**Sahithi Ari**

**sahithiari16@gmail.com**

**+1 223 747 4369**

**LinkedIn: https://www.linkedin.com/in/sahithiari2/**

**Professional Summary:**

* Around 9 years of experience in Java/ J2EE Professional with proficiency in Analysis, Design, Development, Architecture, Coding, Testing, and implementation of Client/Server, Internet, and Legacy Applications.
* Expertise in Java Technology (J2SE and J2EE) with a strong foundation of object-oriented principles.
* Hands-on experience in creating Angular factories for using angular services like $http and $resource to make RESTful API calls to the Java-based.
* Experience in developing web applications using technologies like HTML5, Angular 8/9/11, JavaScript, AJAX, CSS, DOJO, JSP Tag libraries, Tiles, and XML.
* Solid Experience in creating Kubernetes Clusters in AWS and deploying a Docker Image on top of Kubernetes Cluster.
* Used Microservice architecture, with Spring services interacting through a combination of REST and Apache Kafka message brokers.
* Involved in using ReactJS components, Forms, Events, Keys, Routers, Animations, and Flux concepts.
* Used React JS to create views to hook up models to the DOM and synchronize data with the server as a Single Page Application (SPA)
* Experienced in Full stack UI technologies using Grunt.js, Gulp Server, and Yeoman generator for Angular8/9/11.
* Experience with Docker Containers, leveraging Linux Containers and AMIs to create Docker Images/containers and Kubernetes.
* Created the React JS components and triggered Angular code to render the React components using life cycle hooks.
* Used React JS to create views to hook up models to the DOM and synchronize data with the server as a Single Page Application (SPA) and React-Redux Thunk to make REST API calls.
* Worked and learned a great deal from Amazon Web Services (AWS) Cloud services like EC2, S3, EBS, RDS, VPC, and IAM, cloudformation, cloudfront
* Explored Spark, Kafka, and Storm along with other open-source projects to create a real-time analytics framework.
* Deployed the Cassandra cluster in a cloud (amazon AWS) environment with scalable nodes as per the business requirements.
* Utilized Kubernetes and Docker for the runtime environment and integrated them in AWS CodePipeline to automate CI/CD system to build, test, and deploy.
* Developed and maintained Docker images stored in Amazon ECR, ensuring streamlined build and deployment workflows.
* Configured AWS CodeBuild to build Docker images and push them to ECR, utilizing buildspec.yml for automated build processes
* Utilized AWS CodeCommit as a source code repository, managing version control and collaborating with team members on code changes, enhancing code quality and consistency.
* Integrated automated testing frameworks within CodeBuild to ensure code quality and security, reducing the incidence of bugs and improving overall application stability.
* Used Node JS to run Web pack tasks and build properly the project.
* Expertise in microservices to communicate through HTTP protocol for implementing SOA approach.
* Demonstrated expertise in managing Kafka topic partitions and message routing strategies for data distribution.
* Implemented Kafka's fault tolerance mechanisms, including data replication and leader-follower replication, to ensure data durability.
* Experienced in Client-side validations using Ajax in Node.js, Angular 8/9 and Dashboards reports using Backbone. JS.
* Developed API for using AWS Lambda to manage the Serverless architecture and run the code in AWS.
* Strong skills in object-oriented design and development (OOAD) using Rational Unified Process and Unified Modelling Language.
* Designed and implemented a dynamic image rendering web service that is consumed by JSP, jQuery, JavaScript, and AJAX.
* Performed code updates and releases using Git and Jenkins Deploy tools to send out updates to various pieces of software.
* Expert in working in an agile environment, Test Driven Development, and Scrum.
* Experience in SOA and implementing Web Services using SOAP.
* Experience in working with MongoDB, Oracle, SQL Server, MySQL databases, and writing SQL and PL/SQL queries.
* Hands-on experience in creating Angular modules, controllers, directives, and filters to deploy the entire business logic.
* Good understanding of NoSQL databases and hands-on work experience in writing applications on NoSQL databases like HBase, Cassandra, and MongoDB.
* Expertise in working with Collections API, Multithreading, and OOPs technologies.

**Skills**

Languages: Java, Java Script, JSP, UML, WSDL, HTML, CSS, jQuery, PL/SQL, Perl, Python, and Shell Scripting.

J2EE Technologies: Java, J2EE, JDBC, Hibernate, JSF, XML Parsers, JSP, Servlets, Portals, EJB, JMS, Java Swing, Struts, Spring Framework, Java Beans, Web Services.

Frameworks: Spring, Spring MVC, Spring Boot, NgRx, Spring Batch, Spring JDBC, Spring

Security, Hibernate, RxJs, Struts, EJB

ORM: Hibernate 3 and JPA 2

Web/App Server: Apache Tomcat, JBoss, Web logic, WebSphere, Glassfish.

Design Patterns: Singleton, Factory, MVC

Web Technologies: JavaScript, HTML, XML, XSL, XSLT, SOAP, Angular JS, Angular 8/9, Node JS, ReactJS, JSON, Restful web services (API), JUnit.

Web Services: SOAP (Axis) and REST (CXF and Jersey)

Unit testing: JUnit, Mockito, and Jasmine for JavaScript

Databases: Oracle, SQL Server, My SQL, PL/SQL, PostgreSQL, NoSQL, Mongo DB.

Tools: Tortoise SVN, TOAD, SQL Developer, Sub Version, SOAP UI, GitHub,

Splunk, Jfrog, UDeploy, Fortify Scan, and Black duck Scan

Cloud: AWS(ECS,EC2,EMR,ELB, Redshift,RDS,DynamoDB, Redis, S3, Lambda,

Glue, Clouformation, Cloudwatch,Cloudfront, IAM ), Azure

Development / Productivity Tools: JIRA, Jenkins, Eclipse, SQL Developer, WinSCP, Core FTP, Puppet, Putty,

SonarQube, Kibana, Grafana

Domain: Retail, Capital Markets, Security

Build Tools: Maven, Jenkins, Docker, Apache Ant.

Source Control: SVN, CVS, GitHub, GitLab

**PROFESSIONAL EXPERIENCE:**

**Amazon**

**Dallas, Onsite Oct 2022 to Present**

**Lead Software Developer**

**Responsibilities: -**

* Involved in various phases of the Software Development Life Cycle (SDLC) of the application.
* Administered and managed Unix/Linux servers for application deployment, ensuring optimal performance, security, and reliability of the systems.
* Implemented multi-threading concepts for concurrent execution and optimized application performance.
* Used most of the new features of Java8 like lambda expressions for communicating between the business layer and database, stream API for Bulk Data Operations on Collections using both streams and parallel streams.
* Having good implementation experience with installation and configuration of Kubernetes in Unix/Linux systems, clustering them, and managing local deployments in Kubernetes.
* Built a real-time pipeline for streaming data using Kafka and Spark Streaming.
* Developed secure APIs using Java, Spring Boot, and REST, incorporating OAuth2 for authentication and authorization, and ensuring data protection through encryption.
* Implemented data encryption at rest and in transit using AWS KMS (Key Management Service) and SSL/TLS protocols to safeguard sensitive information.
* Managed user authentication and access control using AWS IAM roles and policies, ensuring secure and granular permissions across services.
* Implemented generating the verification hash code using the crypto package on Node.JS and Added user validations and user permissions by using Node.js.
* Developed an application to assess JSON and XML from Restful web service from the consumer side using Angular 8/9.
* Developed complex reactive applications with NgRx, leveraging RxJS observables for handling asynchronous data streams, integrating them with the Angular Router, and implementing selectors to efficiently derive and query data from the store.
* Utilized NgRx to manage application state in Angular 8/9, ensuring predictable state transitions and enhancing the testability of the codebase.
* Implemented NgRx effects to handle side effects, including API calls and other asynchronous operations, providing a clean separation of concerns within the application.
* Developed and maintained new libraries with Microservices architecture using Rest APIs, and Spring Boot.
* Transformed legacy application into a suite of cloud-hosted Microservices using Spring Boot and laid the groundwork for x10 traffic scale.
* Implemented logging across microservices using Log4j and SLF4J, ensuring consistent and comprehensive logging throughout the application.
* Conducted garbage collection (GC) tuning by analyzing GC logs to reduce latency and pause times, optimizing application performance under high load.
* Configured JVM heap sizes based on application memory usage patterns and profiling results, preventing out-of-memory errors and improving stability.
* Monitored JVM performance using AWS CloudWatch, focusing on GC behaviour and heap memory usage to ensure efficient resource management.
* Migrating applications to AWS and application deployment in the cloud (AWS) with CI/CD tools like AWS codecommit, CodeBuild, codepipeline and Maven .
* Managed EC2 administration and utilized AWS CLI for data loading from S3, implementing daily scheduling tasks.
* Developed API for using AWS Lambda to manage the Serverless architecture and run the code in AWS.
* Managed Amazon Web Services (AWS) like EC2, S3 bucket, Lambda, Glue, ELB, Auto-Scaling and IAM through AWS Console and API Integration.
* Implemented Angular8/9 Router to enable navigation from one view to another.
* Developed Python-based Lambda functions for serverless computing, enabling scalable and cost effective execution of code on AWS.
* Integrated Python with AWS Step Functions to orchestrate complex workflows and coordinate microservices within the application architecture.
* Implemented modular approaches for the backend on AWS and implemented interfacing with Postgres SQL.
* Involved in design and development in the process of converting legacy applications into Microservices.
* Worked on Kubernetes, Configuring the Kubernetes pods and testing the pods.
* Used Continuous Delivery / Continuous Integration (CD/CI) tools Docker, and Jenkins to deploy this application to AWS.
* Developed Microservices with Spring Eureka to retrieve API routes for the entire cluster. Using this strategy each spring service in a cluster can be load balanced and exposed through one API gateway.
* Designed and implemented robust messaging solutions using AWS Simple Notification Service (SNS) and Simple Queue Service (SQS), significantly enhancing the reliability and scalability of our video streaming services.
* Created reusable Terraform modules to standardize infrastructure provisioning across multiple environments.
* Leveraged Terraform's remote state management with S3 and state locking with DynamoDB to ensure consistency in collaborative environments
* Automated the creation and management of VPCs, subnets, route tables, and security groups using Terraform.
* Integrated Terraform with AWS Lambda to automate the scaling of serverless applications.
* Managed infrastructure as code (IaC) to version control AWS resources, ensuring traceability and rollback capabilities.
* Designed and developed the application using frameworkAngular8/9 with HTML5, CSS3, SASS, jQuery, AJAX, and Typescript.
* Apache Kafka And EMR services to send and explore message/clickstream objects to client queues and topics.
* Integrated Python with AWS S3 for object storage and processing, enabling efficient handling of large volumes of marketing data within the application ecosystem.
* Utilized Python for data processing and analysis tasks on AWS, leveraging services like AWS Glue for ETL (Extract, Transform, Load) jobs and Amazon Redshift, DynamoDB for data warehousing.
* Leveraged Scala for interacting with Apache Kafka, developing robust message producers and consumers to handle message/clickstream objects efficiently.
* Integrated JMS with AWS services, such as S3 and Lambda, to facilitate file storage and processing, enabling seamless data workflows in cloud environments.
* Integrated Scala applications with Amazon EMR services for large-scale data processing, utilizing Scala's concurrency features for parallel execution and resource optimization.
* Analysed the Node.JS server structure in the legacy project, then mimicked the REST service by using Java JAX-WS API and did the corresponding configurations.
* Experience in Docker Engine and Docker Machine environment, to deploy micro services-oriented environments for scalable applications.
* Designed and developed secure data structures in Java to handle large-scale data processing efficiently while maintaining data integrity and confidentiality.
* Applied secure coding principles in the development of microservices using Spring Boot, ensuring data validation, input sanitization, and protection against common vulnerabilities like SQL injection and XSS.
* Performed optimization of Elasticsearch search queries using Kibana and implemented cache to minimize load and latency.
* Developed various generic JavaScript and Angular functions to implement client-side validations.
* Implemented Python-based monitoring solutions using AWS CloudWatch, CloudFormation, cloudfront and custom metrics, ensuring proactive identification and resolution of performance issues.
* Configured Redis cluster for high availability and scalability, ensuring uninterrupted caching services for mission-critical applications.
* Implemented secure data storage and processing on AWS, utilizing services like Amazon S3 with server-side encryption (SSE) and AWS Glue with encrypted data processing pipelines.
* Developed and deployed containerized applications using AWS Fargate, ensuring efficient resource management and scalability without the need to manage underlying infrastructure.
* Leveraged AWS ECS (Elastic Container Service) to orchestrate and manage containerized applications, ensuring high availability and fault tolerance.
* Scaled elastic search clusters and ensured proper resource allocations to clusters by implementing monitoring tools and alarms.
* Experience with Responsive Web Design, Bootstrap framework.
* Experience with tools like Experience with Docker Images/containers and Kubernetes.
* Pushed the image to the Docker Trusted Registry (DTR) is the enterprise-grade image storage solution from Docker.
* Developed internal application using Angular 8/9 and Node.js connecting to MongoDB on the Backend.
* Implemented Amazon DynamoDB and MongoDB in tandem, utilizing MongoDB for complex query capabilities and DynamoDB for high-availability, low-latency access to critical data, ensuring robust data handling across diverse application needs.
* Developed backend logic using MongoDB for dynamic, schema-less storage, allowing flexible data structures and efficient query handling for various microservices, particularly in applications requiring rapid iteration.
* Integrated Amazon DynamoDB for fast and scalable data storage, enabling efficient access and management of large datasets within the application environment, with optimized read and write capacities for high-demand services.
* Involved in database design and data migration from the existing database as per the microservices requirement.
* Experience in Updating, Modifying queries in Oracle. Developed stored Procedures, Functions, Packages, and SQL Scripts using PL/SQL.
* Monitored and reported Agile metrics such as velocity, burn-down charts, and cycle times to assess team performance and drive data-driven decisions for process improvements.
* Actively participated in sprint planning sessions and retrospectives, collaborating with team members to define user stories, estimate tasks, and identify areas for continuous improvement in the development process.
* Facilitated daily stand-up meetings to ensure alignment within the team, address blockers, and promote transparency on project progress, enabling timely adjustments to meet deadlines.
* Collaborated effectively with cross-functional Scrum teams to deliver high-quality software solutions, adapting to changing requirements and ensuring alignment with business objectives.
* Worked closely with Product Owners and stakeholders to gather requirements and translate them into actionable user stories, ensuring the delivery of high-quality features that meet user needs.
* Advocated for Agile best practices and principles within the team and the broader organization, promoting a culture of continuous learning and adaptability.

**Environment**: Java/J2EE, Python,Angular 6,7,8,9, HTML5, CSS3, NgRx, Spring Boot, Spring Cloud, Spring AOP, Hibernate, REST, Spring Boot, HQL, SQL, Kafka, JMS, Oracle, JPA, JUnit, Mockito, Maven, Jenkins, JIRA, Tomcat, GitLab, GitHub, Agile, Scrum, Redux, JSX, HTML, CSS, JavaScript, AJAX, JSON, XML and jQuery, Terraform, AWS, EC2,ECS, SNS,SQS Fargate, Lambda, cloudformation, Cloudfront , step, glue jobs etc, Linux.

**VMWare**

**Palo Alto, CA Jan 2021 - Oct 2022**

**Java Full Stack Developer**

**Responsibilities:**

* Developed technical documentation in Confluence and used Confluence as a knowledge base.
* Used the JIRA ticketing tool for tracking the issues and bugs.
* Developed webbooks using API to integrate the applications with Bitbucket.
* Applied Object-Oriented Programming (OOP) principles such as encapsulation, inheritance, and polymorphism to enhance code maintainability and scalability.
* Extensively used Core Java collections, Generics, Exception handling, and Design patterns for functionality, also worked on Java functional programming concepts and lambda expressions.
* Implemented the application using the concrete principles laid down by several Java/J2EE Design patterns like MVC, Singleton
* Implemented Spring MVC framework for developing J2EE-based web applications.
* Spring was used for its MVC features, dependency injection, and plug-ability with Hibernate.
* Leveraged design patterns such as Factory, Builder, and Adapter to solve common software design problems and improve code flexibility.
* Hands-on experience in working with Kubernetes for managing related, distributed cluster components.
* Designed Node.js application components through Express and NestJS frameworks.
* Used Bootstrap React.js and Node.js in effective web design.
* Create a AWS Lambda that watches an AWS S3 bucket, loads the data to Redshift, DynamoDB and sends a message using restful web services if the file has errors.
* Experience with container-based deployments using Docker, working with Docker images, Docker Hub, Docker registries, and Kubernetes.
* Deployed and managed applications on Pivotal Cloud Foundry (PCF), ensuring high availability, scalability, and streamlined continuous delivery.
* Integrated PCF with VMware infrastructure to optimize application deployment pipelines, leveraging PCF's platform-as-a-service capabilities for containerized environments.
* Worked in using React components, Forms, Events, Keys, Router, Animations, and Flux concepts.
* Developed CI/CD system with Jenkins on Kubernetes environment, Utilized Kubernetes, and Docker for the runtime environment for the CI/CD system to build, Test and Deploy.
* Integrated AWS CodePipeline to automate the CI/CD process, ensuring seamless transitions from code commits to deployments.
* Utilized AWS CodeBuild for building application artifacts and running automated tests, enhancing code quality and reducing deployment time.
* Managed source code in AWS CodeCommit, facilitating version control and collaboration among team members, resulting in improved code integrity and teamwork.
* Implemented multi-stage pipelines in CodePipeline for comprehensive testing, including unit, integration, and acceptance tests, ensuring high-quality releases.
* Configured build triggers in CodeCommit to initiate automated builds in CodeBuild upon code changes, streamlining the development workflow.
* Using Kafka for Publish/Subscribe pattern in applications dealing with messaging.
* Developed a dashboard based on Ajax calls with Business Customers Count, Flags, and Real-time graphs for Analytical Reporting using Node.js.
* Involved in writing JSPs, JavaScript, and Servlets to generate dynamic web pages and web content.
* Experienced in Docker engine and Docker Machine environments, to deploy microservices oriented environments for scalable applications. Excellent understanding of the OpenShift platform in managing Docker and Kubernetes.
* Developed responsive user interfaces using React for dynamic web applications.
* Implemented state management with React hooks for efficient data handling.
* Integrated third-party libraries and APIs seamlessly into React projects.
* Used Jenkins and pipelines to drive all microservices builds out to the Docker registry and then deployed to Kubernetes, Created Pods, and managed using Kubernetes.
* Used React JS for templating for faster compilation and developing reusable components.
* Developed readable and writable streams of Node JS to process the data from external means of the application.
* Experience in setting up the life cycle policies to back the data from AWS S3 to AWS Glacier, Worked with various AWS, EC2, Lambda, Glue, Redshift DynamoDB, Cloudformation, cloudfront and S3 CLI tools, API gateways.
* Used React JS to implement rich functions in various pages, form validation, crud, grid list, search, sort, multi-selection, authentication, and unit tests.
* Changing the AWS infrastructure Elastic Beanstalk to Docker with Kubernetes.
* Working on Web applications using Spring, Hibernate, apart from other standard J2EE technologies.
* Excellent in deploying the applications in AWS as EC2 instances and creating snapshots for the data that had to be stored in AWS S3.
* Used Hibernate to develop a DAO layer for performing CRUD operations on the database and configured it with Spring’s Hibernate DAO Support.
* Created Hibernate mapping files for mapping POJO to the database tables. Wrote named queries for querying the underlying database.
* Integrated Java Message Service (JMS) for reliable messaging and communication between distributed components.
* Implemented file and data transfer solutions using JMS to efficiently process large files and data streams between distributed systems, ensuring secure and reliable delivery.
* Configured JMS message brokers ( RabbitMQ) for reliable message delivery, ensuring high availability and fault tolerance in distributed systems.
* Developed message-driven beans (MDBs) to process JMS messages, enabling seamless integration with Java EE applications and improving overall responsiveness.
* Integrated JMS with Spring Framework to simplify messaging operations, leveraging Spring’s capabilities for dependency injection and configuration management.
* Developed and maintained error handling and retry mechanisms for JMS message processing, ensuring robust processing of messages even in the event of transient failures

**Environment:** Java, J2EE, React, Typescript, NodeJS, AngularJS, AWS, PCF, NGRX, Spring, Hibernate, Kafka, JMS, RabbitMQ, Oracle, SQL, HTML, CSS, JavaScript, React Js, Redux, JUnit, Jenkins, Agile, Web services, RAD, Tortoise SVN, IBM Web Sphere, Soap, Insomnia, Oracle SQL, Linux/ Unix scripting

**Nike**

**Bengaluru, India Jun 2016– Nov 2019**

**Software Engineer : (Jun2014 - May2016: Internship)**

**Responsibilities: -**

* Designed and developed various modules of the application with J2EE design architecture, and frameworks like Spring MVC architecture, AngularJS and Dropwizard.
* Designed and developed reusable components and services using AngularJS, transitioning to Angular 5+ for building dynamic, single-page applications (SPAs).
* Designed and developed Microservices business components using Spring Boot, Dropwizard for RESTful Web services.
* Developed reusable components and Services using AngularJS.
* Used Node JS to run Web pack tasks and build properly the project.
* Implemented modular approaches for the backend on AWS Elastic Beanstalk and interfaced with MySQL/Aurora DB on AWS’s RDS, allowing for scalability.
* Configured and maintained network and system settings on Unix/Linux servers, supporting the seamless operation of Docker and Kubernetes environments.
* Experience with Docker Containers, leveraging Linux Containers and AMIs to create Docker Images/containers and Kubernetes.
* Used Kubernetes to deploy scale, load balance, and manage Docker containers with multiple namespace versions.
* Wrote Kafka producers to stream the data from external est APIs to Kafka topics.
* Developed cross-browser/platform HTML5, CSS 3, and JavaScript to match design specs for complex page layouts while adhering to code standards.
* Developed Python scripts for data processing and automation tasks, leveraging libraries such as pandas and NumPy for efficient data manipulation.
* Implemented Python-based microservices alongside Java services, utilizing frameworks like Flask or FastAPI for rapid development and deployment.
* Utilized Python for web scraping and data extraction from external sources, enhancing data-driven decision-making within the application.
* Collaborated with cross-functional teams to integrate Python-based solutions into the existing architecture, driving innovation and efficiency across the development lifecycle.
* Documented every Defect that was worked on and saved in the Knowledge Center.
* Collaborated with marketing and design teams to translate business requirements into technical solutions, driving increased conversion rates and revenue growth.
* Experience working with Docker containers, Amazon EC2, S3, Lambda, Glue, Redshift, DynamoDB, ELBs, API Gateways etc in an AWS Environment.
* Developed rich UI components using AngularJS data binding, directives, and modular architecture, which improved code maintainability and enhanced user experience.
* Developed Node.JS readable and Writable streams to process the data from the external source of the application.
* Experienced in Docker engine and Docker Machine environments, to deploy microservices-oriented environments for scalable applications. Excellent understanding of the Open Shift platform in managing Docker and Kubernetes.
* Configuration file and Node JS used at the back end to interact with MongoDB using Mongoose connector.
* Designed and developed asynchronous RESTful Microservices using Spring Boot, Couchbase, and RX Java, integrated into AWS infrastructure (Elastic Beanstalk, RDS with MySQL/Aurora, EC2, and OpenShift for container orchestration), achieving an average response time of 30ms while maintaining scalability and reliability through Kubernetes and Docker.
* Optimized logging with Log4j, fine-tuned GC and JVM heap sizes, and utilized VisualVM MAT for performance and memory management, ensuring efficient resource use and application stability.
* Use Amazon Cloud Watch, CloudFormation and Cloudfront to customize and record logs to monitor the application.
* Used Docker for creating Docker images for launching containers on AWSEC2.
* Leveraged Docker and Kubernetes in OpenShift to deploy and manage microservices in containerized environments, enhancing scalability and load balancing capabilities across AWS cloud infrastructure.
* Collaborated cross-functionally to establish compliance and governance policies within the ESB architecture, implementing caching and routing optimizations that reduced message processing times by 30% and improved system responsiveness.
* Developed RESTful Web services client to consume JSON messages using Spring Boot configuration.
* Used validators, converters, jQuery, and JavaScript inbuilt functions in the application for validation.
* Implemented DAO for data access using Spring MVC, Spring Boot, Hibernate and Dropwizard.
* Implemented REST Microservices using spring boot and Dropwizard. Generated Metrics with method-level granularity and Persistence using Spring AOP and Spring Actuator.
* Developed database persistence layer components for mapping an object-oriented domain model to the relational database using ORM frameworks such as Spring JDBC Support, Hibernate, and JPA.
* Developed socket programming solutions for file transfer, enabling direct communication between client and server applications for high-speed data exchange.
* Analyzed and optimized ESB performance by implementing caching strategies and fine-tuning routing rules, leading to a 30% reduction in message processing times and enhancing overall system responsiveness.
* Worked with cross-functional teams to establish governance policies and compliance measures within the ESB architecture, ensuring secure data exchanges and adherence to industry regulations across integrated services.
* Actively participated in Agile methodologies, including daily stand-ups, sprint planning, and retrospectives, fostering collaboration and ensuring timely delivery of project milestones.

**Environment:** Python , Java 1.8, J2EE,AWS, Spring IOC, Spring MVC, Spring Boot, Spring JPA, SOAP Services and Restful Web Services using microservices architecture, , NodeJS, AJAX, Spring JDBC, Swagger, HTML, CSS, jQuery, JavaScript, JSP, JMS, File/Data Transfer, Servlets, XML, Hibernate, Dropwizard, Jenkins, Apache Tomcat Server, WebSphere, AngularJS, JSON, Oracle DB, JBoss, Linux .

**Education**

Naveen Jindal School of Management, University of Texas Dallas

Master of Science - MS, Information technology and management

Jan 2020 - Dec 2021

 B-Tech : Computer Science engineering – GITAM university (2016)