**Mallikarjun Y**

**Lead Full Stack Java Developer | +1 313-455-3624**

[gprakash@provincesoft.com](mailto:gprakash@provincesoft.com) | https://www.linkedin.com/in/mallikarjun-y-42b396186/

**Summary:-**

* **10 years** of experience in designing, developing, and testing Enterprise Applications using **Java/ J2EE** platform, **object-oriented methodologies**, Web services like **SOAP**, **REST** and **SDLC** approaches for software development in Banking, insurance, healthcare, e-commerce domain.
* Extensive knowledge in developing applications using corejava concepts like **oops, Collections, Concurrency, Data Structures,** **EAI Patterns, Exception Handling, multi-threading,** and very good knowledge in **Advanced Java,** **Java11, EJB** used editions like **J2EE,** **Java SE.**
* Used **Reactive** **programming** libraries like **RxJava**, **Project** **Reactor**, or similar technologies within Spring Boot and Microservices for managing asynchronous I/O and handling data streams.
* Good experience with skillful ability in using **Spring** and **Spring Boot** framework modules like **Spring Core (IoC), Spring MVC, Spring Security, Spring Batch,** **Spring Cloud,** **Spring Configuration,** **Spring integration,** **Spring Data JPA, JPA/Hibernate/JSF, ORM, Spring AOP, spring Transaction management.**
* Hands-on experience in developing and designing front end User Interface web applications and integration of highly transactional and scalable applications using **HTML**, **XHTML**, **DHTML,** **CSS**, **JavaScript**, **TypeScript, XML,** **XSLT**, **Angular JS, React JS, Vue.JS, Node.js, jQuery, Redux,** **Express.js,** **BackboneJS**, **Bootstrap,** **React Native, JSP,** **NPM,** **HTTP, SFTP, SAAS, LESS, DOM, Ajax.**
* Worked extensively with **Angular** using **Typescripts**, **routers**, **custom** **Pipes**, to achieve **MVVM** functionality and good experience in **AngularJS** concepts like **Dependency Injection**, **Promises, Services**, **Controllers** and creatingSinglePageApplications(**SPA**).
* Experienced in developing applications using **React JS** for view rendering and implementing **React JS** with **Redux** and **Flux** to achieve **MVC** design flow to the application.
* Good Experience in using Amazon Web Services (**AWS**) like **EC2 ,AWS Lambda, S3, AWS Aurora** and **Glacier** for storing data, **Amazon Elastic Load Balancing**,  **Amazon EBS**, **Amazon SQS, AWS ECS,** **AWS EKS, AWS MSK,** Pivotal Cloud Foundry **(PCF), CA-Lisa ,VPC, Route53, IAM** (Identity and Access Management)**,** AWS Code Build, AWS Code Commit,AWS Code Pipeline**, AWS CLI,** AWS CloudFormation**, API Gateway, SNS/SES/SQS, AWS WAF, Kinesis**, AWS Event Bridge, AWS CloudTrail**,** AWS DevSecOps**.**
* Experience in Building a highly scalable, highly available, public cloud infrastructure using **Azure services** like **App services**, **Azure Storage**, **Azure SQL database**, **Virtual machines**, **IIS**, **Azure Active Directory** (**AD**), **Azure Resource Manager (ARM)**, **Cosmos DB**, **AZURE Functions**, **Azure Service Fabric**, **Azure Monitor**, **AZURE Service Bus**, AZURE Virtual Networks (**VNets**), subnets, security policies and routing, **Azure Blob Storage, Azure API Management (APIM), Azure Event Hub, Azure Kubernetes Service (AKS).**
* Experience with **Jenkins,** **Bamboo,** **Ant**, **Maven**, **GitHub Actions, Artifactory, Git/Bitbucket, GitHub, GitLab, Hudson, SonarQube, Nexus, AZURE DevOps, jfrog** to achieve Continuous integration (**CI**) and Continuous Development (**CD**)**.**
* Knowledge of **Kubernetes, OpenShift PaaS to** deploy containerized applications and Expertise in Application Deployments & Environment configuration using **Ansible**, **Chef**, **Puppet, Elastic Beanstalk, Elasticsearch.**
* Experienced with Relational databases like **MYSQL**, **SQL server, ORACLE**, **PostgreSQL,** **Aurora**, **DB2, Oracle PL/SQL,** **MariaDB,** **Elasticsearch, and** **NoSQL**Databases like **MongoDB,** **Cassandra, DynamoDB, Redis,** **Cosmos DB** and Query language for APIs like **GraphQL APIs** and **GraphQL** and used **Flyway** as Database version control and migration tool and used **ETL** design for extracting data from various sources.
* Expertise in implementing **ORM** technologies like **Hibernate**, **JPA**, **IBATIS** to connect with **RDBMS** and **Mongoose connector**, **Cassandra Node Driver** for No SQL database.
* Experienced in working with **Core Java 8** features like **Functional Interfaces, Lambda Expressions, Stream API, Optional class, Collectors, String Joiner, For Each (), Time API,**and **Generics.**
* Good Knowledge in **Jakarta EE**(**J2EE) technologies** like **Servlets**, Java Server Pages (**JSPs**), **JSF**, (**EJBs**), **JDBC**, **JNDI**, **JTA**, **JTS**, **JMS**,**JAF.**
* Expertise in developing **Micro services** using **Spring Boot,** **Spring Cloud**, **RESTful API, Spring MVC, Spring Batch, Spring** **JPA**, **ORM, Spring Core (IoC), Spring** **integration**, **Spring Config**, **GraphQL APIs** and **Node JS**to build more physically separated modular applications which will improve scalability, Availability, Agility of application. Developed Rest API’S using **Rest controller** and returned JSON formatted data.
* Experienced in working with patterns like **Service Registry**, **API Gateway**, **SAGA**, **CQRS** and **Circuit Breaker** for specific needs based on application requirements.
* Knowledge in using **Spring Cloud - Netflix OSS stack**for **microservices** deployment and auto configurations like **Eureka**which is a Service Discovery Server, **Ribbon** used for **Dynamic Routing and Load Balancer, Hystrix Circuit Breaker, Zuul (Edge proxy) and centralized log analyses (ELK).**
* Developed **RESTful webservices** using **JAX RS, Apache CXF, Jersey framework, WSDL**and **SOAP** services. Experienced in **SOA** and developing and integrating **SOAP** and **RESTful** web services using **Ajax** Calls
* Experienced in writing **Kafka** Stream applications using **Kafka** stream API implemented ingestion services to connect to RDBMS using Kafka producer API’S/Kafka connect. Experienced in working with Message Queues services like **Rabbit MQ**, **Apache Kafka**, **IBM MQ, JMS.**
* Good Knowledge on working with **OAUTH2.0**by integrating **Spring Security** to provide **authentication** and **authorization** to **Rest services.**
* Experience in using **JSON Web Tokens (**JWT) to securely communicate **JSON** Objects and used as access tokens in **OAuth** 2.0 to securely transmit information between parties in token-based authentication and their implementation in **RESTful** and **web services**.
* Expertise in developing front end web applications using client - side JavaScript frameworks like **AngularJS**, **Backbone**.**JS**, **Ember.JS**, **React JS**, **Vue.JS** , **Handlebar.JS, jQuery,** Express.js.
* Experienced using **Angular**.**js** Template driven and Model driven forms builder by using ng Model. Involved in designing the user experience interface UI/UX strategy. Experienced in working with **AngularJS MVC**, **MVVM** (Data Binding).
* Worked with **Angular**.**js** features like **Components**, **Services**, and **Routers** and created **Custom pipes** to format the data before displaying it to the user.
* Experience in implementing store actions, installing React Router, React - Redux app to handle real time data, API calls from **Thunk**, Redux in **ReactJS** by connecting backend **API's REST** with **JSON**.
* Experienced in React.js for creating interactive UI's using One-way data flow, **Virtual DOM**, **JSX**, **React Native** concepts and very good understanding in in using React.JS components, Forms, Events, Keys, Router, Animations and Flux concept.
* Good knowledge in Setting up databases in AWS using **RDS**, storage using an **S3** bucket and configuring instance backups to **S3** and Identifying and selecting services in AWS Cloud Infrastructure such as **IAAS, PAAS** and **SAAS.**
* Designed and managed public/private cloud infrastructures using Confidential Web Services (AWS) which include **EC2**, **S3**, **Cloud Front**, Elastic File System, **RDS**, **VPC**,**Route53**, **CloudWatch**, **Cloud Trail**, **Cloud Formation**, and **IAM** which allowed automated operations.
* Experienced in Pivotal Cloud Foundry (**PCF**) on **Azure** VM’s to manage the containers created by **PCF** and Used **Docker** for deployment containers and push the code to **Azure Cloud**.
* Worked in deployment and configuration of Application Servers such as **IBM WebSphere, BEA WebLogic, Apache Tomcat, JBOSS.**
* Proficient in testing Angular applications using **Karma**, **Jasmine**, **Protractor** and React Applications with **Mocha,** **Chai, Jest** and **Cypress.**
* Implementing unit testing using **Junit**, **Mockito** and **Spock** and performed functional testing using **Cucumber**, web application and API testing using **Selenium** **Grid**, **SOAPUI** and performance testing using **JMeter**, **JProbe** and API testing using **Postman**, **Swagger**.
* Expertise in working with build tools like **Maven**, **Gradle**, **ANT, Jenkins, Bamboo, GRUNT**, **GULP**, **Webpack.**
* Experience in working with**Log4J, Slf4j** fo**r** logging, debugging and error tracing, also used **Elasticsearch, Logstash, Kibana for** log management, collection, and analysis.
* Expertise in working with **Agile**, **Prototyping, Waterfall, scrum, XP, SDLC** and **RAD Methodologies**.
* Good Knowledge in Batch processing using **Quartz scheduler** and by writing **Shell scripts** and **Unix scripts.**
* Experienced in using version control and repositories like **Git**, **Bitbucket**, **GitHub**, **GitLab**, **GitHub Actions**, **Artifactory** and **Confluence** tools to integrate with version control tools to manage project documentation.

**Technical Skills:**

|  |  |
| --- | --- |
| **languages** | Java, JavaScript, TypeScript, Python, C, C++, AJAX, Shell Script, Reactive Programming. |
| **Java/J2EE Technologies** | SERVLETS, JSP, JDBC, JAVA BEANS, JSF, JNDI, EJB, JTA, JTS, JMS, JAF. |
| **Frameworks** | Spring, Spring Boot, Hibernate, Spring Core (IoC), Spring MVC, Spring Security, , JSF |
| **WEB SERVICES** | SOAP, REST, JAX-WS, WSDL, APACHE AXIS, IBM MQ, WSDL . |
| **Web servers** | Apache Tomcat, IBM WebSphere, WebLogic, JBoss, BEA WEBLOGIC, IBM WAS. |
| **Relational Databases** | MYSQL, SQL server, ORACLE, PostgreSQL, Aurora, DB2, Oracle PL/SQL, MariaDB. |
| **NoSQL Databases** | MongoDB, Cassandra, DynamoDB, Redis, Cosmos DB. |
| **WEB TECHNOLOGIES** | HTML, CSS, XML, XSD, WSDL, DTD, DOM/SAX, ReactJS, AngularJS, Node JS, jQuery. |
| **Cloud** | AWS, PCF, Azure, GCP. |
| **IDE** | Spring Tool Suite, Eclipse, IntelliJ, NetBeans, Visual Studio. |
| **Operating system** | Windows, Linux, Unix, Mac. |
| **TOOLS** | ANT, MAVEN, MAVEN2, LOGGINGLOG4J,CVS, SVNCLEAR CASE, SURROUND, JIIRA, JUNIT, JMETER, JQUERY, CLEAR QUEST, SPLUNK, GOMEZ, Git, Bitbucket, GitLab, GitHub Actions, Artifactory, Confluence. |
| **WEBSERVICE TOOLS** | TCP MONITOR, SOAP UI, HTTP MONITOR, SECURE CRT. |

**PROFESSIONAL EXPERIENCE:**

**Bank Of America, Charlotte, NC           Aug 2022  - Present**

**Lead Full Stack Java Developer**

* **Lead a** 5-member Full stack Technical Development team with development tasks, follow-up, and monitoring**.**
* Working as a **Lead** and involved in translating technical requirements into technical design documents, establish specific solutions, and leading the efforts including programming and testing.
* Responsible for working with that **team** to design, create and improve new and existing products, platforms, and transactions.
* Conducted code reviews and Provided guidance to software developers for continuous improvement.
* Collaborate with cross-functional teams including **DevOps, QA, and Product Owners.**
* Used **(OOPS**) in java and its concepts such as **Multithreading**, **Exception handling**, **Collections, Concurrency, Data Structures,** **EAI Patterns** and also used Object-Oriented design patterns like **singleton, Factory Method, Builder, Decorator, Observer, Adapter.**
* working with **Core Java 8** features like **Functional Interfaces, Lambda Expressions, Stream API, Optional class, Collectors, String Joiner, For Each (), Time API,**and **Generics** and **java 11** features like **VAR, HTTP Package, Epslon.**
* Integrated **Reactive** Extensions in Spring Boot applications, allowing for the creation of **reactive** **endpoints**, handling non-blocking I/O operations, and streamlining data processing.
* Integrated **WebFlux**, a **reactive** programming framework, with **Redis** to build non-blocking and asynchronous payment processing. Designed and developed **reactive** microservices using **WebFlux** and integrated them with **Redis** for data storage and communication between services. Leveraged **Redis** **Pub/Sub** for **reactive** communication patterns.
* Developed end to end application using **Spring** and **Spring Boot** moduleslike **Spring Core (IoC), Spring MVC, Spring Security, Spring Batch,** **Spring Cloud,** **Spring Configuration,** **Spring integration,** **Spring Data JPA, JPA/Hibernate/JSF, ORM, Spring AOP, spring Transaction management.** Designed **MVC** architecture and Front control design pattern using **Spring** **MVC** for presentation layer.
* **Spring Annotations** are used to inject dependencies and **Spring IOC** is used for life cycle management of a Bean.
* Spring framework is integrated with **Hibernate** **ORM** to connect **DAO** layer with relation databases for data retrieval.
* **Micro services** were developed using **spring boot** where Micro service registration is done with spring cloud using **Netflix Eureka**, **Ribbon** is used for Load balancing and **Hystrix circuit breaker** for fault tolerance.
* Developed **REST Services**by implementing **JAX RS**using **Jersey**framework and REST services were documented using **swagger** for clear understanding.
* Implemented **OAUTH 2** in our application to communicate securely with other protected resources by exchanging access tokens for authentication instead of using username and passwords. Single Sign On (**SSO**) authentication is implemented using **Spring** **security**.
* Developed, deployed, and maintained **Spring Boot microservices** for payment processing, Implement **RESTful** endpoints to handle payment requests and responses.
* Implemented caching strategies using **Redis** to store frequently accessed data, such as user sessions, token information. Stored and managed user sessions in **Redis** to enable quick authentication and authorization checks during payment transactions.
* Defined and implemented RESTful APIs using **GraphQL** for efficient and flexible data querying. Implemented **GraphQL** queries and mutations for seamless interaction between clients and the payment gateway.
* **Observable Patterns**are used for connecting to external API’s and **subscribe**to receive data with the help of **RX/Libraries**.
* Implemented event-driven architecture using **Apache Kafka** for real-time communication and data streaming. Designed and defined events and topics related to payment processing and Integrated **Kafka** producers and consumers into the payment gateway for seamless event communication.
* **Typescript** language is used to develop **Angular** applications, providing advanced features like auto-completion, navigation, and refactoring.
* **NPM** is used for managing dependencies in the **Angular** application. **Angular Custom Pipes** were developed to build a search bar as per project requirements. Worked with both **Angular Template-driven forms** and **Angular Model-driven forms** for validations.
* Used **Angular @Input, @Output** annotations to pass data from one component to another and used **Event Emitter** to emit the data.
* Utilized **Angular Routers** for navigating between multiple views without rendering views every time from the server.
* Used **Angular Guards** for authentication to control navigation to requested resources.
* **Bootstrap** was implemented in **Angular** templates for a great look and cross-browser compatibility.
* Configured and deployed **microservices** on **AWS** infrastructure using services like **EC2**, **ECS**, **Lambda**. Analyzed existing **AWS** deployment architecture and plan for the migration to **Azure**. Identify **Azure** equivalents for **AWS** services and determine migration strategies.
* Configured and deployed **microservices** on **Azure** infrastructure using services like **Azure** Virtual Machines, **Azure** Kubernetes Service (**AKS**),**Azure** Functions.
* Containerized **microservices** using **Docker** for both **AWS** and **Azure** compatibility. Implemented Azure Traffic Manager/Azure Application Gateway for efficient routing of incoming requests. and also used amazon API Gateway routing of incoming requests before migrating from **AWS** to **Azure**
* **Kubernetes** was used to deploy containerized applications, and **Replication controllers** were employed to maintain pod copies and for rolling updates.
* Configured **CI/CD** pipelines (e.g., **Jenkins**, GitLab CI) to automate the build, test, and deployment processes.
* **Jenkins** is used for **CI/CD** deployment, and **Maven** is utilized as a build tool. Set up **CI/CD** pipelines using **Azure DevOps** to automate the deployment process. Implemented deployment strategies such as **blue-green** deployments/**canary** releases.
* Planned and executed data migrations between **SQL** and **NoSQL** databases. Written efficient **SQL** queries for data retrieval and manipulation. Ensured transactions adhere to **ACID** properties. Connected application to **SQL** databases using JPA.
* Utilized **NoSQL** database drivers (e.g., **MongoDB** Java Driver) for Java integration. Developed Java components to interact with **NoSQL** databases. **MongoDB** was used to query unstructured data, and **Mongoose connector** was used to establish connections with MongoDB.
* **Unit test scripts** were developed using **JUnit** and **Mockito** and integration testing was performed by Selenium.
* Provided after-deployment support to fix bugs in applications using **CR (Change Request) management**.

**Environment:** Angular 4.0, React JS, Type script, HTML5, CSS3, JSON, Bootstrap, Maven, Reactive, Hibernate, Spring Boot, Spring IOC, OAuth 2, AWS, Rabbit MQ, RESTful, Swagger, GIT, Jenkins, MySQL, NoSQL, Mongo DB, Mocha, Chai, Mockito, Junit, Jasmine, Karma, Webpack, Spring security, Site minder.

**Farmers Insurance Rosemont, IL         June 2020 - Aug 2022**

**Lead Full Stack Java Developer**

**Responsibilities:**

* I worked with **Agile** methodology and actively participated in Scrum meetings as a part of SDLC.
* Orchestrated project planning, defined goals, milestones, and executed strategies to ensure the successful completion of tasks and adherence to timelines as **lead**.
* Assigned tasks to team members, closely monitored progress, and provided guidance to ensure deliverables aligned with project objectives.
* Offered technical guidance, mentoring, and expertise to the **team**, assisting in problem-solving, architecture design, and implementation strategies. Conducted performance reviews, set objectives, and provided constructive feedback to team members, contributing to their growth and development.
* Applied OOD principles and patterns such as **Factory Method** for creating instances of insurance-related objects, Observer for managing event-driven communication between components, and **Singleton** for maintaining a single instance of critical services. Implement the **Strategy Pattern** to encapsulate algorithms for premium calculations and risk assessment, allowing for interchangeable implementations. Adhere to **SOLID** principles.
* Experienced in Java **collections**, **multithreading**  used Java 8 features like **Functional Interfaces, Lambda Expressions, Stream API, Collectors** also used **SOLID** state principles **SRP, OCP, LSP, ISP, DIP.**
* Integrated **reactive** middleware, like **Spring** **WebFlux**/**Vert**.x**,** to manage and handle high-throughput and low-latency data processing, ensuring scalable and responsive applications.
* Employed **reactive** stream processing frameworks such as Akka Streams or Kafka Streams to manage and process continuous data flows with high-throughput requirements, ensuring real-time data processing. **Reactive** **frameworks** like **Vue**.js with **VueRx** or React with **Redux** Observable, applying **reactive** principles in the frontend.
* All the functionality is implemented using **Spring** **Boot** and Hibernate **ORM**. Implemented Java EE components using **Spring** **MVC**, **Spring** **IOC**, Spring transactions and **Spring** **security** modules. Provided both authentication and authorization using **Spring** **security**.
* Spring **core**, **Spring** **Context**, **Spring** **MVC** Framework etc. are used for the application configuration of different components, services in the application and implemented **Spring** **AOP** and Spring IOC.
* Implemented a **microservices** architecture using patterns like **Service Discovery** for dynamic service registration, **Circuit Breaker** for resilience in handling service failures, and Event Sourcing to capture and store insurance-related events. Employ the **API Gateway pattern** to manage and secure API endpoints, and used the **Saga Pattern** for coordinating complex, distributed transactions within the insurance application. Leverage the **CQRS** pattern to separate write and read operations, ensuring efficient querying, and reporting for insurance data.
* Extensively used Spring Application Context for wiring dependencies and used **Spring** **DAO** and **Spring** **AOP** modules to avoid cross cutting concerns.
* **Node JS** with **Express JS** is used in developing **micro service** components to attain non-blocking event driven platform that interact with multiple data sources.
* I worked with **Express JS** which responds to **HTTP** Requests and dynamically renders **HTML** Pages based on passing arguments to templates.
* Developed mobile, Tablet and desktop-based applications to run on server-side **Node JS** Platform.
* **Node.js** is used to route client requests to backend web services and stores user preferences and other data in a distributed database.
* Worked with **Express** JS, which is a fast, web framework of **Node** **JS** that helps manage a server and routes.
* **RabbitMQ message** broker is used to transfer messages from application to customers for payment confirmation details.
* Integrated Spring Boot **Microservices** with Apache **Kafka** for Asynchronous messaging. Developed **Kafka** producer **microservice** and consumers **microservices** to push data into queues and consume data from queues.
* User interface is developed using **Bootstrap, HTML5, CSS3, JSON, AJAX, Es6**,**React.**
* Responsive Single Page Applications (**SPA**) were developed using **ReactJS** for faster rendering of data. Designed modular and reusable **React** components using **Redux**Architecture and Material-UI. **React**-**Redux**is used for state management and **React**-**Router** is used for programmatic navigation.
* JSX attributes were used in React JS to maintain unidirectional flow from parent to child components. Used react lifecycle methods like componentDidMount, shouldComponentUpdate, getSnapshotBeforeUpdate, componentDidUpdate, componentWillUnmount, render, constructor, static getDerivedStateFromProps along with React Hooks like useState, useEffect ,useContext, useReducer.
* Applied **React JS** components like **Functional Components**, **Class Components**, Forms, Events, Keys, Router, Animations in developing the application.
* Worked extensively with **Props** and **States** in **React** to manage data and Implemented the Drag and Drop functionality using **React**-Draggable. Implemented the Drag and Drop functionality using React-Draggable.
* Application was installed on **Azure Virtual Machines** and configured the storage using **Azure Blob Storage.**
* Microservices that were developed using **Spring Boot** are deployed on **Azure Kubernetes Service** (AKS).
* Micro-Services are implemented using the **Pivotal Cloud Foundry (PCF)** built upon Azure cloud services.
* Collaborated with architects to design a target architecture on **GCP**, leveraging services like Compute Engine, Kubernetes Engine, and Cloud SQL. Planned and executed the migration of databases from Azure to **GCP's** Cloud SQL, ensuring minimal downtime and data consistency.
* Containerized the application using **Docker** for consistency and portability. Created **Kubernetes** manifests to deploy the containerized application on **GCP's** Kubernetes Engine. Set up **VPC** and networking configurations on **GCP**, including firewalls, subnets, and DNS settings.
* **Docker**tool is used to build **docker** images from **docker** files where code and dependencies are stored in **Docker** containers. **Docker Hub**is used later to store images acting as a repository.
* **GIT**is used to maintain version control Commit code changes with meaningful messages, ensuring a clear version history. Created branches for feature development, bug fixes, and releases for efficient code management. and Bamboo is used for **CI/CD** deployment. Created and managed repositories for the Application on **GitHub/GitLab.**
* Designed data models for both **SQL** and **NoSQL** databases based on the specific requirements of each microservice. Optimized **SQL** queries to enhance the performance of transactions and reporting functionalities.
* **Cassandra**Database is used to handle unstructured data and connected by using **Cassandra** Node Driver. Tuned queries and indexes in **NoSQL** databases like **Cassandra** for optimal data retrieval in microservices.
* **GRADLE**is used as a build tool in the application and **NPM**is used to build the **Node JS** application, a server-side JavaScript application.
* **BABEL**and **Webpack**are used to build and deploy the application whereas **SPOCK**and **MOCKITO**were used to write and run unit test cases whereas **MOCHA** and **CHAI** were used to test React applications.

**Environment:**  Node JS, Express JS, Spring Boot, Microservices, RESTful API, React, Reactive, Redux, Material-UI, React-Router, JSX, React-Draggable, ECMAScript6, Azure, Azure Blob Storage, AKS, PCF, Docker, GIT, Bamboo, Cassandra, Gradle, NPM, Babel, Webpack, Mocha, Chai, Mockito, Spock, Log4j, Apache Kafka, WADL, Jenkins, NoSQL, HTML5, CSS3, JSON, Bootstrap, JPA, RESTful, React JS, TypeScript, Karma.

**Azalea Health, Atlanta, Georgia         Aug 2018 - May 2020**

**Senior Full Stack Java Developer**

**Responsibilities:**

* Involved in **SCRUM** ,**SPRINT** Meetings,understanding **(SDLC)** for requirements, gathering, modeling, analysis, architecture design, development and Testing of application using **SCRUM** methodology.
* Involved in designing, developing, and testing Enterprise Applications using **Java/ J2EE platform** andused **core java** concepts like **oops, Collections, Concurrency, Data Structures,** **EAI Patterns, Exception Handling, multi-threading, Advanced Java,** **Java11, EJB** used editions like **J2EE,** **Java SE.**
* Applied OOD principles and patterns such as **Singleton** for ensuring a single instance of critical services, **Factory Method** for creating instances of health-related objects, and **Observer** for handling event-driven communication within the application. Utilize **Strategy Pattern** to encapsulate algorithms for health analytics. Implemented the **Composite Pattern** to represent complex health care structures as compositions of simpler objects. By adhering to **SOLID** principles
* **Spring MVC framework** is used to develop enterprise applications and it helps to maintain loosely coupling. **Spring IOC** is used to inject dependencies and manage the life cycle of a bean.
* Implemented **Spring Annotations, Spring AOP** to develop Business layer components and **Spring batch** programming is executed by using a quartz scheduler.
* Developed **APIs** to support patient engagement features, including appointment scheduling, telemedicine, and health monitoring. Implemented secure authentication and authorization mechanisms for patient-related data.
* Developed microservices using **Java, Spring Boot** Implemented business logic for features like health record management, medication tracking, and wellness programs.
* Implemented **microservices** design patterns such as **API Gateway** for centralized API management, Event Sourcing for capturing and persisting health-related events, and **Saga Pattern** for managing distributed transactions. Utilize **CQRS** to separate read and write operations, ensuring optimal performance for analytics microservices. Applied the **Circuit Breaker** pattern to enhance system resilience.
* Implemented **microservices** for health analytics, processing data for insights into patient health trends and population health management.
* Experienced with distributed transaction management with isolation levels with **Spring JPA**.
* **Spring security** is used for authentication and authorization and to provide protection against attacks like **session fixation**, **clickjacking,** and **cross site request forgery**.
* Developed our application using **SOA** developedRestful web service using **JAX-RS** with Jerseyframework.
* **OAUTH 2.0**authentication is used to provide security to Restful web services by generating access tokens.
* **Swagger** is used to create interactive documentation of Rest API and to automate the test scripts.
* **Apache camel framework** is used for message-oriented middleware with rule-based routing and mediation engine.
* **Apache camel framework** is used for concrete implementations of **Enterprise Integration Patterns and** connectivity to great variety of transports and API’s.
* Messaging queues were developed using JMS framework with **IBM-MQ** to communicate between applications by sending messages.
* UI Web pages were developed by using **HTML5**, **CSS3** and JavaScript frameworks like **AngularJS.**
* Used **Angular** **JS** as framework to create a Single Page Application (**SPA**) which can bind data to specific.

views and synchronize data with server.

* **AngularJS** is used to create **Controllers** to handle events triggered by clients and send requests to servers whereas **Angular Directives**like Ng**-Model, Ng-App, Ng-repeat** etc**.** were used to develop views and data binding in our application.
* Worked with **built-in Angular Filters**and developed custom **Angular Filters** inapplication as per business requirements and used **Angular scopes, Angular Routes , Angular Directives.** **Rest API** calls and **data binding** are done by using **Angular JS.**
* **Pivotal Cloud foundry** is used to build and deploy applications. Leveraged **PCF** to facilitate the seamless building and deployment of scalable and resilient applications.
* Utilized **AWS** services like **EC2, ECS, or EKS** for resource provisioning based on microservices requirements.
* Configured and managed **CI/CD** pipelines using tools like Jenkins/GitLab CI to automate the end-to-end development lifecycle. Integrated **CI/CD** platform as part of the GitLab version control system.
* Implemented **CI/CD** pipelines using **AWS** Code Pipeline and Code Build for automated building, testing, and deployment of microservices.
* **Hibernate ORM** is used to achieve persistence and make connections with MySQL Database. Applied **Oracle** Database performance tuning techniques to optimize SQL queries and improve overall system performance.
* Backend interfaces are developed using **stored procedures** and **PL/SQL packages**.
* **Log4j**and **JIRA is** used for logging and debugging.
* Test cases are developed and implemented using **Junit testing**.

**Environment:** AngularJS , HTML5, CSS, JSON, Bootstrap, MAVEN, Hibernate, Spring Boot, Spring IOC, Spring Batch, Spring cloud, AWS, EH cache, Jax Rs, Jersey, Jax WS Apache CXF, Spring Batch, IBM-MQ, SVN, GULP, Protractor,1 Pivotal cloud foundry, Log4j, JIRA.

**BayCare Health System Inc, Tampa, FL     Nov 2015 - July 2018**

**Full Stack Java Developer**

**Responsibilities:**

* **Collection** framework is used in business layer to retrieve multiple records, inserting records in database.
* **Multithreading** is implemented in our application to handle multiple requests to boost performance.
* Developed Enterprise application using **Spring MVC**to maintain loose coupling between layers.
* **Spring IOC** container is used to manage the life cycle of a bean and used core annotations like @**autowired** to inject dependencies in our application.
* Configured **Dispatcher servlet** for processing requests and response and used **View Resolver** to map web pages.
* Spring **AOP** module is used to handle cross cutting concerns like **transaction management, logging, and profiling.**
* **Spring AOP** is used for **exception handling** and useractivitytracking whereas **Spring Security** is used to provide support for **Authentication** and **Authorization.**
* Developed Data layer using **Hibernate** framework, configured **XML** mapping files, wrote **POJO** classes and **PL**/**SQL** stored procedures.
* **Hibernate ORM**is used to connect **DAO** layer to **MYSQL** database andImplemented in-built **CRUD** operations to retrieve data from database.
* **HQL** is used to write complex queries for data retrieval.
* Developed Web pages in our application using **HTML, CSS, Ajax. jQuery** is used for handling user events, webpage manipulation and fetching data from servers with cross browser compatibility.
* Implemented Two-way databinding in **AngularJS** which helps the application to handle automatic synchronization of data between the model and view components using typescript and **angular.**
* Created **Angular** 9 components, implemented interpolation, input variables, Bootstrapping, NgFor, NgIf, Router Outlet, binding the events, decorators.
* Involved in writing **PL/SQL Stored Procedures** in order to reduce the number of calls sent to database.
* Employed cloud-based deployment solution **GCP** to efficiently host and manage the Application.
* Utilized **Docker** ,**Kubernetes** to streamline application deployment, implementing **CI**/**CD** pipelines for automated build, test, and release processes.

**Environment:** HTML, CSS, Ajax, jQuery, Backbone.js, Handlebar.js, Ember Js Spring MVC, Hibernate, Spring AOP, Spring Security, Junit, MySQL, Hibernate, HQL, collections.

**IBM India private limited, Hyderabad, India June 2012 - July 2014**

**Java Full Stack Developer**

**Responsibilities:**

* Implemented **Multithreading** in our application to improve performance and used core java concepts like **Collections** and **Exception Handling** to handle Exceptions.
* Developed our application using **Spring MVC** framework and core annotations were used to inject dependencies.
* **Spring AOP** module is used to handle transaction management services for objects in any Spring-based application and **Spring Batch**is used to convert Xml files to database records using **Quartz Scheduler**.
* **Hibernate** ORM is used to make connections with MYSQL database and to make database transactions.
* Developed **SOAP** web services by implementing **JAX WS** using Axis **2.** andLoad testing of **SOAP** Web Services is performed by using **SOAP UI** and **Apache JMeter** tools.
* Developed Front End Web pages using **HTML**, **CSS**, **jQuery,** and **AJAX** and Involved in implementing the complete Application in the in-build MVC Architecture with **angular** and **typescript**.
* Developing the components using **Angular** needed for the application from scratch and customizing from various **Angular** related libraries to meet the application's functionality.
* Utilized **Angular** to implement rich functionality in various pages: formvalidation**,** CRUD**,** grid**,** search, sort, pagination, multi-selection, tags-input, advanced directives, authentication, server-side proxy, unit tests, cloud integration, slide pages, users switch etc.
* Developed UI using **JSF** andprime-faces components and utilized various **AJAX** features that are provided in Prime faces and **jQuery** is used to load the content dynamically without reloading page by make use of **AJAX** and to provide cross browser **compatibility**.
* Developed UI using **JSF** and **prime-faces** components and utilized various AJAX features that is provided in Prime faces and by using **JSF** framework developed User Interface using reusable UI **components**, **Events**, **Listeners** and **Validators**.
* **Prime faces** are used with JSF to enrich web pages and **AJAX** validation done by Prime faces.
* View pages are created by using components of prime faces in **XHTML** files and configured **JSF** framework.
* **Jenkins** is used for continuous deployment and Continuous integration process.
* Developed optimized **SQL** queries to enhance the query performance and written **store procedures** in **MYSQL** database.
* **Log 4j** is used to record logs and for error tracing.
* **UNIX Shell Scripts** are developed for batch jobs to transfer data on different servers.

**Environment:** HTML, CSS, Ajax,JQuery**,**Primefaces **,**JSF Spring MVC, hibernate, MySQL, Multi-threading, SOAP web services, WSDL, Drools rule engine, Shell, Maven, Spring AOP, Spring Batch, UNIX shell scripts, Java Server Faces (JSF).