ROHITH REDDY DEPA

+1 (346) 212-8961 ◇ rohithreddydepa@gmail.com ◇ linkedin.com/in/rohithreddydepa ◇ rohithreddydepa.github.io/latest-portfolio

EXPERIENCE

University of Houston   Oct 2023–Present
Full Stack Developer | Houston, TX

- Developed a modular, scalable UI in **Angular** with **lazy loading**, **reactive forms**, **route guards**, **RxJS observables**, dynamic rendering, and **OnPush change detection** for university-wide reuse.
- Built asynchronous, hypermedia-driven **REST APIs** with **Spring Boot**, implementing **HATEOAS**, **OpenAPI specs**, centralized error handling (**@ControllerAdvice**), and shared services for multi-department integration.
- Optimized **PostgreSQL** operations using **Hibernate ORM** with **Redis L2 caching**, improving performance by **50%**, enhancing **entity relationships**, **custom JPQL queries**, **connection pooling**, and ensuring transactional consistency.
- Deployed to **AWS** using **Docker** for containerization, integrated **PostgreSQL with RDS**, utilized **S3** for static assets, and implemented **CloudWatch** for real-time logging and autoscaling to ensure system resilience and uptime.

OpenText   Nov 2021–Aug 2023
Software Engineer | Hyderabad, India

- Developed a high-performance full-stack application using **React** and **Node.js** in an **Agile Environment**, improving **API response times by 35%** with **Redis cache** and enhancing **UI load by 40%** with **lazy loading** and **pre-fetching**.
- Designed and optimized a complex, modular UI using **React**, **Redux**, and **TypeScript**, implementing **server-side rendering (SSR)**, **dynamic routing**, and **code-splitting** for improved **SEO**, performance, and scalability.
- Enhanced **database performance** by implementing **materialized views** and **index tuning**, improving **report generation speed by 40%** and enhancing **query performance** on large datasets.
- Architected an efficient **ETL pipeline** using **AWS Glue**, **Lambda**, and **S3**, reducing **processing time by 60%**, while designing a scalable **CI/CD pipeline** with **GitHub Actions**, **Jenkins**, **Docker**, and **Kubernetes**.

TECHNICAL SKILLS

Programming Languages	Java, Python, JavaScript/TypeScript, C/C++, Bash, Shell
Web Development	Angular, React, Redux, Node.js, Spring Boot, GraphQL, REST APIs
Databases	PostgreSQL, MySQL, MongoDB, Redis, DynamoDB, CockroachDB, Firebase, SQLite
Cloud	AWS, Docker, Kubernetes, Terraform, GitHub Actions, Jenkins
Data	Apache Kafka, AWS Glue, Pandas, NumPy, ETL pipelines, Redis caching
CI/CD & Tools	Git, GitHub, GitLab, Postman, SonarQube, Jira, Confluence, VS Code, PyCharm, Eclipse

EDUCATION

University of Houston   May 2025
M.S. in Computer Science | GPA: 3.67/4.0
Relevant Coursework: Cloud Computing, Machine Learning, Artificial Intelligence, Database Management, Computer Architecture, Digital Image Processing.

PROJECTS

School Management Software

- Architected a school management platform using **Spring Boot** and **React** on **AWS**, implementing **JWT/OAuth2** for secure authentication, **REST APIs** for seamless communication, and **MySQL** for reliable data storage.
- Implemented **Redux** for state management, **role-based access control (RBAC)** for security, and **Docker** for scalable deployment. Integrated **Razorpay** for online payments, processing **1,000+ monthly transactions**.

ISP Messaging System

- Engineered a scalable messaging system using **Apache Kafka** for real-time microservice communication.
- Integrated **OpenAI API** for query resolution, reducing customer support workload by **40%** through automated responses.
- Utilized **PostgreSQL** for persistent storage and **Docker** to containerize services, enabling seamless deployment.

Traffic Asset Defect Detection for TxDOT

- Designed an **AI-powered computer vision system** using **YOLOv5** with **transfer learning** and **OpenCV**-based preprocessing to detect traffic asset defects, improving detection accuracy by **25%**.
- Automated defect reporting pipeline using **Python scripting** and scheduled batch processing, reducing manual inspection and documentation time by **40%**.