# Madhu Lakkoju

(934) 451-9569 madhu.lakkoju@stonybrook.edu linkedin.com/in/madhu-lakkoju github.com/madhulakkoju

## Education

**Stony Brook University, State University of New York** **August 2023 – December 2024**
*Masters of Science in Computer Science* *Stony Brook, NY, USA*
**Coursework**: Distributed Systems, Decentralized Systems, Operating Systems, Database Management, Computer Vision

**Jawaharlal Nehru Technological University Hyderabad** **August 2017 – July 2021**
*Bachelor of Technology in Computer Science* *Hyderabad, India*
**Coursework**: Computer Networks, Cloud computing, Databases, Web Technologies & Security, Operating Systems, Big Data

## Work Experience

**Software Developer Intern - WolfieONE** **August 2024 - Present**
*Stony Brook Medicine IT* *Stony Brook, NY*

- Building WolfieONE, an Oracle ERP system to address several business functions at Stony Brook University.
- Implementing real-time Data Integrations and Synchronizations module for Stony Brook Medicine IT - WolfieONE.
- **Tech Stack: Java, C#, .NET, Javascript, SQL, ETL pipelines, DB Design, Performance Tuning**

**Graduate Research Assistant - Distributed Systems Research** **Jan 2024 – Present**
*Stony Brook University* *Stony Brook, NY*

- Researching on core **Adaptive Blockchain Systems** under supervision of Prof. Mohammad Javad Amiri
- Developed core architectures and Byzantine Fault Tolerant protocols in performing State Machine Replication.
- Implemented Adaptivity in choosing these Architectures and Protocols based on real-time performance metrics.
- **Improved Throughput by 27%** over fixed architectures using a Reinforcement Learning Agent that learns optimal architecture - protocol combinations considering features such as contention, write ratio & latency in episodes.
- **Tech Stack: Java, Distributed Systems, Threads, Machine Learning, Python, gRPC**

**Software Developer** **Aug 2021 – Aug 2023**
*Standard Life UK | Tata Consultancy Services* *Hyderabad, India*

- Optimized SQL Queries and Angular UI, significantly **reducing load time from 3.5 to 1.8 seconds** with efficient data structures, async calls, and caching to improve experience of **12 million customers** & enhanced reliability by 22%.
- Designed and implemented event-driven features with **AWS Lambda, SNS & SQS** improving performance by 15-20% and custom **Glue ETL jobs** to perform data synchronizations between customer app and internal CRM systems.
- Implemented a new set of APIs with **API Gateway** which uses **Athena queries and KDAs** to generate real-time metrics from huge data streams and leveraged **DynamoDB** and **S3** to allow external systems to access results
- Designed and deployed Microservices architecture with REST API, improving overall website performance by over **30%** more than legacy MVC implementations (Metrics: FCP, FMP, LCP, load time, HTTP requests, errors).
- **Tech Stack: Angular, Typescript, JavaScript, Java, Spring Boot, AWS, SQL, REST API, Microservices**

## Technical Skills

**Languages**: Java, C++, Core Java, C#, Python, HTML/CSS, JS, SQL, NoSQL, React, Angular, Typescript
**Cloud**: AWS (Lambda, DynamoDB, SNS, SQS, Athena), Azure (Web Jobs, Logic Apps, Redis, SQL, Kubernetes)
**Technologies**: Spring Boot, ASP.NET, .NET Core, JUnit, Kafka, RabbitMQ, Docker, MySQL, Jenkins, GraphQL
**Certifications**: Cisco CCNA, MTA - Networking Fundamentals, PCAP: Python, Data Structures

## Projects

**APaxos** | *Distributed Systems, Java, gRPC, Multi-Threading, Shell Scripts, Database Management*

- Developed a variant of Paxos Protocol with XOV architecture similar to Hyper Ledger Fabric Blockchain.
- Implemented Leader Election, State Transfer, Synchronize states and handled Node Failure adhering to Paxos

**PBFT - Two Phase Commit**

- Impelemented PBFT, State replication, checkpointing, View-Change, Sharding, and acive Re-Sharding mechanisms
- Implemented Cross Shard Transaction management with 2-Phase Commit across multiple clusters in each transaction

**PAXOS - Two Phase Commit**

- Developed Paxos and configurable clustering mechanism, event-based re-sharding, and 2PC for Cross Shard Tnxs.

**Incremental Web Crawler** | *AWS, Lambda, Dynamo DB, CDK, Java, React JS, SQL, AWS SDK*

- Developed and deployed **Web Explorer**, an advanced web crawling application leveraging AWS features and Jsoup.
- Implemented robust rate-limiting to reduce 429 code on Too Many Requests to a server and optimized performance resulting in a **30% increase in crawl efficiency** and response times.

**Expense Management System** | *AWS, Lambda, RDS, React JS, Spring Boot, Mongo DB, Hibernate, Java*

- Multi-tier Full-Stack Web Application to display reports & visualizations on **React UI** using **MongoDB NoSQL** and **Spring Boot** to store and process transactions. Hosted on the Heroku server through CI/CD pipelines