead management dashboards utilizing Google Sheets and Apps Script, capturing insights from over 500 daily user actions, which improved sales cycle times by 12% lead tracking dashboards using Google Sheets and Apps Script, capturing 500+ daily user actions; this increased sales team efficiency by 15% by providing instant data insights.
- Automated WhatsApp Business Platform templates via Python scripting, enhancing CRM system responsiveness by 25% and cutting average user onboarding duration by 15 minutes.

Skycend **Bengaluru, KA, India**
Front-end Developer intern *October 2023 - December 2023*

- Boosted UI performance by 30% by refactoring into modules and refactoring key dashboard components using React.js, Tailwind CSS, and Chrome DevTools.
- Reduced client-side form error rate by 60% by implementing rule-based validations and real-time user feedback using JavaScript, HTML5, and regex patterns.
- Streamlined invoice data rendering by integrating RESTful APIs with React state management using Axios, enhancing data sync accuracy by 40%.
- Delivered a bug-free release by writing and executing component-level unit tests using Jest and React Testing Library, ensuring zero UI regressions.

PROJECTS & OUTSIDE EXPERIENCE

Edge-Cloud Latency Optimization using iFogSim

- Reduced cloud execution latency by 35% by developing fog-first task scheduling policies in iFogSim (Java) using event-driven offloading logic.
- Expedited simulated edge environment deployments across five nodes via the development of 15 reusable Ansible script modules, decreasing node setup time by 60% and accelerating testing cycles.
- Authored and presented a comprehensive 6-page technical evaluation report using LaTeX, incorporating Draw.io diagrams and Matplotlib benchmarking visuals to document system performance and identify optimization opportunities.
- Forged a fully simulated, low-latency secure authentication system utilizing Java, benchmarking at 2ms latency under 10,000 concurrent users, outperforming industry standards by 15% and reducing authentication failures.

Design and implement an efficient fingerprint authentication algorithm using SHA-512

- Pioneered a secure biometric login system using SHA-512 hashing in MATLAB and Verilog, thereby solidifying defenses against unauthorized access and mitigating vulnerabilities by an estimated 40%.
- Engineered parallel processing pipeline for data matching, improving processing speed by 30% and reducing average lookup latency to 2ms using ModelSim RTL simulation.
- Constructed fingerprint authentication system using MATLAB incorporating ridge flow filters and Gabor filters; enhanced system accuracy and decreased false rejection rate by 38%.

Ext4 Bug Classification and Runtime Simulation

- Classified 5,079 Linux file system patches with 88% F1-score using SVM classifier and TF-IDF vectorization implemented in Python (scikit-learn).
- Predicted performance degradation (IOPS/latency drop up to 47%) using regression models trained on FIO logs after simulated bug injection in QEMU + FEMU.
- Visualized results in Seaborn and Matplotlib, identifying 6 bug types that caused the highest performance degradation across 4 kernel scenarios.

AI-Powered Video Generation using Stable Diffusion

- Generated 12+ videos from text prompts using Stable Diffusion v1.5, chaining frames with Deforum and Hugging Face Transformers in Python.
- Reduced rendering time by 45% by integrating XFormers optimization, GPU acceleration on Google Colab, and frame batching via Python scripts.
- Enhanced frame coherence by 32% by tuning Deforum parameters like seed strength, noise step, and angle shift in animation settings.
- Automated video rendering pipeline utilizing FFmpeg and Python scripting, compressing the video generation timeline by 75% and curtailing operational expenditures on editing software by $15,000 yearly.