ESWAR V

Contact Number: +1 (614) 654-1642

Email: [eswarjava633@gmail.com](mailto:eswarjava633@gmail.com)

# Professional Summary:

* 9+ years of experience as full stack developer in Analysis, Design, Development, Implementation and Testing of web based, enterprise business applications using the JAVA/J2EE technologies.
* Expertise in working with SDLC Waterfall, Test Driven Development and Agile software methodologies.
* Expertise experience in designing and developing user experiences of client-side web applications using **HTML5/HTML, CSS/CSS3(SASS, LESS), JavaScript, TypeScript, jQuery, DOM, JSON, AJAX, Bootstrap, ES5** and **ES6.**
* Extensive experience in **HTML5** and **CSS3** attractive features and worked on usability, accessibility and resolved cross browser issues.
* Proficient in using **Bootstrap** and **CSS** Media Query to create Responsive Web Designs (**RWD**).
* Extensive experience in developing **MVC** architecture for client-side web applications with **JavaScript** frameworks such as **React JS**, **Angular JS, Angular 7.x/11.x** for unique and customize front end app development also to create Controllers to handle events triggered by clients and send request to server.
* Solid experience in developing the rendering view **React JS** components, forms, events, routers and **Redux asynchronous** functions and implemented **React - Flux** pattern for core dependency.
* Extensive experience in developing **model-view-view-model** architecture for client-side **TypeScript** frameworks such as **Angular 11.x.x/8.x.x/7.x.x** to develop single page, rich and dynamic web applications.
* Expertise in using **Shadow DOM** which is a feature of **Angular 11.x.x/8.x.x/7.x.x** to encapsulate visual behavior.
* Good knowledge in providing support for passing messages between publishers and subscribers using **observables** as a part of **Angular 11.x.x/8.x.x/7.x.x.**
* Experience in designing and developing various **Angular 11.x.x/8.x.x/7.x.x** Template basics, Template directives, Routing and navigation, Components, Services, Data Binding etc.
* Experience with features of **Angular 11.x.x/8.x.x/7.x.x** like **Ng-Modules** which configure the injector and the compiler and help organize related things together.
* Expertise in using of **JavaScript** build tools and task runners like **Webpack** for **React** and **Angular 11.x.x**
* Extensive work experience in usage of bundle package and task runner tools like **NPM, Bower** on UI testing with **Karma**, **Jasmine** for **JavaScript** frameworks
* Hands on experience in using many Core Java 8 and Amazon Corretto 11 features like lambda expressions, Functional interfaces, Steam API.
* Extensive experience with **Core Java, Multithreading, Transaction Management, Exception Handling** and **Collection API.**
* Implemented Micro service architecture using **Spring Boot** and **Spring REST** to create REST end points and integrated **Swagger 2.0** Spring Boot to get dynamic documentation.
* Developed Multi-Threaded Producer and Consumer MQ applications using **Spring boot** and Implemented exception handling in **Java** **Spring boot** for **REST API** by making use of **Exception Handler and Controller Advice annotations.**
* Experience in applications using SOA architecture with SOAP, Restful Services and used XML for message communication.
* Expertise in developing **SOA** (Service Oriented Architecture), implementing Restful webservices based on (JAX-RS) using frameworks like **Jersey**, **Spring** **AOP**, **Spring REST**.
* Hands on expertise in Spring modules such as **Spring REST, Spring JTA, Spring AOP, Spring JDBC, Spring Transaction Management** and **Spring Cloud.**
* Implemented Batch jobs to deal with large number of chunks using **Spring Batch** Framework to execute the similar jobs simultaneously.
* Extensive experience with **Java** based **ORM** frameworksinvolving **JPA, Hibernate JPA, Spring JPA, Spring Data, Spring JSP, Spring JDBC** andSpring Data with **Reactive Cassandra**.
* Hands on experience with **Python, Flask API, Pip** and implementing python server less applications using **AWS Lambda functions, AWS Glue ETL Processes** and **Step functions** with pandas, cx\_oracle and etc...**,**
* Experience in Database Design, Creation and management of Schemas, writing Stored Procedures, views, functions, Triggers, DDL, DML SQL queries. Experience in writing **SQL** and **PL/SQL** programming.
* Experience with **SQL** and **PL/SQL** to query various Relational Database Management Systems including **Oracle, MS SQL, MySQL,** and **PostgreSQL.**
* Experience in working with **NoSQL** databases **Mongo DB,** **Apache Cassandra** and **DynamoDB**
* Used **Hibernate** **ORM** framework as persistence engine, configured O/R mapping and wrote **hibernate** queries.
* Exceptional Experience with **NoSQL** based **ODM** frameworks like **Mongoose Connector** and **Cassandra Node Driver.**
* Comprehensive knowledge of J2EE standards and compatible application servers **Tomcat, JBoss, Eureka with Zuul**.
* Strong understanding in implementation using **Spring Cloud-Netflix OSS stack** for **Micro-services** deployment and auto configurations like **Eureka (Service Discovery Server), Ribbon (Dynamic Routing and Load Balancing), Hystrix (Circuit Breaker), Zuul (Edge Proxy)** andMonitoring and centralized log analyses **(ELK).**
* Experience in Message Queue services like **Apache Kafka, Active MQ and IBM MQ** andworked with **Kafka Cluster** using **Zoo Keeper**.
* Working experience with monitoring tools like **Splunk, Datadog, Kibana** and **AWS cloud watch log.**
* Extensively used **Splunk** and **AWS Cloud watch log** to create dashboards and preparing charts for analyzing dashboards.
* Experience in using various **AWS Cloud** and **Amazon Web Services** components like **EC2** for virtual servers, **S3 buckets, AWS Lambda functions, AWS Glue ETL Processes, AWS Step Functions** and **Glacier** for storing objects**, Elastic Beanstalk, Elastic Cache** and **Dynamo DB** for Elastic load balancingandstoring data.
* Hands on experience using **Docker** for installing, testing, upgrading, removing and running software applications.
* Experience in deploying the applications using **Docker container** services and building continuous integration and continuous deployment pipeline using **Jenkins** with **GitHub** and **Bit-Bucket** repositories
* Created **Jenkins** build pipeline and integrated **SonarQube** to track the code quality effectively.
* Extensively used **Orchestration** container **Kubernetes** for load balancing and to schedule the tasks across the computer cluster and load balancers for backend API services.
* Experience in **Spring Security** Framework for **User** **Access Management,** **JWT** token-based authentication and involved **OAuth 2.0** configuration for authentication and authorization.
* Implemented **JWT** to get the token from the back once the user is verified and used **Interceptor** and **Router-Guard** to do authentication and authorization.
* Worked with build tools such as **MAVEN** and Version Control tools like **Git, Bit Bucket, GitHub, SVN** for maintaining code repositories.
* Used **Log4J** which is a reliable, fast and flexible logging framework (APIs) written in Java, which is distributed under the Apache Software License.
* Professional experience in **Log4J** and **SLF4j** for extensible logging, debugging and error tracing.
* Good Knowledge in documentation of REST API's using **Swagger 2.0** and **WADL.**
* Experience on performing unit Testing utilizing **JUnit, TestNG, Spock, Power Mock, Spring Unit Runner, JMeter, Mock MVC** and ridiculing systems like **JMock** and **Mockito** and ensured code changes won't break existing Unit Tests.
* Experience in functional and integration testing tools like **JBehave**, **Selenium Web Driver, SOAP UI Cucumber (Behavior Driven Development testing)**
* Experience in conveying and arranging application servers, for example, **IBM WebSphere, Oracle WebLogic, Nginx proxy web server, Jboss and Apache Tomcat.**
* Hands on experience with Integrated Development Environments (IDE's) tools like **Eclipse, STS, Visual Studio Code, Spyder** and **IntelliJ IDEA**.
* Possessing good interpersonal skills which were applied in coordinating with development team, QA team and support team as required.
* Have the motivation to take independent responsibility as well as ability to contribute and be a productive team member.

# Technical Skillset:

Operating System Linux, MacOS, Windows

Programming languages Java, Python, Typescript, JavaScript, SQL

Databases DynamoDB, Apache Cassandra, MySQL, Oracle, Mongo DB, PostgreSQL

Web HTML5/4, CSS3, Bootstrap, Angular, React JS, XML, JSON, REST, SOAP

Frameworks Spring, Spring Boot, Hibernate, Jersey RESTful

Application and Web Servers Tomcat, JBoss, Apache web server, Oracle/BEA WebLogic, Jetty Server.

Testing Frameworks Junit, Mockito, Power Mock, Jbehave, Karma, Jasmine

Build Tools Maven, Ant, NPM.

CI/CD Jenkins, SonarQube, OpenShift

Monitoring Splunk, Kibana, Datadog, AWS cloud watch log

Version Control Git, SVN

Cloud Amazon Web Services (AWS) Cloud

Repositories SVN, GitHub, Bit Bucket

# Technical Work Experience:

## Prudential Financial Nov 2022 – Till Date

**Role: Full Stack Developer Newark, New Jersey**

**Project Description:**

**Model Data Input (MDI)** will provide a user interface to users which will help them to create, modify rules to validate the data received from vendors for insurance claims. It will also provide the rules engine and validation engine to run validations on top of received data and generate reports in a schedule manner and a dashboard to track and audit all the validation runs.

**Actuarial Data Souring (ADS)** will provide an automation schedule process for **Life CAD, Alight and ILI** vendors to validate and generate the final reports.

* Used NWOW Agile methodology for developing application and participated in all phases like analysis, specification, design, implementation and testing of SDLC.
* Interacting with client, product managers, developers to build initial models and design.
* Evaluating technologies, reducing the complexity of coding by developing new frameworks, reviewing code and advising to team.
* Designed and developed efficient User Interface using **HTML5, CSS3(LESS), React JS** and **React Bootstrap** framework and media queries to create responsive web pages.
* Worked in using **React JS** components, Forms, Events, Keys, Router, Animations, and Redux
* Added Excel-Builder to download the Tabular data in Excel format using React.
* Responsible for **React UI** and architecture. Building components library, including React Tree, and React-Table Grid.
* Configured **React-Redux** for state management and **React-Router** for programmatic navigation.
* Handled **Http Requests** using **Axios** which supports **Promise** **API** functionality native to **JavaScript** **ES6**.
* Extensive used **AWS Cloudscape Design** componentsto develop react web pages.
* Implemented API services using **Python, AWS Lambda functions with API gateway, Load Balancers** and **Flask API.**
* Extensively worked on **AWS Lambda functions** to implement server-less API services using Python and used **AWS Boto3 library** to access and store data from **Dynamo DB**.
* Extensively used **CX\_ORACLE** and **Pandas** to connect, retrieve and validate data from **Oracle** database.
* Implemented **AWS** solutions using **Lambda functions**, **S3 buckets**, **Dynamo DB, Step Functions, AWS Glue ETL processes, AWS Crawlers,** **Elastic Load Balancer**, Auto-scaling groups, Optimized volumes and **lambda** instances and created monitors, alarms, and notifications for **Lambda** hosts using Cloud Watch.
* Used **Apache Spark** sessionsfor AWS ETL data transformation and save data into S3 buckets.
* Implemented RESTful web services using **Spring Boot, Spring JPA.**
* Used Java 8 features such as l**ambda functions, functional interfaces, streams** and **data time** to develop business logics.
* Created an **Apache** **Maven** project to integrate **Spring** **Boot**, **RESTful** **APIs**, **Oracle** & **Microservices**
* Designed a modern, domain-driven, event-driven **microservices** architecture to be hosted on internal and public Cloud platforms, such as **AWS**.
* Developed an extensive array of **Spring** **Controllers** to address the distinct requirements of the application, encompassing both **DAO** and service layer components.
* Implemented new services within a **microservices architecture** using **Spring** **Boot** **framework**.
* Possess extensive experience with **Spring** and **Hibernate** for effectively managing database transactions.
* Wrote **SQL** queries, stored procedures, and improved performance by analyzing and optimizing execution plans.
* Created **S3** **buckets**, configured and managed policies for **S3** **buckets** and utilized **S3** **buckets** and **Glacier** for storage and backup on **AWS**.
* Utilized the **AWS** **SDK** through **Java** to create an image and **Amazon WorkSpace** in the **AWS cloud**, offering **Virtual** **Desktop** **Images** **(VDI)** to users.
* Demonstrated a strong understanding and practical experience in deploying instances on cloud platforms like **Microsoft** **Azure** and **AWS**.
* Participated in **Continuous Integration** and **Continuous Deployment** **(CI/CD)** process, including **Git version control**, **Jenkins** setup, **Apache** **Maven** build and publishing, and implementation of MDI and Alight Jenkins pipelines.
* Extensively used **Apache Spark** session to connect, retrieve and update data from oracle DB and Redshift.
* **Karma** test runner and **Jasmine** test framework tool are used to performing behavioral driven testing for the **JavaScript** application
* Used **UNITTEST** unit testing framework for major unit testing of web services, validation and rule engines.
* Able to create and manage External Lookups in **Splunk** to match event data with description.
* Implemented **Single Sign-On** and policy-based authentication using **Ping Service** security framework
* Configured **Docker** **container** service and build pipeline **Jenkins CI/CD** tool for continuous integrating and deploying an application and also using **SonarQube** for validation checks
* Able to create and manage External logging in **AWS Cloud Watch** to log event data with description.
* Provided production support/on-call support to fix production issues after the deployment.
* Configured the **Log4j** **framework** to facilitate logging at different levels such as debug, info, warning, and error.
* Used **Spyder** and **Visual studio code** to develop and debug both **Python** and **JavaScript** applications.

**Environment:** **React JS, Redux, Axios, NPM, Jasmine, Karma, Python, Flask Framework, Pandas, Boto3, Java, Spring Boot, Spring Data, Unittest, AWS Cloud, Lambda Functions, AWS Step Functions, AWS Glue ETL, AWS S3, AWS Cloud Watch, Dynamo DB, Jenkins, Docker.**

## Starbucks Corporation May 2021 – Nov 2022

**Role: Sr. Java/J2EE Developer Seattle, WA**

**Project Description:**

**Customer Data Exchange Transaction (CDX-TXN)** provides customer transaction services like first data transaction (FDT), basket transaction, tipping transaction and also provide customers to earn rewards, points using their SVC and Non-SVC card purchases and also used to redeem the points.

* Used Agile methodology for developing backend code, coordinated project activities, and actively participated in the **Software Development Life Cycle (SDLC),** including Requirement Analysis, Development, System Testing and Unit Testing.
* Involved design discussions with product managers, developers to build initial models and design.
* Creating a documentation to each small work to understand the project workflow to new team members and other teams who are collaborating with us.
* Engaged in **Object** **Oriented** **Programming** **(OOP)** using **JAVA/J2EE** technologies such as **Core** **Java** (**OOP**, **Collections**, **Threads**), **JSP**, **JDBC**, **Spring**, **Hibernate**, and **Restful** web services.
* Implemented version based RESTful based Microservices using **Spring Boot, Spring Data** and **Spring AOP**.
* Migration of project from Scala to Java.
* Used many Core Java 8 features like lambda expressions, Functional interfaces, Steam API. Used concepts of Collections, Multi-Threading Executor Services and Functional interface programming.
* Developed a **RESTful**-based service for data retrieval, ensuring it is independent from the current application.
* Implemented **Microservices** architecture, establishing seamless communication between **Spring** **Framework**-based services through a **RESTful** interface**.**
* Executed authentication and authorization measures through **Spring** **Security** and **OAuth 2.0**, capitalizing **Spring** **Boot’s** rapid development advantages for **cloud-based** **microservices**.
* Developed comprehensive unit and integration tests to enhance **microservices** reliability using **Junit** and **Mockito**.
* Used **Apache Cassandra DB** for storing and retrieving configuration data, transaction data and audit info.
* Thoroughly verify data integrity on the **Apache** **Cassandra** **cluster** during batch loading processes.
* Utilized **Spring** **Core** annotations for effective **Dependency** **Injection** **(DI),** employed **Spring MVC** to create **REST** **APIs**, and harnessed **Spring** **Boot** to develop optimized **microservices.**
* Integrated an **Apache** **Cassandra** **NoSQL** database to store key-value pairs, leveraging it for project’s data analytics database needs.
* Integrated **Swagger** to comprehensively document and test **REST** **API’s,** enhancing communication and understanding across the team and with stakeholders.
* Worked on both **Azure Event Hub** and **Apache Kafka** message systems.
* Documented the REST API's using Swagger2 API for briefing the services involved in the application.
* Used **Kibana, Splunk** to configure and monitor the process and also used to debug the applications.
* Integrated with **Splunk**, **Datadog** for API traffic monitoring, health checks for creating dashboards of business metrics.
* Used **Datadog** to check application matrices, processing times and health checks.
* Developed test cases by using **Junit 5**, also using Mockito and Easy Mock for unit testing for each module developed in test driven development.
* Used JUnit for unit testing and **JBehave** for component, functional and Integration testing.
* Used Git for version control and Jenkins for building the CI/CD pipeline and used Maven as a build tool.
* Extensively used **JWT** token-based authentication for the **REST API.**
* Used **JIRA** as the bug tracking system to track and maintain the history of bugs/issues on everyday basis.
* Technical assistance to the team members in resolving issues.
* Involved in Production release work and supporting PROD issues and Provided support to the release testing team at the time of code movement and deployment to live.
* Involved in the UAT Testing along with Business Analyst.
* Used **IntelliJ IDEA** to develop and debug **Java** applications.

**Environment:** **Java 1.8 and 13, Apache Kafka, Azure Event Hub, Jbehave, Apache Cassandra, Swagger 2.0, Junit, Log4j, GIT, GitHub, Splunk, Datadog, Kibana, Maven and Jenkins.**

## S & P GLOBAL Aug 2018 – May 2021

**Role: Sr. Java/J2EE Full Stack Developer New York City, NY**

**Project Description:**

**FIERCE** provides a powerful index calculation engine and report engines to vendors to deal with daily based end of the day (EOD) index calculations in real world stock market for all kind government, municipal corporations bonds, equities and mortgages.

**SPIDER** also providespowerful option calculation engine and report engines to vendors to deal with second based PUT and CALL option calculations in real world stock market for all kind government, municipal corporations bonds, equities and mortgages. Both of these applications will schedule, validate, calculate and report index and option calculations to the vendors based on data provided by them.

* Used Agile methodology for developing application and participated in all phases like analysis, specification, design, implementation and testing of SDLC.
* **Micro Service Oriented Architecture** used to build the Single Page Applications (SPA) using **MEAN** (**Mongo DB, Express, Angular, and Node) stack,** where **Spring Boot** and **Node** with **express** used for the backend services, **Angular 11.x.x/8.x.x/7.x.x** is used in front end and **Oracle SQL, MySQL, Amazon S3 and Mongo DB** areused as Databases.
* Developed and maintained the client-side application using **HTML5**, **CSS3(SASS & LESS)**, **TypeScript**, **Bootstrap**, **Angular 11.x.x/8.x.x/7.x.x with Angular Material and Prime Ng Components.**
* Involved in design discussions and designed few screens for client needs.
* Created client-side validation code using **Angular 11.x.x/8.x.x/7.x.x Validators** and Integrated and dispatched client requests using **Angular Services**.
* Implemented form validation on **HTML** forms using the built-in validation properties and methods in **Angular 11.x.x/8.x.x/7.x.x**.
* Used **CSS3 SASS** to Organize and Maintain, reuse as well as compass **CSS3** styling and Used **Angular-CLI** for initializing and for project setup with scaffolding features.
* Built several **Angular 11.x.x/8.x.x/7.x.x** components, UI services to consume rest services using Component/Service based architecture provided by **Angular** andresponsible for data validation on the Client side by using **TypeScript**.
* Created **custom pipes** implementing Angular for formatting the data before displaying it to the user.
* Developed **angular** views to hook up models to the **DOM** and synchronize data with server as a **SPA.**
* Implemented **Angular** component routers for navigation(routing), services to connect the web application to back-end APIs made use of **ES6** features.
* Worked with **custom directives** in Angular to generate reusable components and developed **custom views** using **Bootstrap** components.
* Developed **lazy loading route** with separate modules and child routes including data transfer with Resolve, Route Data and Route Parameter and implemented Rout Gradmechanism using can-activate, can-deactivate can-load- child etc.
* Implemented **Webpack** plugin to perform various tasks that build our application.
* Worked with **NPM** commands and using **package.json** for managing dependencies and dependencies of **Node JS** applications.
* Created **Node JS** middleware application server to encapsulate modern **JavaScript** widget framework and control set.
* Built **RESTful** services using **Node** **JS** and **Express** **API** to communicate with **Angular** front-end for faster data transfer.
* Custom **Node JS** proxy used to simulate **API** responses before client implementation was complete, and to proxy requests to finished client APIs, handling authentication, custom headers, virtual application root, etc.
* Managed projects with **Grunt** task runner and combined with **Bower** package install **JavaScript** latest version of libraries based on running in the **Node** **JS** Environment.
* Responsible for implementing the business requirements using **Spring Boot, Spring Data** and **Spring Security.**
* Extensively worked on **Micro services** for creating API’s services with **Spring Boot** to modularize code and implemented **Swagger** for **dynamic API** documentation. Created **Spring MVC** Controllers, **Spring MVC-REST** servicesbased on **JSON** services.
* Used **Spring Core** Annotations for **Spring** **Dependency Injection** and **Spring MVC** for **REST API’s** and **Spring Boot** for micro-services.
* Developed the application as a **Spring Boot** Application to get range of non-functional features.
* Implemented **RESTful** web services with **JAX-RS** using **Spring Boot** andWorked with **JAX-RS** to create **Web Services** and consuming the existing services
* Used complete annotation driven approach to develop the application using **Spring Boot** and was involved in setting up the **Spring** **bean** profiling.
* Worked with **Spring Boot** for automatic configuration and starter for Maven configuration and Performed metrics, health checks and externalized configuration.
* Developed Micro services using **Spring Cloud Netflix Eureka** to address Load balancing using Ribbon **API Gateway** using **Zuul proxy server,** fault toleranceusing **Hystrix** for circuit break down**.**
* Developed **Micro services** with **Spring Eureka** to retrieve API routes and Implemented Spring Circuit breaker pattern, integrated **Hystrix** dashboard to monitor **Spring** M**icro Services**.
* Implemented security functionality with **OAuth** **2.0**, **Spring** **security**, **SSL** and **site minder** (**SSO**). Secure authentication with **OAuth 2.0** for generating tokens in the application using **Spring Security**.
* Configured **Spring Data** second level **cache** using **EhCache** to reduce the number of hits(traffic) to the configuration table data.
* Used **MySQL** Developer for development of **MySQL** local data base with **Spring JPA** frameworkfor mapping and creating tables, creating stored procedures and back-ups.
* Handled unstructured data by using **NoSQL** databases **Amazon S3** and **MongoDB** by writing rich queries and achieved data in a collection to be distributed across multiple systems for horizontal scalability as data volumes increase through Automatic Shading.
* Composed and executed **Mongo DB scripts** and queries to fetch and test data from **NoSQL data bases** and **API** functions to insert, search, delete and update (key, value)
* Maintaining and troubleshooting **Amazon S3** Clusters this includes monitoring and performing troubleshooting as per requirements.
* Created computing and data resources on **Amazon Web Services (AWS)** using **EC2**. Installed **Pivotal Cloud Foundry** on **EC2** to manage the containers created by **PCF**.
* Used **Pivotal Cloud Foundry** to quickly build, deploy and update the applications and managed Pivotal Cloud Foundry based platform built on **Kubernetes** programming with deployment manager and containerized using **Docker**.
* Used **Tortoise Git** for **Git** (Subversion tool) and **Jenkins** for building the **CI/CD** pipeline and used **Maven** as a build tool.
* Experience with container-based deployments using **Docker**, working with **Docker images**, **Docker hub** and **Docker registries**.
* Deployed application into **Docker container** and **Docker** consoles for managing the application life cycle on **Kubernetes deployment** manager and monitored on **Git**.
* Focused on **Test Driven Development (TDD)** for UI testing we have used **Jasmine** and **Karma** for **Angular, Mocha** and **chai** for **Node JS.**
* Developed backend unit test cases by using **TestNG**, also using **Junit4**, **Mockito, MockMVC, PowerMock,** and **Easy Mock** for unit testing for each module developed in **Test Driven development**. Used **SpringJunit4Runner** Functions for integration Testing
* Expertise with employment of **ActiveMQ** and **Zookeeper** apart from **JMS** as messaging service.
* Worked with the automation team to understand the **Selenium scripts** and review the code, run the tests and review the results and provide inputs to the team.
* Created data processing pipeline using **ELK Stack (Elastic-search, Kibana, Log-stash)** to ingests data from multiple sources and then visualize data using **Kibana**.
* Implemented **Kibana** data visualization plug-in for **Elastic Search** in **Mongo DB** for **synchronization** of data in the database.
* Added and configured **Swagger** middleware for generating useful documentation and for describing **REST API’s.**
* Extensively used **Rational** **Clear case** for version control.

**Environment:** **Angular 11.x.x/8.x.x/7.x.x, Java 1.8, Amazon Corretto JDK 11, TypeScript 3.7.x, Node JS, MySQL, Spring 4.2.0, AWS, Amazon S3, Mongo DB, Oracle SQL, REST services, Apache CXF, OAuth 2.0, Swagger 2.0, Jasmine, Karma, Grunt, ActiveMQ, Junit 4, Log4j, Selenium 3.0, GIT, Maven, Apache Web server and Jenkins.**

## NBCUniversal Media LLC Jul 2017 – Aug 2018

**Role: Sr. Java/J2EE Full Stack Developer Englewood Cliffs, NJ**

**Project Description:**

**Mediator** **8** application provides users to update and modify the live data feeds for sub titles and news telecast and also provide data formatting and scheduling features to allow better monitoring live feed news

* The application is built by **Micro Service Oriented Architecture** using **MERN** (**Mongo, Express, React and Node)**stack and developed Single Page Applications (SPA).
* Designed and developed efficient user Interaction screens using **HTML5, CSS3(LESS), React JS** and **Twitter Bootstrap CSS** framework and media queries to create responsive web pages
* Responsible to Style, look and feel of the web page with **LESS** that extends CSS with dynamic behavior such as variable, mixing, operations and functions.
* Worked on **React JS** **Virtual Dom** and **React** views, rendering using components which contains additional components called custom **HTML** tags.
* Implemented **React** container and presentational components (as Stateless and Functional components when applicable).
* For client-side view rendering services, used **React JS** typically used various predefined **Redux** libraries
* Implemented various screens for the front end using **React JS** and used various predefined components from **NPM** (Node Package Manager) and **redux Library.**
* Worked in using **React JS** components, Forms, Events, Keys, Router, Animations, and **Redux**
* Configured **React-Redux** for state management and **React-Router** for programmatic navigation.
* Implemented stable **React** components and stand-alone functions to be added to any future pages.
* Utilized **create-react-app** to launch product to stay update with the most current **webpack** and babel build configurations.
* Built-in **Axios AJAX HTTP** Service with observables is used extensively for making quick and incremental updates to UI of the application
* Handled **Http Requests** using **Axios** which supports **Promise** **API** functionality native to **JavaScript** **ES6**.
* Worked on **ECMA 5** (**ES5)** features implemented in Google's V8 which is used by **Node JS**
* Implemented **Node JS** as a server-side proxy for an event-driven, non-blocking I/O model to make the application lightweight and efficient
* Developing **RESTful** web services using **Node JS** used **Node JS server** to interact with the Rest services and database, hosted on multiple load balanced cloud instances.
* Implemented **Node JS** **Utility Modules, Buffers, streams, Event loops** for developing the application at server side.
* Used **Node JS** to run **Grunt** tasks and compile, minify and build the project properly.
* Utilized the inbuilt **NPM** package manager to configure the right versions of the packages and to achieve dependency management.
* Implemented **RESTful** Microservices using the **Spring boot, Spring MVC** and **Hibernate**
* Extensively worked on **Jersey** for creating **REST API’s services** with **Spring Boot** to modularize code and used **Spring Data** to access data from **MYSQL** database.
* Implemented a robust **RESTful** web service and automated documentation to the web services in the application provided by **Swagger.**
* Configured **JBoss** Application server and used **Oracle database** as a local Database with **Hibernate ORM** framework and used the functionalities of **Hibernates** second level **EhCache.**
* Created **Hibernate mapping files, sessions, transactions, HQL Queries** to fetch data from data base.
* **Hibernate Session Manger** used to configure multiple data source and provide independent database connectivity.
* Used **Mongoose connector as ODM Framework** to connect with **Mongo DB** and worked with Mongoose libraries like validation, type casting, query building.
* Restructured database schemas using mongoose migration systems for **Mongo DB** and Involved in implementing Queries to the backend **Mongo DB** database using Http service to fetch **JSON** contract.
* Unstructured data was handled using **Mongo DB** and used **Mongoose** Connector for connecting to database.
* Installed, configured, and administered **NoSQL** database clusters in **AWS** Virtual Private Cloud Network for available data on the cloud.
* Implemented **AWS** solutions using **EC2**, **S3**, **RDS**, **EBS**, Elastic Load Balancer, Auto-scaling groups, Optimized volumes and **EC2** instances and created monitors, alarms, and notifications for **EC2** hosts using Cloud Watch
* **Unit** testing of web services is done using **Spock** framework by extending the **Groovy** syntax
* Configured **Radis** (Remote Dictionary Server) with node server for caching the responses via **cat box**
* Unit testing for **Node JS** is done using **Mocha** and **chai** and for **spring boot** used **Mockito, JUnit, MockMVC** and thereby achieving **Test Driven Development** (TDD)
* **Karma** test runner and **Jasmine** test framework tool are used to performing behavioral driven testing for the **JavaScript** application
* Used **JWT (JSON web tokens)** mechanism for authentication and authorization security configurations using **Node JS.**
* Implemented **Single Sign-On** and policy-based authentication using **Site Minder** security framework
* Able to create and manage External Lookups in **Splunk** to match event data with description.
* Configure and Install **Splunk** Enterprise and **Apache** Server for user and role authentication and **SSO**
* Configured **Splunk** data visualization plugin for **Elastic-search, Log-stash** and **created bar**, line and scatter plots, or pie charts and maps on top of large volumes of data
* Configured **Docker** **container** service and build pipeline **Jenkins CI/CD** tool for continuous integrating and deploying an application on servers
* Responsible for migrating existing modules in **IBM MQ** to **Rabbit MQ** and worked on creating **RabbitMQ** adaptors for decoupling the application dependency
* Implemented **Spring Cloud-Netflix OSS** Stack along with using Ribbon for client-side load balancing, **Netflix Zuul API** Gateway with **Eureka** service registry for reverse proxy infrastructure and **Hystrix** Circuit Breaker for fault tolerance.
* **Kubernetes** is configured in the application for automating deployment, scaling, and management of containerized applications
* Configured New Relic to measure and monitor the performance of the application
* Provided production support/on-call support to fix production issues after the deployment
* Used Eclipse based **Spring Tool Suite (STS)** for development and debugging **Spring Boot** applications.
* Configured and administered **Jenkins** for **CI/CD** into **Tomcat Application Server**.

**Environment: React JS, Redux, Axios, Node JS, Webpack, Pivotal cloud foundry, Jenkins, Docker, Mongoose, Mongo DB, Chai, Mocha, jasmine, Karma, Spring boot, Hibernate ORM, Rabbit MQ, IBM MQ, AWS, Docker, Jenkins.**

## Cerner Corporation Jun 2016 – Jul 2017

**Role: Full Stack Java/ J2EE Developer Kansas City, MO**

**Project Description:**

Cerner Corporation is an American supplier of [health information technology](https://en.wikipedia.org/wiki/Health_information_technology) (HIT) solutions, services, devices and hardware. The project deals with developing of application that helps hospitals to send notifications to patients when their appointments have been cancelled in case of any emergencies.

* Involved in various phases of **Software Development Life Cycle (SDLC)** of the application like Requirement gathering, Design, Analysis and Code development.
* Implemented **Service Oriented Architecture (SOA)** Web Services using **SOAP, WSDL, UDDI** and **XML** using **Apache CXF** framework tool.
* Developed and designed User Interface using **JavaScript, HTML, CSS, DOM, XHTML, AJAX, Bootstrap, Less, AngularJS.**
* Used **HTML5, CSS, jQuery, Ajax** and performed client-side validations using **JavaScript**.
* Worked with **Angular JS MVC framework** including authoring complex directives, services, controllers and filters.
* Used **Angular JS** to utilize its rendering functions and create controllers to handle events triggered by clients and send request to server.
* Designed and developed various UI features using **Angular JS,** performed routing using **Angular Router**.
* Used **AngularJS** Factories and Services to make Restful Calls to get data in **JSON** Format and developed UI modules
* Created two-way data binding in **Angular JS Components** and provide access to the web server.
* Developed user interactive web pages using **jQuery UI** plugins like **Draggable, Resizable, Autocomplete, AJAX, JSON, Angular JS.**
* Utilized **Core Java** concepts like **Collections, Multithreading** and **Exception handling** to develop business components.
* Developed and consumed **SOAP Web Services** using **JAX-WS** and **Apache CXF** and used access to send service requests to Web Services.
* Developed Web services component using **XML, WSDL, and SOAP with DOM** parser to transfer and transform data between applications.
* Executed web service test scenario by using **SOAP UI** and automated test case using **SOAP UI**
* Designed application to meet international market standards using **Spring MVC Internationalization** and **spring core modules**.
* Developed **Controllers** and handled request using **DispatcherServlet** and redirected the pages to view based on **ViewResolver**.
* Handled static, dynamic pages using **Spring MVC** framework and **spring MVC** forms.
* Server-side validations are achieved using **Spring Validation** framework.
* Used **PostgreSQL** is the local database used to store application related data and to perform operations like Indexing, cursors and triggers.
* Developed **DAO**s using **Ibatis persistence API** for the database operations to interact with **PostgreSQL** database.
* **ORM** framework is implemented using **iBatis** and to access **SQL** data is used to communicate with database.
* Used **iBatis ORM** is a persistence framework which automates the mapping between SQL databases like automatic data mapper.
* Development of **Entity classes** and establishment of relationships between them using Annotations.
* Performed Installation, Configuration and Maintenance of production Cloud Environments such as **Tomcat, Apache, Jetty,** and **JIRA.**
* Utilized **Dynamo DB** for storing the external stream of data and building real time analytics dashboard using **Google Charts API** with **Angular JS Components**
* Performed Load testing of **SOAP** Web Services using **SOAP UI** and Apache **JMeter** tools.
* Deployed and monitored web services using **Spring Cloud Foundry** Managed Domains and Routes with **Spring Cloud Service Registry**.
* **JMS** is implemented over a reliable channel and **RabbitMQ** isused asynchronously as a message queue**.**
* Implemented **Quartz scheduler** to execute batch programming using **Spring Batch.**
* Implemented various validations rules using **JBoss** Business rules in the drools using **Drools rule engine.**
* Developed Test classes using **Junit** and **Easy Mock** and test coverage using EMMA plug-in.
* Utilized **AWS Services** platform to upload data into **AWS S3 buckets**, **creating EC2** instances and to deploy the **Docker** Container services.
* Unit, functional, end-to-end testing is achieved using TestNg framework.
* Integration testing methods and Behavioral driven testing in the application are implemented by using **Selenium and Cucumber** respectively.
* Developed the **ANT** scripts for preparing **WAR** files used to deploy J2EE components.

**Environment: Java/J2EE, Spring Core, JMeter, Dynamo DB, Spring MVC, JavaScript, Angular JS, HTML, CSS, AJAX, Bootstrap, Rabbit MQ, jQuery, Spring MVC, Amazon Ec2, Amazon S3, PostgreSQL, Drools.**

## State Farm Insurance Aug 2015 – May 2016

**Role: Full Stack Java/ J2EE Developer Bloomington, IL**

**Project Description:**

State Farm Insurance is an insurance provider concentrated on meeting auto insurance, health insurance, life insurance and property insurance needs for the customers in the United States and Canada. The Enterprise Claims System (ECS) is a web-based system, which will replace the current mainframe application, Claim Service Record (CSR) system. ECS is designed to meet many goals like Increased Flexibility, Reduced Complexity, better time to market. Features in ECS include but not limited to Automated Tasks, Electronic Task Lists, Electronic Bills, Automatic Payment, and Medical Treatment Lists.

* Developed application using **Sprig MVC** as backendandto create user interface used **Backbone JS,** for storing used **Oracle Database.**
* Worked on a responsive website for smart devices using Responsive design and media queries using dust template by using **HTML, CSS, JavaScript, jQuery, Backbone JS.**
* Developed the UI with extensive use of **Backbone JS** framework that helped to organize the code.
* Designed and developed various UI features using **Backbone JS,** performed routing using **Backbone Router**.
* Used **Backbone JS** to utilize its rendering functions and create controllers to handle events triggered by clients and send request to server.
* Developed various building blocks such as **models, views, events, routers and collections** for assembling the client-side web applications using **Backbone JS**
* Made **Ajax calls** using **Backbone JS** to render the view from the server side.
* Used **jQuery libraries** to query the DOM and handled the Events using **Backbone JS**
* Developed reusable components using Backbone **JS** through **XML** file view templates.
* Implemented **Spring** Beans using IOC and Transaction management features to handle the transactions and business logic, spring AOP to increase code modularity by cross cutting concerns.
* Implemented Internationalization using **Spring MVC Interceptors** and spring core modules.
* Expertise in developing flexible and loosely coupled Application using **Spring MVC** and handled different **HTTP Request** and **Responses** using **Dispatcher Servlet**.
* Implemented **Spring MVC Validation** to validate the form inputs at the server side.
* Redirected the pages based on the Http Request and **handled Exceptions** using **Spring MVC**.
* Implemented **Spring Security** for posing security constraints on **UI** page views.
* Scheduled **Spring Batch** jobs to convert the payment files format to CSV files, used **Quartz scheduler.**
* Implemented **spring batch framework** to create custom readers and writers to work with the application and Spring security for authentication and authorization.
* Used **SpringBeanFacesResolver** i.e., **Spring Resolver** to integrate the **Spring MVC** application with the UI pages.
* Developed and enhanced data processing tool in Java by implementing **connection pooling** and **multithreading.**
* Created stored procedures, triggers, functions, views, indexes using **Oracle** and written complex queries using JOINS and Subqueries to retrieve data.
* Used **Hibernate** as **ORM framework** and **Hibernate JPA** implementation, annotated the classes with JPA annotations.
* Worked with Hibernate **session objects** to interact with database and used **HQL** to query the database.
* Worked with message queueing using **RabbitMQ** for exchanging data between processes, applications, and servers and involved in setting up a connection and publishing/consuming messages from a queue.
* Worked with Maven for project build, dependency and documentation.
* Used **GoCD** for continuous delivery, continuous integration of the build and deployment for release process.
* Used **Drools** rules engine for defining business rules and Validations.
* Developed Applications using Rule Engines, **Drools**. Explored Drools modules such as **Drools Expert**, **Drools Fusion**, and **Drools Flow**.
* Involved in performing Unit and Integration testing by using **JUnit.**

**Environment: Spring MVC, Spring Batch, jQuery, Java, backbone JS, Hibernate JPA, JMS, RabbitMQ, Spring Security, HTML, CSS, JavaScript, Oracle 11g, JUnit 3.x, Maven, Jenkins, Drools.**

## Reliance Industries May 2014 – Jul 2015

**Role: Full Stack Developer India**

**Project Description:**

Reliance Industries Limited (RIL) is an Indian [conglomerate](https://en.wikipedia.org/wiki/Conglomerate_(company)) [holding company](https://en.wikipedia.org/wiki/Holding_company). Reliance owns businesses across India engaged in energy, petrochemicals, textiles, natural resources, retail, and telecommunications. Reliance is one of the most profitable companies in India. The company's [petrochemical](https://en.wikipedia.org/wiki/Petrochemical), [refining](https://en.wikipedia.org/wiki/Refining), oil and gas-related operations form the core of its business; other divisions of the company include cloth, retail business, telecommunications and [special economic zone](https://en.wikipedia.org/wiki/Special_economic_zone) (SEZ) development.

* Developed a **Monolithic application** based on **Model View Controller** (MVC) architecture and generated UI screens using **JSP, CSS, HTML, Java script**.
* Involved in Agile development process based on iterative development.
* Design development of presentation layers using **HTML**, **CSS**, **JavaScript** and **jQuery**.
* Worked with System Analyst and the project team to understand the requirements.
* Developed web pages applying best standards.
* Designed dynamic and multi-browser compatible pages using **HTML**, **DHTML**, **CSS** and **JavaScript**.
* Developed **UI** using **HTML**, **CSS**, **JavaScript** and **JSP** for interactive cross browser functionality and complex user interface.
* Used **Spring** **Framework** to develop and implement the server-side implementation.
* Developed **Session beans** for functionality as per the requirement.
* Used **Spring** framework for implementing the **Web tier** of the application.
* Used different Design patterns like **MVC** pattern, **DAO** pattern, and **singleton** to achieve clean separation of layers.
* Developed Custom tag libraries to separate presentation and processing logic.
* Worked on **Hibernate persistence tier**. This involves developing mapping files and **Hibernate** (**POJOs**/ **DAOs**).
* Build **PL\SQL functions**, **stored procedures, views** and configured **Oracle Database 10g** with JDBC data source with connection pooling enabled.
* Responsible for creating the screens with table-less designs meeting **W3C** standards.
* Implemented **AJAX** to speed up web application.
* Applied **jQuery** scripts for basic animation and end user screen customization purposes.
* Used **AJAX** and **JSON** to make asynchronous calls to the project server to fetch data on the fly.
* Extensively used **Ajax** calls to send parameters and fetch data based on the user-triggered events.
* Manipulated data to both send and fetch data from the server through **GET** and **POST** requests.
* Developed print functionality and download to Excel functionality.
* Prepared Unit test cases for existing functionality as per the requirement and execute the same.
* Implemented various Validation Controls for form validation and implemented custom validation controls with **JavaScript** validation control.
* Used **SVN** as version control. Used **Log4j** debugging tool to monitor the Error Logs.
* Involved in developing **SQL** queries, stored procedures, and functions.

**Environment: HTML, XML and SQL, CSS, JavaScript, JQUERY, JSP, Spring MVC, Hibernate, AJAX, JAVA, Servlets, JDBC, SVN, Oracle 10g.**