**Name: Satya Deep Yannamani**

**Role: Full stack Java Developer**

**Email:** [**Satyadeep332@gmail.com**](mailto:Satyadeep332@gmail.com)

**Ph#: +1(443)-255-1225**

**LinkedIn:** [**www.linkedin.com/in/satya-deep-12b311256**](http://www.linkedin.com/in/satya-deep-12b311256)

**Professional Summary:**

* **Full Stack Developer** with over **7+ year** of professional experience in development and deployment of various Object oriented and web - based Enterprise Applications using **Java/J2EE** technologies and working on the complete **System Development Life Cycle (SDLC).**
* Experience in **Spring Framework** such as **Spring IOC, Spring DAO, Spring ORM, Spring Resources, Spring JDBC, spring wicket, Micro-Services.**
* Professional skills in authentication and authorization with **Spring Security** in the mode **of LDAP, OAuth2, SAML, JWT (JSON Web Token)**
* Experience in designing and implementation of multi-tier applications using web-based technologies like **Spring MVC** and **Spring Boot.**
* Experience in using **IDE** like **Eclipse, Spring Tool Suite** and **version control tools** like **SVN, Git.**
* Experience in **Java 8** features including **Lambda expressions, Method references, Functional interfaces, Stream API, Default methods, Optional class, Collectors class, Parallel array sorting etc.**
* Experience on **Hibernate.** Used **Hibernate Connection Pooling, HQL, Collections, Hibernate Caching, Hibernate Transactions, Optimistic Locking.**
* Experience in designing dynamic web applications using **HTML5, CSS3, JavaScript, jQuery, AJAX, JSON** with responsive web applications using **Bootstrap.**
* Experience with **Angular 2/4** practices in commonly used modules and creating reusable Components, Pipes and Services.
* Experienced in working with **React.JS and Redux.**
* Experience in working on **Node.JS** Fundamentals including **Modules, Callbacks, Events, Streams and Error Handling** by making use of its **Core Features** like **Event-Driven, Single Threaded model and NonBlocking I/ O model.**
* Experience in **JSON** and **XML** technologies such as **XSL, XSLT, XPath, parsers like SAX, DOM and JAXB.**
* Experience in using and implementing design patterns such as **Model-View-Control (MVC), Data Access Object (DAO), Service Locator Factory, Singleton & Business Delegate.**
* Experience in **Core Java** concepts like **Exception handling, Multi-threading, Synchronization, and Serialization, IO, Collections framework, Java Beans, Executor Services and Thread pools.**
* Experience in using **AWS (Amazon Web Services)** services like **EC2, VPC, Cloud Watch, Cloud Front, Cloud Formation, IAM, S3, Amazon RDS, Elastic Cache, SNS, SQS.**
* Extensive experience in writing **Node.js** express server, and using technologies like **JavaScript/JQuery**, **JSON** to make connections between server and databases like **PostgreSQL** and **Web API.**
* Strong experience in database technologies like **SQL, Queries, Stored Procedures, Triggers, Functions, Hooks, Validation, Entity Framework, PostgreSQL and Sequelize.**
* Experience in creating **Docker** images, Containers, push/pull and integrating with **Spring Boot** for running applications securely on container platform by flexible integrating existing processes with **log4J** for logging.
* Experience in application/web servers like **IBM Web Sphere, Web Logic Application Servers, JBoss and Tomcat Web Servers.**
* Experience on **SOA** and Web Services like **SOAP (JAX-WS) & REST (JAX-RS)** using different frameworks like **Apache Axis2, Spring-WS, Apache CXF and Jersey.**
* Good working experience on **Spark (spark streaming, spark SQL), Scala** and **Kafka**.
* Good knowledge on **Kafka** for streaming real - time feeds from external rest applications to **Kafka topics**.
* Experience in Integrating **Apache Kafka** with and created **Kafka pipelines** for real time processing.
* Experienced in writing test cases and performed unit testing using **JUnit Framework** in Test Driven Development (TDD) approach. Used Postman and **SOAP UI** tools for testing **RESTful** and **SOAP** Web Services.
* Created a **REST API** using **Node.js, Express.js** and **MongoDB** (MEAN) using Redis Cache for performance
* Experienced in writing **SQL, PL/SQL** and Stored Procedures for accessing and managing databases queries with **Oracle, MySQL, NoSQL, MongoDB, SQL Server, and DB2.**
* Excellent interpersonal and communication skills, the ability to provide solutions from a functional and technical perspective, meet deadlines, quickly build client rapport and be a team player.

**Technical Skills:**

|  |  |
| --- | --- |
| **Languages** | Java 8, PL/SQL, SQL |
| **Java/J2EE Technologies** | Core Java, JSP, Servlets, JMS, Java Beans, Java Multithreading, Generics and Collections, EJB, Tiles, Design Patterns |
| **Web Technologies** | HTML5, XML, DOM, JSON, AJAX, CSS3, JavaScript, AJAX, Angular 4/2, Angular.js, Node.js, React.JS< REDUX, React Native |
| **Web Frameworks** | Spring, Hibernate, Spring Boot. |
| **Design & GUI/Tools** | Eclipse, IBM RAD, Log4J |
| **Databases/Tools** | Oracle, SQL Server, My-SQL |
| **Methodologies** | Agile, Waterfall |
| **Operating Systems** | Windows, Linux, UNIX |

**Professional Experience:**

**Client: Texan Credit Union, Denton, TX**

**Dates: May 2021 - Present**

**Role: Full Stack Developer**

**Responsibilities:**

* Involved in various phases of **Software Development Life Cycle (SDLC),** including requirement gathering, modeling, analysis, architecture design, prototyping, developing and testing.
* Developed web application using **Spring MVC** framework.
* Utilized **Spring MVC** framework to implement design patterns like **IOC (Dependency Injection), Spring DAO (Data access objects), Data Transfer objects, Business objects, ORM Mappings.**
* Designed and developed **Micro-services** business components using **Spring Boot.**
* Responsible for creating Batch processes using different modules of **Spring Boot** like **Spring Boot Batch, Spring data etc.**
* Developed the user authentication and authorization by **using Spring Security OAuth2.0 and JSON Web Token (JWT)**
* Responsible for creating **REST API** using **Spring Boot.**
* Developed server side application to interact with database using **Spring Boot** and **Hibernate.**
* Used **Eclipse IDE** for designing, coding and developing applications.
* Designed and developed very complex and large web pages using **HTML5, CSS3, JQuery and Bootstrap** for Dynamic web pages.
* Developed all the **UI** pages using **HTML5, DHTML, XSL/XSLT, XHTML, DOM, CSS3, JSON, JavaScript, jQuery, Ajax**, **Adobe** Creative suite.
* Experience in the Web Application GUI Development using **HTML, CSS, JavaScript, XML and AJAX.**
* Developed **Graphic User Interface (GUI)** to display report data using **JQuery** in Cross-browser compatibility and Web application scaling.
* Used web services **SOAP** and **RESTful** for transmission of large blocks of **XML/JSON.**
* Coded to implement **AJAX** based application by fetching the response from back-end **REST API** as a part of **JSON.**
* Wrote client-Side code using **React.JS** and used **NPM** and gulp directories to generate the server environments used flux with **redux framework** to manage targeted URL’s and to monitor the application.
* Used **React.JS** to implement rich functions in various pages: form validation, crud, grid list, search, sort, pagination, multi-selection, tags-input, advanced/custom directives, authentication, unit tests, etc.
* Used **React.JS** with **RXJS** for building rich internet applications, and to optimize in-browser use and to load the module and to improve the Speed. Linked with API data to view render virtual **DOM** dynamically.
* Built **REST** by building **Node.JS** in the back-end to handle requests sent from the **JQuery Ajax** calls.
* Used **Node.JS** for building web applications with client-server architecture where both can initiate communication in two-way connections and allows them to exchange the data.
* Developed the CRUD API with **Node.js** and interacted with **MongoDB** using Mongoose.
* Involved in Integration of all the modules and deploying them on the Application server and used **JMS.**
* Involved in deploying systems on **Amazon Web Services (AWS)** Infrastructure services EC2. And Utilized most of the AWS services like S3 as a data store for storing the files that falls into the bucket, IAM roles, Elastic Bean Stalk, generated Lambda Functions to trigger an event that occurs in S3, Code Commit as a Version control service, AWS Cloud Watch.
* Used **Docker** for creating Docker images for launching containers on AWS EC2.
* Developed session **EJB** and **message driven bean (MDB)** to process **JMS XML** messages.
* Used **Log4J** for logging and tracing the messages.
* Performed unit testing using **Junit** and developed test cases using **Mockito** with in **Test Driven Development (TDD).**
* Involved in developing **JUNIT** Test Cases to validate the type of data in the **XML Files.**
* Hands on experience in handling Hive tables using **Spark SQL**.
* Redesign **Oracle** application architecture to prepare for migration to **PostgreSQL environment**.
* Create and maintain shell scripts to backup and restore **Oracle** and **PostgreSQL databases.**
* Designed and developed data management system using **PostgreSQL**.
* Involved in designing and developing relational database, validation and hooks using **PostgreSQ**L and **Sequelize**.
* Setup the continuous Integration (CI) process for the application using **Jenkins.**
* Established **PostgreSQL** databases for blog entries, created a client login portal using **JWT** authentication
* Implemented **Service Oriented Architecture (SOA architecture)** using **Web Services (SOAP, REST).**
* Integrated **REST API** using **JWT token** for authentication and security for the microservices.
* Used **Spring Kafka API** calls to process the messages smoothly on **Kafka Cluster** setup.
* Experienced in writing live Real-time Processing and core jobs using **Spark Streaming** with **Kafka** as a data pipe-line system.
* Worked on both Producer API and Consumer API in **Kafka.**
* TLS implementation across the **MQ** and **Kafka** internal communication
* Worked on **MongoDB** database concepts such as locking, transactions, indexes, Sharding and replication.
* Involved in Stored Procedures, User Defined functions, Views and implemented the Error Handling in the Stored Procedures and **SQL** objects and Modified already existing stored procedures, triggers, views, indexes depending on the requirement.
* Developed unit test cases using **Mockito framework** for testing accuracy of code and logging is done using **SLF4j + Log4j**.
* Followed **AGILE** Methodology and participated in **SCRUM** Meetings.
* Actively participated and provided feedback in a constructive and insightful manner during weekly Iterative review meetings to track the progress for each iterative cycle and figure out the issues.

**Environment:** Java 8, Spring, Spring Boot, MVC Micro services, Hibernate, HTML5, CSS3, JavaScript, React.JS, REDUX, React Native, DOM, Eclipse, AJAX, JQuery, JSON, Node.JS, AWS, Docker, Kafka JMS, XML, EJB, Log4j, JUnit, Mockito Jenkins, SOAP, RESTFul, MongoDB, SQL, PostgreSQL Agile and Windows.

**Client: Christus Health, Irving, TX**

**Dates: October 2019 - April 2021**

**Role: Full Stack Developer**

**Responsibilities:**

* Involved in Design, Development and Support phases of **Software Development Life Cycle (SDLC).**
* Worked on **Java 1.8** features such as **Lambda expressions** for making anonymous inline function calls by using **Functional Interfaces** which makes code readable and concise.
* Designed the **N-Tier** architecture for the application following **Spring MVC** pattern.
* Used **Spring MVC** as Web module, involved in design, development of new Action classes to interact with business logic and Used **Hibernate** as **ORM** to map **Java classes** to database tables.
* Used **Micro Services** architecture, designed and developed **RESTFUL** web services using **Spring Boot** framework.
* Used **Spring Boot** framework to create properties for various environments and for configuration.
* Deployed **Spring Boot** based micro services **Docker container** using **Amazon EC2 container** services and using **AWS admin console.**
* Used **Eclipse IDE** to develop an application.
* Implemented the business layer by using **Hibernate** with **Spring DAO** and also developed mapping files and **POJO java classes** using **ORM.**
* Designed & Developed **JAVA API's**, Camel Routes, **REST Web Services**, **MQ/Solace** queue configuration, **Spring beans**, **JMS** Utilities, **UNIX**/Autosys scripts, **Oracle SQLs**/procedures, **hibernate** configuration changes etc.
* Developed an **Angular 4.0** workflow manager leveraging Angular-UI's state router for flexible configuration and management of multi-step user input flows.
* Used **Angular 4.0** for building web-apps and is highly efficient in integrating with **Restful** services.
* Used **Angular.JS** as framework to create a **Single Page Application (SPA)** which can bind data to specific views and synchronize data with server.
* Developed web application by using **Angular 4.0, Angular Material, Angular CLI, REST API, Node.js, Express.js, and MongoDB.**
* Worked with **VUE JS** Forms, Validations, Modules, Components, directives, controllers and HTTP Request.
* Worked on **Vue.js** events distribution and parent chain for parent-child component communication.
* Manipulated DOM, handled events and applied animation effects with **Vue.js.**
* Used **Node.JS** as a proxy to interact with Rest services and interacting with Database.
* Created and configured **JMS connection factories (Queue, Topic)** on Weblogic server.
* Involved in deploying systems on Amazon Web Services Infrastructure **services EC2, S3, Dynamo DB, SQS, Cloud Formation.**
* Developing and maintaining **cloud-based architecture in AWS**, including creating machine image like AMI.
* Implementing jobs using Groovy Scripts for creating **Jenkins** jobs for continuous integration. Utilized most of the **AWS services** like **S3** as a data store for storing the files that fall into the bucket, IAM roles and generated lambda functions to trigger an event that occurs in **S3**.
* Involved in setting up **AWS services** with **EC2 instances, S3 bucket,** took part in deploying application in **AWS** along with writing and retrieving files to and from **AWS S3 bucket** for UI to render data faster.
* Perform application modernization, design re-engineering to migrate existing apps into **cloud platform (PCF, Azure).**
* Worked on **Windows Azure, Azure Active Directory, SQL Azure**, creating resource groups, subscriptions, build definitions and release definitions.
* Experience working with **Microsoft Azure** based **IAAS, PAAS** and **SAAS**.
* Implemented build stage-to build the **Micro services** and push the **Docker container** image to the private **Docker registry.**
* Involved in developing business components using **EJB Session Beans** and persistence using **EJB Entity Beans.**
* Used **Log4j API** for logging in development, testing and production environments, configured to send email notifications.
* Used **Spark Streaming APIs** to perform transformations and actions on the fly for building common learner data model which gets the data from **Kafka** in Near real time and persist it to **Cassandra**.
* Experience in integrating **Apache Kafka**
* with **Apache Spark** for real time processing
* Experience in Message oriented Middleware implementation using **JMS** and **Apache Kafka.**
* Experience in working with messaging systems like **Kafka, Kafka** event sourcing.
* Implemented **Spring boot microservices** to process the messages into the **Kafka cluster** setup.
* Setup and exercising disaster recovery and failover process for **MQ** and **Kafka** servers.
* Enabling Triggering on **MQ queues** based on the requirement of various interfaces.
* Developed test cases to perform unit testing using **JUnit5** and **Mockito Framework**.
* Performed code review and unit tested the application using **JUnit** framework.
* Worked with **Jenkins** in setting up CI/CD pipeline for dev environment and involved in setting up **GIT** web hooks.
* Developed both **Restful and SOAP** web services depending on the design need of the project.
* Worked on **MongoDB** concepts such as locking, transactions, indexes, sharding, replication, schema design.
* Developed **PL/SQL Packages** for loading the image Files to store for different Database Systems.
* Worked on a process for extraction of data from various subsystem and transformation of data in order minimize the reports generation time using **PL/SQL** functions and procedures
* Developed procedure to create **Postgres SQL** instance on **AWS EC2** server.
* Written **SQL Queries** and **PL/SQL** stored procedures to satisfy various business requirements of the application.
* Experienced of Database design in **Oracle/SQL** Server query-based programming and **PL/SQL** procedure programming and proficient in Database programming using **JDBC, Oracle, MySQL, DB2, PostgreSQL, NoSQL, MongoDB,Cassandra DB**
* Used Sqoop to import the data on to **Cassandra tables** from different relational databases like **Oracle, MySQL.**
* Perform analytics on Time Series Data exists in **Cassandra** using **Java API** and Involved in data modeling the tables in **Cassandra**.
* Used **agile** methodology for developing the application.
* Participated in meetings weekly status report and also involved in web operations with clients for gathering requirement information.

**Environment:** Java 8, Spring, Spring MVC, Spring Boot, Hibernate, HTML5, CSS3, JavaScript, Angular 4/2, Angular.JS, Node.JS, AJAX, JSON, REST, JMS, JSF, XML, Java Classes, AWS, Docker, kafka, MQ, EJB, Log4j, TDD, JUnit, Jenkins, SOAP, RESTFul, MongoDB, MySQL, Cassandra, Agile and Windows.

**Client: Nordstrom, Seattle, WA**

**Dates: January 2018 - September 2019**

**Role: Java/J2EE Developer**

**Responsibilities:**

* Involved in various SDLC phases like Design, Development and Testing. Interact with business managers to transform requirements into technical solutions.
* Developed and implemented **spring** and **J2EE** based **MVC (Model-View-Controller)** framework for the application.
* Used **spring** as an application framework to build the Services layer and implemented its core modules such as **Spring Core Container, Inversion of Control, Data Access and MVC** and Used **XML/java /annotation spring boot** **configurations.**
* Implemented **Micro services** based architecture using **Spring Boot** interacting through **REST API.**
* Designed and developed **Micro Services** business components using **Spring Boot.**
* Developed **Spring Boot** application and deployed it into **AWS** using **EC2 instances.**
* Implemented **ORM (Object Relational Mapping)** framework with **Hibernate** for mapping **Java classes** and database tables.
* Used **Eclipse IDE** to integrate the different frameworks to build the overall application.
* Design and code backend web services for web applications with **HTML5, CSS3, JQuery and JavaScript** as frontend technologies and also for consumption for other process.
* Used two-way data binding features of **Angular.JS** to implement Data binding to bind various server controls to the data from database.
* Used **Node.JS** as a proxy to interact with Rest services and interacting with Database.
* Used **JMS** for loosely coupled, reliable and asynchronous exchange of information among **J2EE** components.
* Developed and debugged **EJB** with WebLogic Application server and developed **EJB components** to implement business logic using session and message bean.
* Implemented using **Log4J** for logging errors, debugging and tracking.
* Developed test cases and performed unit testing using **JUnit** classes.
* Used Web services **(SOAP, REST)** for transmission of large blocks of data over **HTTP.**
* Worked with an Agile (Scrum) Development Team to develop a Single Page Web Application based on **Angular JS** for front-end, Spring Boot for back-end, and **Oracle** for database
* Developed different POJO classes to map corresponding tables in **Oracle** and implemented table relationship by using One-To-Many annotation. 
* Followed **agile** methodology that included iterative application development, weekly Sprints and daily stand up meetings.
* Performed a thorough unit testing and resolved several data related issues before migration to production.

**Environment:** J2EE, spring, Spring Boot, MVC, Micro Services, Eclipse, Hibernate, Angular.JS, HTML5, CSS3, JMS, Maven, XML, EJB, Log4j, JUnit, RESTful, SOAP, ORACLE and Windows.

**Client:EdgeVerve, Bangalore, India**

**Dates: October 2015 - December 2017**

**Role: Java Developer**

**Responsibilities:**

* Involved in all phases of **SDLC** to structure the unformatted requirements.
* Involved in designing and implementation of **MVC** design pattern using **spring framework** for web-tier.
* Used **spring MVC** for building the application framework.
* Developed action classes and form beans and configured the **spring-config.xml.**
* Used **Spring Framework’s** Dependency Injection to have a loose-coupling.
* Developed entity's with **Hibernate.**
* Implemented persistence layer using **Hibernate** that uses the **POJOs** (which do not have the business processes) to represent the persistence database tuples.
* Implemented **Service Oriented Architecture (SOA)** using **JMS** for sending and receiving messages while creating web services.
* Worked on parsing the **XML** files using **DOM/SAX** parsers.
* Involved in development and integration of the Application using **Eclipse IDE.**
* Developed **Core Java** components to develop the transaction report. Stored the dynamic data on to Collection Objects and used the predefined methods to perform all DB related operations efficiently.
* Implemented stored procedures and dynamic **SQL** on **SQL Server.**
* Responsible for requirement analysis, designing, planning, tracking and execution of the application.
* Presented website mockups to clients to give them better visual of the end product and to ensure that all parameters were met.

**Environment:** Java, spring, MVC, Hibernate, Eclipse, POJO, XML, DOM, MDB, JMS, Log4j, SQL, Agile and Windows.

**References:** Will be provided upon request.