**SRINIVAS REDDY SINGAREDDY**

**FULL STACK JAVA DEVELOPER**

**Email:** srinivas.singareddy321@gmail.com

**Phone: +1 4693051307**

|  |
| --- |
| **PROFESSIONAL SUMMARY** |

* Full Stack Java Developer with over 11+ years of experience, leveraging a robust proficiency in **Java 8/11**, **J2EE**, and utilizing frameworks like **Spring, Spring Boot, Spring MVC, Spring IOC, Spring Security**, and **Hibernate** for robust backend solutions.
* Expertise in front-end technologies including **React**, **Angular 4/5/6/8/9/11/13/16, Node.js**, and proficient in **HTML5, CSS3, Bootstrap, jQuery, AJAX, SASS, and SCSS.**
* Skilled in **TypeScript** and various **JavaScript** frameworks like **AngularJS**, **React.js**, **Redux**, **Next.js**, plus **Node.js** and **Express.js** for both client and server-side development.
* Proficient in **Java**, **C++**, **PL/SQL**, and **Python**, demonstrating expertise in building robust and scalable backend systems.
* Proficient in **Agile** and **Waterfall** methodologies, excelling across the **SDLC** with strong collaboration, communication, and time management skills, ensuring efficient and scalable code delivery aligned with business goals.
* Proficient in developing microservices and managing servers like **Apache Tomcat, WebLogic, JBoss,** and **WebSphere.**
* Skilled in database design and management with **Oracle, MS SQL, MySQL**, **PostgreSQL** leveraging **NoSQL** databases like **MongoDB, DynamoDB and Cassandra**.
* Proficient in project management and documentation using **JIRA, Confluence, Swagger**, and **Grafana**.
* Extensive experience with cloud platforms, including **AWS** (EC2, ECR, ECS, IAM, EKS, RDS, Lambda, S3, CloudWatch, SQS, Glue, Elastic Beanstalk), **Azure** and **PCF.**
* Proficient in managing and streaming messaging systems with a focus on event-driven architecture, using tools like **Apache Kafka, Active MQ, RabbitMQ,** and **Spring JMS**.
* Implemented robust security protocols and features like **LDAP**, **OAuth 2.0**, **Spring Security** and **JWT** for token-based authentication, ensuring the safeguarding of applications and sensitive data.
* Hands-on experience with big data technologies such as **Apache Spark**, skilled in data visualization with **Kibana**, providing actionable insights for strategic decision-making.
* Managed **Kubernetes** for optimized scalable container deployments and high availability.
* Utilized **Terraform** for infrastructure as code, streamlining resource provisioning.
* Skilled in managing serverless **APIs** with API Gateway, integrating backend services using JAX-RS (**REST**) and JAX-WS (**SOAP**) for web service development, and proficient in connecting them with front-end applications.
* Expertise in **GraphQL** for efficient data queries, **Graph DB** for complex data relationships, and **Solr** for high-performance search functionality.
* Demonstrated expertise in **Scala** programming and leveraging the **Spark** framework to develop scalable and efficient data processing and analytics solutions.
* Designed and implemented authentication solutions using **OpenID** **Connect**, enabling secure and streamlined user authentication for web and mobile applications.
* Designed and developed dynamic SPAs using the **MEAN** stack toolkit.
* Strong expertise in **J2EE** testing, proficient in developing and executing test cases with **JUnit, Mockito, Jasmine, Karma, Jest, Enzyme, Selenium** and **Cucumber**, adhering to **TDD and BDD** principles.
* Experienced in **Agile** methodologies, including **SCRUM** and **Kanban**, with advanced skills in logging using **Log4J** and **Splunk** for quality assurance in dynamic development environments.
* Skilled in software development and version control using **GIT**, **GitLab**, **GitHub** and **Bitbucket**, adept in **IntelliJ IDEA** and **Eclipse IDEs**, proficient in build automation with **Maven and Gradle.**
* Specialized in Continuous Integration and Continuous Deployment (**CI/CD**) using **Jenkins**, adept at constructing streamlined pipelines and effective application delivery through **Docker** container services.

|  |
| --- |
| **TECHNICAL SKILLS** |

|  |  |
| --- | --- |
| Languages | C, C++, Java 11/8, J2EE, JavaScript, Python 3.6.0, SQL |
| Web Technologies | jQuery, HTML, HTML5, CSS, CSS3, Bootstrap, TypeScript, AJAX, DOM, JSON, XML, SASS, SCSS |
| JavaScript Technologies | AngularJS, Angular 4/ 5/6/ 8/ 9/ 11/ 13/16, React.js, Next.js, Node.js, Express.js |
| Java Technologies | Spring, Spring Boot, Spring Cloud, Spring MVC, Spring IOC, Spring Security, Hibernate, Servlets, JDBC, JSP, JSTL, Swing, EJB, Scala, JAXB, Struts |
| Database/RDBMS and ORM | Oracle, MS SQL Server, MySQL, MongoDB, Redis, Cassandra, PL/ SQL, PostgreSQL, Graph DB Dynamo DB, BigQuery, Spark |
| Web Services | RESTful, SOAP, WSDL, GraphQL |
| Operating Systems | Windows, UNIX, LINUX, Mac OS. |
| Testing Frameworks | JUnit, Jasmine, Karma, Mockito, Enzyme, Jest, Log4j, Cucumber, Splunk, TDD, BDD, Postman, Selenium |
| Web Servers | Apache Tomcat, WebSphere, WebLogic, JBoss, Jetty, RBAC |
| Tools and Packages | GIT, IntelliJ IDE, Eclipse IDE, GitHub, Maven, Gradle, Jenkins, JIRA, Swagger, Grafana, Bitbucket, Docker, Kubernetes, Apache Camel, Kibana, Confluence, YAML, Helm charts, Solr. |
| Cloud | AWS, Azure, OpenShift, PCF. |
| Messaging | JMS MQ, Rabbit MQ, Apache Kafka, Active MQ, Amazon SQS |

|  |
| --- |
| **PROFESSIONAL EXPERIENCE** |

**Java Full Stack Developer**

**Client: Bank of America, Charlotte, NC Jul 2022 – Present**

**Key Contributions:**

* Developed dynamic, responsive user interfaces with **Angular 16**, TypeScript, Bootstrap, HTML and CSS focusing on high-quality, maintainable code in main application.
* Responsible for the development of Single Page Applications (SPAs) using **Angular** frameworks to improve the productivity of the internal team.
* **Point of contact for multiple PODs** (Ordering, Address, and Payments) and used to handle **end-to-end tasks**.
* Designed web pages adhering to **WCAG** standards using **Angular 16** accessibility features.
* Experience in using **React** **JS** for web development and used **axios** and **redux** **saga** for making calls (GET, POST, PUT, DELETE) for triggering or fetching data from **Restful** **API’s** in reporting tool application.
* Utilized **NgRx** and **RxJs** for state management, aligning with **Agile** and **Kanban** methodologies for rapid, responsive development.
* Enhanced data stream processing in microservices using **Java 8** features like Streams, Collectors, and lambda expressions, streamlining code, and improving backend efficiency.
* Leveraged **Java 8**'s new Date and Time API for precise and efficient handling of time-related operations in application logic.
* Enhanced performance with **Java 11**'s improved garbage collection and HTTP Client API; Applied new lambda and switch expression syntax for readability.
* Streamline code using **Java 11** features like var in lambdas, enhanced string handling for efficiency and improved readability.
* Implemented **Multithreading**, Concurrency, Exception Handling and Collection whenever necessary.
* Used several **J2EE** **Design** Patterns (Session Façade, Aggregate Entity) for the Middle Tier development.
* Worked on developing **UML** Use case diagrams, **Class diagrams**, and **Sequence diagrams** using Visio tool.
* Designed robust **RESTful APIs** using **Spring MVC** and **JAX-RS** within a microservices architecture, ensuring seamless front-end communication and durable endpoints.
* Converted a **monolithic** **app** to **microservices** architecture using **Spring** **Boot** using **12-factor app** methodology.
* **SOAP-UI** was used extensively for testing webservices across all environments.
* Extensively participated in working with **Spring framework.** Involved in writing **JSP** and **Servlets**.
* Leveraged **Spring** **Cloud** for distributed systems, configuration management, and service discovery in microservices.
* Customized query methods in **Spring Data** repositories, using query derivation mechanisms to create complex queries from method names, improving readability and development speed.
* Utilized **Spring Data REST** to automatically expose repository methods as **RESTful** endpoints, facilitating rapid development of data-driven APIs with minimal configuration.
* Configured and managed **OAuth** **2.0** and **OpenID Connect** flows, including authorization code flow, implicit flow, and hybrid flow, to meet diverse authentication requirements.
* Utilized **AJAX** with **JSON** and **XML** for efficient, asynchronous data exchange in web services.
* Written Stored Procedures using **PL/SQL** and functions and procedure for common utilities using **Oracle11g**.
* Integrated **Oracle** databases with Java back-ends using **JDBC** and **JPA/Hibernate** for seamless data communication and **CRUD** operations, Integrated with API Gateway for scalable **RESTfu**l APIs.
* Developed **DAO** layer using **Cassandra** and expertise in **CQL** (Cassandra Query Language), for retrieving the data present in **Cassandra cluster** by running queries in **CQL**.
* Created application to create control files for performing seed data migration from **DB2** to **Cassandra** called **init** load. Data files are created by Mainframe system. Various formats of data files are Fixed Width and delimited.
* Tracking and managing project issues using **JIRA**, leveraging its **agile** board and custom workflows.
* Optimized **Spring** **Batch** for data handling in microservices and **CI/CD** with chunk processing and retry/skip logic.
* Designed and implemented **Apache** **Camel** routes for simplifying integration patterns, enabling easy and flexible routing of messages across different endpoints.
* Created multiple **Kubernetes** Clusters in **AWS** and deploying a **Docker** Image on top of Kubernetes Cluster for testing purpose.
* Developed **Terraform** scripts to automate infrastructure provisioning, configuration, and deployment in **AWS**, resulting in a 50% reduction in deployment time.
* Optimized **Kafka** **cluster** performance by monitoring and tuning **Kafka** configurations and server resources.
* Integrated **Kafka** with real-time data processing frameworks such as **Apache Flink** and **Spark** Streaming.
* Developed error handling and message re-routing strategies using **Spring Integration**, ensuring robustness and resilience in message processing across the application workflow.
* Applied **Spring Integration's** support for Enterprise Integration Patterns (EIPs) to solve common integration problems, such as message filtering, transformation, and aggregation, streamlining data flow.
* Overseen **CI/CD** pipelines using Jenkins for automation along with **LINUX** Shell scripts to automate routine jobs, **Maven** for dependency management, and **Git, GitLab** for version control.
* Developed custom **GitHub** **Actions** workflows for multi-environment deployments, allowing for differentiated deployment strategies across development, testing, and production environments.
* Implement **Groovy scripts** in Jenkins pipelines to automate and enhance **CI/CD** processes, particularly for managing and deploying Java-based applications.
* Conducted system monitoring and performance tuning using **Elasticsearch** and **Kibana**, focusing on high application performance and availability.
* Implemented robust security measures with **Spring Security** using **OAuth 2.0** for authorization and **LDAP** for authentication.
* Managed **AWS EC2** for scalable web applications and **S3** for secure storage, integrating **AWS Lambda** with API Gateway for serverless **Python** and **Node.js** scripts.
* Deployed microservices efficiently using **AWS ECS** with Fargate and orchestrate containers with **AWS EKS** for automated, secure management.
* Utilized **AWS IAM** for robust access control and user management; implemented **AWS RDS** for database management.
* Adopted **GraphQL** as the query language through **the apollo-server and graphql-yoga** for building flexible and efficient APIs, allowing clients to request exactly the data they need.
* Implemented asynchronous processing patterns with the **Node.js async/await** syntax and used the **graphql-yoga** server for a powerful and easy-to-setup **GraphQL** server.
* Enhanced **Solr** for full-text search, indexing, and scalable **Solr Cloud** in Java, aligning **SolrJ** with server.
* Using **Redis** cache in **Microservices** to reduce server overload. Designing several DATA pipelines using **Kafka**.
* Tested APIs with **Postman** and ensured code quality with **Sonar Lint**, Documented APIs using **Swagger**, ensuring clear and concise API documentation.
* Developed robust unit and integration tests using **JUnit**, **Mockito**, and **Selenium**, adhering to Test Driven Development (**TDD**) principles for comprehensive end-to-end testing.
* Developed Angular unit tests with **Jasmine** and **Karma**, used **log4j** for effective logging and tracing messages.
* Managed application server environments with **Apache** **Tomcat**, focusing on deployment, performance optimization, and ensuring system reliability.

**Full Stack Developer**

**Client: AbbVie, Chicago, IL Oct 2020 – Jun 2022**

**Key Contributions:**

* Applying **Agile** methodologies and **Scrum** for efficient project management.
* Creating dynamic user interfaces using **ReactJS** for its **virtual DOM** feature, **JavaScript** for client-side scripting, and **HTML/XML and CSS** for webpage structure and style.
* Developed Ul using **ReactJS**, **Redux**, **NextJS** and used **Redux**-**Promise** in application to retrieve data from Back-End.
* Used **Grunt** build to build **JavaScript**, jQuery, CSS3, HTML5 files that embeds bootstrap, NodeJs bower components.
* Implemented **Redux** for state management and **Redux Saga** for asynchronous **API** calls, including middleware integration for **RESTful** services and side effects management in React applications.
* Managed data exchange and **API** interactions using **AJAX, JSON, XML**, ensuring robust communication between frontend and backend.
* Managed data serialization and deserialization using **JAXB** for **XML** binding, and **JSON** for data representation.
* Enhanced functionality using **Java 8**, focusing on Lambda Expressions and Streams; plan migration to **Java 11** for advanced features such as the improved ZGC (Z Garbage Collector).
* Utilized **Java 11**'s HTTP Client API to build robust and responsive applications.
* Optimized multi-threading for **Core Java** applications and minimized memory leaks in Core Java code.
* Built on-premise data pipelines using **Kafka** and **Spark** streaming using the feed from API streaming Gateway **REST** service.
* Build efficient **Kafka** producers and consumers for reliable messaging in distributed systems.
* Apply **Spring Framework** in **MVC** for modular and loosely coupled backend development.
* Designed and configured message channels and adapters in **Spring Integration** to connect with external systems and services, such as databases and messaging systems, enhancing integration capabilities.
* Implemented security best practices in **OpenID** **Connect**, such as token validation, secure storage of tokens, and protection against common vulnerabilities like token replay attacks.
* Optimized data storage using **PostgreSQL** for relational databases and management with **Hibernate ORM** and **JPA** for efficient data access and transaction handling.
* Leveraged **Spring Boot** **4.x** for microservices, emphasizing auto-configuration and rapid development.
* Integrated **Spring Data** with Spring Cache, optimizing application performance by caching query results and reducing the number of database calls for frequently requested data.
  + Implemented **JWT** with **Spring Security** for secure authentication and integrate Spring Boot Actuator for enhanced security and management of microservices.
* Worked with **Spring MVC** Framework to Design and Develop the **REST APIS**.
* Created advanced web applications with **JSP** and **JSF**, incorporating **AJAX** for real-time user interactions and data updates.
* Developed a **CI/CD** system using Jenkins within a **Kubernetes** container environment, leveraging **Docker** for building, testing, and deploying applications.
* Led the integration of **AWS** **Lambda** to create a seamless, serverless architecture, enhancing the efficiency of data processing and risk assessment modules.
* Seamlessly integrated **AWS ECS** and **S3** services with **Node.js** applications, optimizing data flow and improving overall system efficiency.
* Leverage **AWS Glue** for serverless data integration and ETL, streamlining data transformation and analytics processes.
* Configured **AWS** **CloudWatch** Logs for centralized logging, setting up log streams and retention policies to manage logs from **EC2**, **Lambda**, and other services efficiently.
* Utilized **GraphQL** for efficient and flexible data queries, enhancing client-server interactions and data retrieval processes.
* Used **Maven** and **GIT** for project builds and version control, maintaining code integrity and facilitating collaboration.
* Developed custom **Splunk** data collectors for Java applications, enabling the capture of critical application metrics, logs, and events. This allowed for proactive monitoring and quicker issue resolution.
* Developed and executed unit tests using **Jest**, **Enzyme**, and **Selenium**, adhering to **TDD** methodologies.
* Integrated **GitHub** **Actions** with **Docker**, automating the creation, testing, and deployment of Docker containers, facilitating consistent and reliable application delivery.
* Simplify complex **JavaScript** with **jQuery** to enhance frontend development efficiency and user interface.
* Configured and monitored numerous **MongoDB** and **NoSQL** instances. **MongoDB** Schema Design using DB Ref, Manual Ref, Embedded Data Model Design.
* Performed **CRUD** Operations like Update, Insert and Delete data in **MongoDB** and handled database access and data transmission based on **RESTful** Web Service.
* Utilized **Apache Camel's** extensive component library for connecting a wide range of transport protocols and APIs, enhancing the application's integration capabilities.
* Designed and configured management reports and dashboards using **Kibana**.
* Managing project issues with **JIRA** and documenting processes in **Confluence** for seamless collaboration.
* Implement Role-Based Access Control (**RBAC**) for secure system access and manage application configurations with **YAML** and **Helm** charts.
* Integrated **Redis**/**Memcached** for caching to enhance application performance.
* Implemented **ELK** stack by integrating **Kibana** with **Elasticsearch** and **Logstash** for comprehensive log management.
* Utilized **Spark** for large-scale data processing and analytics, employing **BigQuery** for efficient handling of extensive datasets.
* Process large-scale data with **Scala** and **Spark**, utilizing **Spark SQL** for structured data querying and **GraphX** for graph data analysis.
* Managed application server environments with **WebLogic**, optimizing for performance and reliability.

**Full Stack Developer**

**Client: State of Nebraska, Lincoln, NE Apr 2019 – Sep 2020**

**Key Contributions:**

* Developed scalable applications by creating **TypeScript** classes and services in **Angular 5/6** to interface with **RESTful** APIs, leveraging Angular’s component-based design for enhanced scalability and maintainability.
* Transitioned application development from **React Native** and **JavaScript** to **Angular**, focusing on exploiting Angular's service workers to enable robust offline functionality for a device logger library.
* Mastered the development of responsive web applications using Angular, transitioning from **React/Redux** to Angular's HttpClient for REST API consumption, and adopting **RxJS** for sophisticated reactive programming patterns.
* Optimized code using **Java 8's** concurrency features such as CompletableFuture for asynchronous programming, enhancing application responsiveness and scalability.
* Employed **Java 8’s** Interface Default and Static Methods to create more flexible and easily maintainable code structures, promoting better API design.
  + Implemented **Bean** classes and configured in spring configuration file for **Dependency Injection, Developed Controller Classes using Spring MVC, Spring AOP, Spring Boot, Spring Batch modules, handled security using Spring Security.**
* Upgraded to **Spring Boot 3.x** for new features and built microservices with Spring Cloud.
  + Build rich user interfaces with the **JSF** Framework, Using **Hibernate** to communicate with **PostgreSQL** database and queries using HQL.

• Tested web services using **SoapUI**, validating **SOAP** based web services and their responses.

* Implemented **SOAP** web services using **JAX-WS** for interoperable machine-to-machine communication,

• Handled data using **MongoDB** for high-performance and scalability, focusing on efficient data management.

* Optimized data retrieval with **MongoDB** indexing and configured cloud clusters with **MongoDB Atlas.**
  + Write and maintain unit tests with **Junit** using **TDD** principles, ensuring code reliability and robustness.
  + Used **Azure** DevOps for automated **CI/CD**, Implement event-driven microservices with **Azure** Functions.
  + Utilized **Azure App Service** and Functions for scalable hosting and event-driven computing.
  + Automated deployment of secrets and certificates from **Azure KV** into Azure Kubernetes Service (**AKS**), securing containerized applications.
  + Integrated **Azure** **KV** with **Azure** **AD** for identity-based access control, ensuring secure and selective access to sensitive information based on roles.
  + Optimized data storage and retrieval processes with **Azure** **Blob** and File Storage, ensuring high availability and secure access to application data.
  + Utilized **Azure's** support for **PostgreSQL** to manage and scale relational databases seamlessly, ensuring efficient data storage and high availability.
  + Utilized **Docker** for containerization, and **Kubernetes** and **Terraform** for orchestrating and managing containerized applications.
  + Utilized **RabbitMQ** for efficient message brokering across distributed microservices, ensuring scalability and reliability.

• Used **Log4J** for logging and tracing messages. Used **JBPM** to manage workflow and schedule Jobs.

• Managed project workflows and issue tracking with **JIRA**, streamlining development processes.

• Used **Git** for version control and **Gradle** for build automation, enhancing consistency and efficiency.

• Designed and optimized **MongoDB** database schemas and queries for efficient data storage and retrieval.

* Used **Graph DB** for complex data relationships, ensuring efficient data processing and retrieval.
* Integrated **Spring Security** with **JWT** for robust protection against vulnerabilities and enhance code quality through Spring Boot testing.
  + Implemented caching strategies using **Redis** and optimized server-side performance with **NodeJS** for efficient data operations.
* Apply **Selenium**, **Jasmine**, and **Karma** for comprehensive front-end and back-end automated testing, ensuring application reliability.
* Leveraged **Apache Tomcat** for server management, focusing on deployment and environment handling.

**Full Stack Developer**

**Client: Panasonic Avionics Corporation, Lake Forest, CA Sep 2017 – Mar 2019**

**Key Contributions:**

* Implemented web application request-response handling using **Servlets**, enhancing server-side processing and seamless frontend integration.
* Utilized **XML** for configurations, promoting structured data management, ease of integration, and maintainability.
* Develop, consume, and maintain **SOAP**-based web services with **WSDL** to ensure interoperability and seamless data exchange.
* Involved in working with **MySQL**, **Oracle** databases and Proficiency in writing **SQL** Queries and stored procedures.
* Deploy and manage applications on **Apache** **Tomcat**, optimizing for performance and scalability., and employ **Maven** for build automation and dependency management.
* Created web services using **JAX-WS** for interoperability and API design and optimize **Oracle** performance with **PL/SQL** scripts and **Hibernate ORM**.
* Configured **Apache Server** for web hosting, focusing on performance, security, scalability, and utilize **Struts** for **MVC**-based web application development.
* Applied **DAO** patterns for database interactions and **AOP** in Spring for efficient logging and transaction management, utilizing **AWS** for scalable cloud infrastructure and services.
* Implemented **Spring Batch** for robust batch processing, optimizing data handling and task automation in backend operations.; maintain unit tests with **JUnit**, following **TDD** principles.
* Designed **XML** schemas for standardized data structures, integrate **Hibernate** for database interactions, and implement scalable microservices with **Spring Boot** for rapid development and cloud adaptability.
* Enhanced message handling with **RabbitMQ** in microservices architecture, ensuring rapid development, maintainability, and effective communication.

**Full Stack Developer**

**Client: Golden Living Canters, Canton, MA Dec 2015 – Aug 2017**

**Key Contributions:**

* Used WSDL and SOAP for the development of XML based applications.
* Used JAXB for XML marshaling and unmarshalling.
* Involved in writing complex SQL queries and developed Stored Procedures to populate the database.
* Used ANT build tool for compiling and generating war files.
* Developed SCM by using the JSP/HTML for each user interface. Servlets standard validations are used as the controllers for business logic and using JDBC, XML parsing techniques using MVC.
* Deployed application on the Client Linux server.
* Used Multithreading concept to post data from files to tables involving the transformation of files from one format to another.
* Worked with Log4J to capture the log that includes runtime exceptions.
* Performed unit testing using JUnit and participated in integration testing.
* Configured Azure Virtual Networks (VNets), Network Security Groups (NSGs), and Azure Load Balancer for optimizing network connectivity, traffic routing, and security for Java applications running on Azure cloud infrastructure.
* Kept up the bean life cycle by utilizing Spring IOC, which keeps session information about beans.
* Configured Struts-config.xml, Tiles-def.xml, and various properties files provided by Struts Framework for the implementation module.
* Utilized various utilities for User Interfaces like JSP, JSTL, JavaScript, HTML, CSS, PHP, jQuery, and Ajax.
* Experience in installing, configuring, and deploying applications on Web Logic Server.

**Environment**: Core Java, J2EE, JSP, HTML, MVC, Servlets, Struts, JMS, MDB, EJB3.1, JUnit, PL/SQL, DAO, Web Logic Server, Ant, SOAP, JAX-WS, WSDL, Junit, SQL.

**Java Developer**

**Client** **GDIT, Rensselaer, NY Oct 2013 – Nov 2015**

**Key Contributions:**

* Developed **Java/J2EE** applications focusing on backend logic, data processing, and system integration, utilizing **Core Java** for efficiency, and integrating scalable **microservices** with **Spring Boot**.
* Worked on enhancements with the existing application, which was implemented using **MVC** paradigm, implemented by **Spring Framework**.
* Developed and consume **RESTful** web services for secure and scalable data exchange between systems.
* Involved in the development of user interface applications and professional web applications using **HTML, CSS3, JavaScript, jQuery, Ajax, Bootstrap.**
* Designed and implemented database schemas in **PostgreSQL**, optimizing tables and relationships for performance and scalability in web application development.
* Micro services development using spring boot and deployment in **Pivotal Cloud Foundry (PCF).**
* Experienced with Middleware Technology Messaging Brokers like **JMS, Active MQ** in **SOA** (Service Oriented Architecture).
* Managing project issues with **JIRA** and documenting processes in **Confluence** for seamless collaboration.
* Implement **Selenium** for automated testing of web applications, enhancing reliability and robustness.
* Manage data with **DynamoDB**, focusing on performance and scalability for handling unstructured data.
* Conduct performance testing and follow **TDD** principles for web applications using **Apache JMeter.**

**Environment**: Java, J2EE, Git, Spring Framework, Core Java, REST Web Services, JavaScript, HTML5, CSS, SASS, JIRA, Confluence

**EDUCATION**

* Bachelors in Computer Science from Bharath University (2012)