RAMA KRISHNA ARLA

Farmington, Michigan | +1 (313) 573 2271 | A black and white envelope

Description automatically generated [arla1r@cmich.edu](mailto:arla1r@cmich.edu) | A blue square with white letters

Description automatically generated [LinkedIn](https://www.linkedin.com/in/Ramakrishna-Arla/) |

# EDUCATION

**Central Michigan University Mount Pleasant, MI**

*Master of Science (M.Sc.) in information systems*  ***May 2024***

**Koneru Lakshmaiah Education Foundation Guntur, India**

*Bachelor of Technology (B. Tech) in Computer Science Engineering* ***May 2020***

# SKILLS & PROFICIENCIES

# Programming Languages*:* C, C++, Python, Java, Go Lang, HTML, CSS, JavaScript, SQL, PowerShell, Shell Scripting

# Web Development: HTML, CSS, JavaScript, Bootstrap, Angular, React

# Frameworks and Libraries: Spring Boot, Spring MVC, Spring Security, JUnit, JDBC

# Tools and IDEs and Operating Systems: Visual Studio IDE, Eclipse, STS, Docker, GIT, Postman, Linux, Windows

# Database Management: MySQL, MongoDB (NoSQL)

# PROJECTS

# Hospital Management System:

## Description:

Developed an integrated Hospital Management System designed to streamline operations for patients, doctors, and administrators. This system leverages a modern front-end framework with Angular, complemented by HTML, CSS, JavaScript, TypeScript, and Bootstrap for a responsive and interactive user experience. The back-end architecture utilizes Spring Boot, ensuring a robust and scalable application.

## Features and Functionality:

## Patient Dashboard:

Patients can easily register and log in to their accounts. They can search for doctors by specialty (e.g., urology) and book appointments directly through the system. Patients can also view their medical prescriptions and diagnostic reports, providing them with a comprehensive overview of their medical history.

## Doctor Dashboard:

Doctors can log in to view and manage appointments booked by patients. They also have the capability to update patient prescriptions and medical details, ensuring accurate and up-to-date records.

## Admin Dashboard:

Admins have the capability to add new doctors, update their professional details, and manage their salaries and other administrative information through the system. They also oversee the overall functioning of the hospital management system, ensuring smooth operations and efficient resource allocation.

# Rest Api:

## Description:

I created a RESTful API using Spring Boot to perform CRUD (Create, Read, Update, Delete) operations on a database. It provides endpoints for managing resources efficiently through HTTP methods. Built with Java and Spring Boot, the API ensures robustness, scalability, and flexibility in managing data entities.

**What I Did:**

I've implemented CRUD operations for entities using Spring Data JPA within my REST API project. Leveraging Spring Boot for seamless dependency management and auto-configuration, I've configured RESTful endpoints to handle HTTP requests efficiently. Additionally, I've ensured endpoint security by implementing appropriate authentication mechanisms where necessary."

**What It Does:**

My REST API project enables seamless creation, retrieval, updating, and deletion of resources via intuitive HTTP endpoints. It ensures robust data persistence and integrity by integrating with a relational database such as MySQL or PostgreSQL. Additionally, the API supports JSON as the data exchange format, facilitating smooth integration with front-end applications and other services."

# PROFESSIONAL EXPERIENCE

**Mphasis**

**Jr. Software Engineer Chennai, India**

***March 2021 – Dec 2021***

* Applied **OOP** **concepts**, collections, exception handling, I/O systems, annotations, concurrency, multi-threading, lambda expressions, and generics in various projects.
* Developed **JMS** publisher and subscriber components to facilitate asynchronous message processing.
* Developed robust error handling and retry mechanisms for JMS message processing.
* Optimized JMS message producers and consumers to enhance throughput and reduce latency.
* Documented JMS configurations and best practices to streamline development and maintenance processes.
* Designed and implemented **CI**/**CD** pipelines to automate the build, test, and deployment processes, ensuring rapid and reliable delivery of software.
* Automated code quality checks, unit tests, and integration tests using tools like **SonarQube**, **JUnit**, and **Mockito** within CI/CD pipelines.
* Developed and optimized Splunk queries, reports, and dashboards to provide real-time insights and actionable intelligence.
* Configured alerts and notifications in Splunk to proactively identify and respond to critical incidents and anomalies.
* Developed custom Splunk applications and add-ons to extend the functionality and tailor solutions to specific business needs.
* Conducted performance tuning and optimization of **Splunk** environments to ensure efficient data processing and query execution.
* Implemented performance monitoring and testing solutions using **AppDynamics** to ensure the reliability and scalability of applications.
* Configured health rules and alerts in AppDynamics to proactively detect and respond to performance degradation and anomalies.
* Integrated AppDynamics with **CI**/**CD** pipelines to enable continuous performance monitoring and testing throughout the development lifecycle.
* Utilized Eclipse, SLF4j, Log4j, JUnit, and Mockito for building, logging, and testing applications, including performance testing.