Siddhant Thakare

Riverside, CA | siddhant.thakare@outlook.com | in/siddhant-thakare | iambyt3z | siddhantthakare.github.io

Education

University of California, Riverside — **Sep 2023 – Dec 2024**
Master of Science in Computer Science — *GPA: 3.95 / 4*
Coursework: Advanced Algorithms, High-Performance Computing, GPU & CUDA Programming, Machine Learning & AI

BITS Pilani, Hyderabad — **Aug 2018 – Jun 2022**
Bachelor of Engineering
Coursework: Data Structures, Algorithms, Object-Oriented Design, Database, Operating Systems, Computer Architecture

Technical Skills

Programming Languages: C/C++ (2 yrs) · JavaScript & TypeScript (3 yrs) · Python (1 yrs) · Java (1 yrs) · SQL (1 yrs)
Frameworks/Libraries: React.js · Next.js · Tailwind CSS · Redux, Express.js · Spring Boot · Flask · Jest (Unit Tests)
Tools/Technologies: Node.js · REST APIs · Web sockets · GraphQL · gRPC · Kafka · Docker & Kubernetes
Database Frameworks: MongoDB · PostgreSQL · MySQL · Redis · DynamoDB (AWS)
Development Tools: Git · Terraform & AWS CDK (IaC) · Jenkins (CI/CD), SonarQube
Cloud Services (AWS): Lambda · EC2(Linux) · ECS · EKS · RDS · S3 · IAM · CloudFront · CloudFormation

Experience

Kfintech — **Jul 2022 – Aug 2023**
Software Engineer — *Hyderabad, India*

- Led the frontend architectural decisions for **3 SaaS products**, while managing 7 interns and working with 2 developers.
- Generated **30%** profit within the first quarter, by delivering a **multi-tenant SaaS solution** and Onboarding **4 clients**.
- Implemented **SaaS solutions** using **React, Next.js, Python, Flask, AWS Lambda, MongoDB, PostgreSQL, and AWS**.
- Administered the organization's **AWS** account by overseeing resource allocation, security, and **supporting 30 developers**.
- Decreased deployment time by **80%** through automated **CI/CD pipelines with Jenkins**, replacing manual deployment.
- Enhanced stack reliability by automating deployment of the architecture using **Terraform, AWS CDK & AWS CloudFormation**.
- Elevated code quality by integrating **ESLint, Jest, and SonarQube** into the team's **test-driven development workflow**.

Amazon — **Feb 2022 – Jun 2022**
Software Engineer Intern — *Bangalore, India*

- Enabled easier analysis of log files stored in **6 AWS S3 buckets**, by building a team's internal **S3 Document Viewer**.
- Cut the team's bug identification time by **40%** through **S3 Document Viewer's** integration into their **agile workflow.**
- Achieved a response time of **500 ms** and capability of handling **600,000 requests/min** for the Document Viewer.
- Created Document Viewer by leveraging **React & Java** for coding, and **AWS Lambda, API Gateway, and CloudFront** for cloud.
- Automated deployment of the Document Viewer's architecture in **under 2 min** using **AWS CloudFormation and CI/CD pipeline**.
- Lessened API performance metrics retrieval time by **80%** by creating **15 AWS CloudWatch dashboards** using **AWS CDK**.

Projects

Graph RAG Research Paper Summarizer | *Team Project (~ 80 Hours)*

- Implemented a **Graph RAG model** which gave **50%** more contextually complete summaries compared to **LLMs & normal RAG**.
- Built an app using **React frontend**, and **Python backend** and hosted on the **Google Cloud Platform** that allows users to upload research papers and use the **Graph RAG model** to get summaries and mind-maps of the uploaded papers under **1 minute**.

Whiteboard Application with Real-Time Sharing | *Personal Project (~ 50 Hours)*

- Enhanced engagement of remote teams by **30-40%**, through developing a whiteboard application with real-time sharing.
- Implemented the app utilizing **React, TypeScript** for frontend, **Firebase** for real-time database, **WebSockets** for instant updates.

High-Performance Stock Price Simulator | *Team Project (~ 35 Hours)*

- Built a simulator capable of simulating **10 Billion** stock price data points of **1000** companies over **15 sec**, using **C++** and **Linux**.
- Achieved a **reduction of 35%** in the runtime & **50% increase** in throughput, by implementing **multithreading using OpenMP**.
- Improved insights of stock price data, by utilizing **Python, Pandas and Numpy** to analyze & visualize the stock price data.