Akash Gajjar

📞 +1 (352) 709 8678 | ✉ akashgajjar8@gmail.com | in linkedin.com/in/akashagajjar | ⬯ github.com/skywalker212

Education

University of Florida	**August 2022 − Exp. December 2023**
Master of Science in Computer Science; GPA: 3.89/4.0	*Gainesville, Florida*
Dhirubhai Ambani Institute of Information and Communication Technology	**August 2015 − May 2019**
Bachelor of Technology in Information and Communication Technology; CPI: 8.31/10.0	*Gandhinagar, India*

Technical Skills

Languages:	TypeScript, Java, Python, C, SQL
Frameworks:	Angular, Express.js, Jasmine.js, Next.js, JUnit
Courses:	Algorithms, Advanced Data Structures, Database Management Systems, Introduction to Cryptography, Distributed Operating Systems Principles, Programming Language Principles, Computer Networks, Malware Reverse Engineering, Machine Learning
Tools:	Git, VS Code, Vim, Jupyter Notebook
Other:	Node.js, React, Redis, GraphQL, Apache Kafka, PostgreSQL, MongoDB, Kubernetes

Experience

Infosys Limited **October 2021 − July 2022**
Power Programmer Level II *Bangalore, India*
- Lead developer of a multi-million dollar revenue-generating web application development platform - Infosys Equinox Studio
- Responsible for site reliability (*99.9% up-time*) of the web applications deployed using the Infosys Equinox Studio platform
- Designed and developed telemetry capturing and processing pipeline capable of processing ∼*3000 Messages/second* using Apache Kafka, StreamSets Data Collector, and ELK stack
- Collaborated with technology architects to design the deployment flow for the Infosys Equinox Platform, reducing the deployment time from *3 hours* to *30 minutes*
- Mentored a Junior Developer

Infosys Limited **June 2019 − September 2021**
Power Programmer *Bangalore, India*
- Implemented RFC 6902 compliant JSON document versioning
- Developed a TypeScript library to generate a Server Side Rendered web application using a JSON configuration
- Decreased the site loading time from *20 seconds* to *1.5 seconds* by identifying and fixing computation-heavy code, configuring database indexing, making Kubernetes configuration changes, and setting up device based caching
- Developed an HTML Drag and Drop interface to generate JSON with a custom schema
- Implemented Distributed resource locking mechanism using Redis, AWS AppSync, and GraphQL Subscriptions

Dhirubhai Ambani Institute of Information and Communication Technology **Jan 2019 − May 2019**
Machine Learning Research Intern *Gandhinagar, India*
- Worked on Human Protein Atlas Image Classification problem
- Developed a Machine Learning model capable of localizing mixed patterns of proteins in microscopic images of human cell
- Studied existing methods used for protein classification
- Experimented with various hyper-parameters and features to study their effects on the accuracy of the machine learning model
- Implemented a Data Generator to prevent out-of-memory exceptions when training the machine learning model

Projects

Twitter Engine | *Erlang, Cowboy* **2022**
- Using actor model, designed and implemented a twitter engine providing basic functionalities via REST API and WebSockets: post and query tweets, subscribe to users' tweets and access to home feed
- Incorporated challenge based public key cryptography protocol, Diffie-Hellman key exchange protocol and HMAC signatures to make client-engine communication secure
- Developed a client simulator application to spawn clients and perform subscribe, tweet, retweet, and query requests to stress test the system and to generate performance metrics

Actor based Chord Protocol | *Erlang* **2022**
- Implemented, simulated, and analyzed Chord protocol for a peer-to-peer distributed hash table using the actor model in Erlang
- Developed a sample application on top of the chord to look up the value associated with a key

Programming Language Compiler | *Java* **2022**
- Defined lexical structure and phrase structure for a programming language with limited functionality
- Developed a Lexer to generate tokens from an input program
- Implemented a Parser to generate an Abstract Syntax Tree from an input of tokens
- Developed visitor interfaces to perform type checking and name resolution
- Implemented a code generator to generate Java bytecode using ASM framework by traversing the Abstract Syntax Tree