

Name: Nikhil D
Sr. Full Stack Java Developer
Mobile Number: **862-261-0052**
Email: nikhilkdk76@gmail.com

Summary:

- Having around 7 years of experience in all phases of the **Software Development Life Cycle (SDLC)** which includes Design, Development, Integration and Testing of Cross-Platform (Web & Client-Server) applications using **Object-Oriented Programming, Core Java, J2EE technologies**.
- Intense exposure in developing applications using methodologies like **Agile (Scrum) and Waterfall Model** including creating product / Sprint catalogues, active participation in **Scrum, Sprint planning, Test-driven development (TDD)**.
- Skillful hands-on capability in **Core Java** Concepts like **OOPs, Garbage Collector, Collections, Multithreading, Exception Handling, Enums, Lambda, Generics, Synchronization, Serialization, and De-serialization**.
- Strong Experience in UI Design and Development like **HTML5, XHTML, DHTML, CSS3 (SASS, LESS), Bootstrap, JavaScript, Node JS, JSON, XML and AJAX** that follows W3C Web standards.
- Good ease in building Dynamic **Single Page Application (SPA)** using **MEAN (Mongo, Express Angular, and Node)** full-stack JavaScript framework.
- Proficient in developing web interfaces using **Angular 2/4/5/6/7, DOM Scripting and jQuery**.
- Designed and developed **J2EE MVC** applications using **Struts, Spring MVC and Spring Boot**.
- Expertise in designing, developing, and deploying applications using **J2EE** technologies including **Servlets, JSP, EJB, JSTL, Struts, Spring, Spring Boot, JSF, JMS, AJAX, JDBC, JNDI, GWT Java Beans, Hibernate and Web Services (SOAP, REST)**.
- Hands on experience in test driven development (**TDD**), Behavior driven development (**BDD**) and acceptance test driven development (**ATDD**) approaches.
- Experienced in Middleware persistence frameworks like **Hibernate/JPA** Entities for mapping Java classes using **Hibernate Query Language (HQL), HSQL Named Queries, Criteria, Projections**.
- Extensive experience on Server-side JavaScript structures like **NodeJS (Hapi.JS, Express.JS)** to build event-driven, non-blocking i/o model web applications.
- Experience in **React Js** in building reusable User Interface Components.
- Highly proficient in designing n-tier Enterprise application with **J2EE** design patterns like **IOC** (Inversion of Control), **AOP** (Aspect Oriented Programming), **MVC, Business Delegate, Service Locator, Session Façade, DTO, DAO, Front Controller, Factory, and Singleton**.
- Experience with Asynchronous processing using **JMS IBM MQ Series**.
- Expertise in configuring, deploying web applications on IBM **WebSphere, Oracle/BEA WebLogic, JBoss, Tomcat**.
- Experience in **Microservices** Architecture with **Spring Boot and Docker**.
- Enhanced and deployed **Microservices** based applications using **Spring Boot and Spring Cloud** and created dynamic documentation for **RESTFUL** web service using **Swagger**.
- Experience in Creating and consuming **RESTful APIs** with JavaScript framework Express.js, Node.js and database tools like PostgreSQL and Mongo JS
- Experience in **XML** technologies like **XSL, XSLT, SAX, DOM, XPath, WSDL, DTD, XSD, SOA, JAXB, Apache Axis**
- Developed projects with **SOAP** and **RESTful** Web Services using **JAX-RS, JAX-WS, WSDL, JAX-B APIs, Apache CXF, WSDL**, and implemented using **Jersey, Jackson, and Spring REST** libraries.

- Excellent experience with relational databases – **Oracle, My SQL , DB2 and PostgreSQL** and Understanding in using NoSQL databases **MongoDB, CouchDB, DynamoDB, Graph DB and Apache Cassandra.**
- Sophisticated knowledge in **RDBMS Design, ER modeling.**
- Well versed in creating Queries, **Triggers, Cursors, Views, Functions, Stored Procedures,** and **Triggers,** Indexes Views, Cursors, Joins, Constraints, DDL, DML and User Defined Functions, using **SQL.**
- Experience in using **JBoss JBPM** as a workflow that enables creating and automatization of business processes.
- Having measurable experience with built tools using **Maven, Ant, Gradle,** and scripting **Docker files** and deploying onto **Docker containers.**
- Proficient with container systems like **Docker** and container orchestration **Kubernetes** worked with **Terraform**
- Experience with **Kubernetes** and **OpenShift** to manage the containerized applications, creating and deploying application containers.
- Experienced in **Apache Spark** for implementing advanced procedures like text analytics and processing using the in-memory computing capabilities written in **Scala.** **Depth** understanding of Scala programming language along with lift framework.
- Good working experience in **Amazon Web Services (Amazon SQS, Amazon EC2, Amazon EKS, S3, Amazon Elastic Load Balancing, Amazon Simple DB, Amazon EBS, Amazon Lambda, and Amazon Cloud Front).**
- Extensive experience in Infrastructure Development and Operations involving **k Cloud Services, EC2, S3, Lambda, ECS, EBS, Elastic Load Balancers, Elastic Bean Stalk, Amazon RDS, AWS-Lambda, VPC,** and **IAM Security Groups** that are utilized for different environments like Dev, Testing, Production.
- Implemented automated local user provisioning instances created in **AWS cloud** and **google cloud.**
- Knowledge in implementation of **Pivotal Cloud Foundry (PCF)** services such as **Pivotal Container Service, Pivotal Application Service,** and **Pivotal Functional Service.**
- Working Experience on **Azure Cloud Services, Azure Storage, SQL Azure, Azure Resource Manager, Azure Storage, Azure, Blob Storage, Azure VMs, Azure Functions, Azure Service Fabric, Azure Monitor, Azure Service.**
- CI/CD Automation script writing -Manage and support customer environments in Microsoft **Azure cloud.**
- Hands-on experience in deploying applications using **Jenkins** for **Continuous Integration and Continuous Deployment.**
- Experience in **JUnit Framework** using **Mockito, Easy Mock, Spring Junit, JMeter, Junit, Karma, Jasmine** and Protractor.
- Good experience in using Version Control Systems like **TFS, Git, Bitbucket** for maintaining code repositories.
- Experience in integration tools like **Mule ESB, Spring Integration, Apache Kafka, Kafka connect, Kafka proxy** and **Apache Camel.**
- Practical expertise using IDEs like **IntelliJ, Eclipse, NetBeans, Spring Tool Suite, Visual Studio, Notepad++.**
- Hands on Experience in integration with **JUnit,** and **Log4j** frameworks.
- Knowledge of using Routed Protocols like **FTP, SFTP, SSH, HTTP, HTTPS** and Direct connect.
- Experience with browser testing, knowledge of **cross-browser/cross-platform** compatibility.

- Excellent working knowledge and thorough exposure on compatibility issues with different versions of browsers like **Internet Explorer, Mozilla Firefox, Safari and Google Chrome.**
- Experience in using **React JS components, Forms, Events, Keys, Router, plus Redux, Animations and Flux concept.**
- Good Experience in **React.js** for creating interactive **UI's** using **One-way data flow, Virtual DOM, JSX, React Native** concepts.

TECHNICAL SKILLS:

Programming Languages:	C,C++, Java, SQL, PLSQL, Socket Programming SQL, PL/SQL, XML, XSL, DOM, and JAXB, Scala.
Scripting Languages:	XML, XHTML, HTML, CSS, JavaScript, AJAX, jQuery, JSON, Angular2.0, Bootstrap, Node JS, React JS
Skills:	HTML5, CSS3, JavaScript, JQuery,
Tools:	AngularJS, ReactJS, Redux, BackboneJS, EmberJS, MongoDB, Bootstrap, Knockout JS, Vue JS, Polymer, Bower, NPM, Webpack, Spring, Java, Hibernate, Photoshop, Illustrator, InDesign, Fiddler, Firebug, Maven, ANT, Junit, Selenium web, Driver, TestNG, Protractor, Mockito, Adobe DTM
Testing:	Jasmine, Karma, Mocha, Selenium, Junit 4.1.2, JUnit-Perf, JMock
Software Development:	Waterfall Model, V-model, RUP, Agile, Scrum, Test Driven Development (TDD)
Operating Systems:	Windows NT, 2000, XP Pro, Windows 7, Windows Server 2000, 2003, Linux (Ubuntu, Solaris) and Mac.
Databases:	Oracle, Teradata, DB2, SQL server, MS Access, Oracle 11g, MySQL, PostGreSQL, DB2, MongoDB, Cassandra
Web Services:	SOAP, UDDI, WSDL, JAX-RS, JAX-WS, RESTful Web services, AWS, Microservices.

Web/App/DB	Servers:	IBM WebSphere, WebLogic, Apache Tomcat, JBoss.
IDE Tools:		Eclipse, Net beans, IntelliJ, IBM RAD, ANT, Maven, STS

PROFESSIONAL EXPERIENCE:

CLIENT: TEMENOS

2021 Dec - Present

Role: Sr. Full Stack Java Developer

RESPONSIBILITIES:

- Used **Core Java** Collections, Exception Handling, Multi-Threading, Serialization and **Java 8** features like Lambda expressions, Stream API, Local Variable Syntax.
- Performed Bulk Data operations on **Collections** which would increase the performance of the Application, facilitate pipeline processing and Method references.
- Implemented **JAVA/J2EE** design patterns such as Factory, Template, MVC, Singleton.
- Implemented business process, database retrievals and access of information using **Java** and **Spring Core, Spring MVC, Spring DAO, Spring Security framework**.
- Involved in analysis, design and developing **front end/UI using** HTML5, CSS3, Bootstrap and React and AngularJS
- Developed Single Page Application (SPA) using Angular and injected generic services using **Angular 6/7**.
- Implemented **Angular 6** Routing and navigation, Components, Services.
- Created Typescript reusable components and services to consume **REST API's** using Component based architecture provided by **Angular**.
- Utilized the features of **Angular7 web components, Kendo UI and Human Centred Design** to present effective web design.
- Performed form validations using reactive forms from the **Angular7** framework.
- Built **Angular 7** framework including MVC architectures, different modules, templates, custom directives and custom filters and Used it to store data from backend in model and populated it to UI.
- Built SPA using react and used **React JS components, Forms, Events, Keys, Router, plus Redux, Animations and Flux concept**.
- Worked on creating components, directives, services and routing in **Angular 7** for better component communication by sharing the code between the components.
- Did extensive JavaScript and jQuery programming to give AJAX functionality for the website.
- Used **NodeJS** to run Web pack tasks and build properly the project and used for server-side web applications for **JavaScript** codes to build real-time web APIs.
- Applied **Node.JS** for server-side code and **Express.JS** as the framework.
- Worked with **ASP.NET MVC 3.0, Partial Views, and Master Pages** and developed **Cascading style sheets (CSS)** for consistent look and feel of the website.

- Provided technical design and support on a project while leading hands-on coding efforts utilizing **ASP.NET, .NET Framework REST Services**.
- Used **SOAP UI** tool to test SOA based architecture application to test SOAP services & **REST API**.
- Developed Presentation and Controller layers with JSP, HTML5, CSS3, Custom Tag Libraries, **JSTL, Business layer using Spring IOC and AOP, Persistent layer DAO, JavaScript, and Hibernate for all modules**.
- Used **Spring MVC** (Model View Controller) to handle/intercept the user requests and used various controllers to delegate the request flow to the Backend tier of the application.
- Involved in the authentication process. Configured Spring for customized **JwtTokenProvider**, incorporated **JwtFilter**, yet not conflicting global **CORS** requirement.
- Developed **Microservices** with Spring and tested the application using Spring Boot.
- Used **Spring Core annotations** for **Spring Dependency Injection, Spring MVC for Rest API's and Spring Boot for microservices**.
- Developed distributed **Java** components using **EJB** and **JPA** for persistence.
- Deployed, Scaled, Configured, wrote manifest file for various Microservices in **PCF**.
- Eureka service registry of **PCF** configured for each service to enable communication via the cloud.
- Used in-built marketplace services of PCF for **RabbitMQ, MySQL** config.
- Involved in writing **JPQL Queries** and developed **JPA repositories** in Persistence Layer.
- Developed **Controllers** for request handling using **Rest Controllers**.
- Generated **CSV file** reports for Database dump.
- Experience in handling messaging services using **Apache Kafka** and used Kafka proxy to connect to Kafka cluster.
- Developed data transformer for converting legacy data from source tables to Avro format and fed into **Kafka** using **Camel** to process the file for multiple routes to different host locations.
- Handled consumer logs and monitor **Kafka Metrics** using **Kafka Manager**.
- Implemented advanced procedures like text analytics and processing using the in-memory computing capabilities like **Apache Spark** written in **Scala**.
- worked on **Azure web application, App services, Azure storage, Azure SQL Database, Virtual machines, Fabric controller, Azure AD, Azure search, and notification hub**.
- Used **Restful web services** to communicate with **Mongo DB** and performed **CRUD** operations on Mongo DB using restful web services.
- Used **Express.JS** framework to design **RESTFUL APIs** for **CRUD** operation that interact with Back-end Database.
- Implemented and developed **Restful** web services with **JAX-RS** using jersey as provider and implemented security using OAUTH.
- Created/modified **Oracle** procedures, packages, **Triggers, Tables, Views, and User-Defined Functions** to move the change data across multiple environments using database links.
- Worked on **Oracle** to integrate with spring to retrieve Data Access Objects and Performed **CRUD** operations using **Spring data Oracle-DB repositories**.
- Managed **AWS** (Amazon Web Services) like **EC2, S3 bucket, ELB, Auto-Scaling, SNS, SQS, AMI, IAM, VPC** (Virtual Private Cloud), **Cloud Front** through **AWS** Console and **API** Integration.
- Deployed and managed services on Amazon Cloud. Used **AWS** console and participated in DB replications on **AWS** and used **S3, IAM, Route53, ECS, EC2** and other **AWS** services for API deployments and exposure to VPC, AMI's, Auto Scaling.
- Developed real time tracking of class schedules using Node JS (socket.io based on socket technology, Express JS framework).
- Deployed Application instance on **JBoss** and **Tomcat Server**.
- Developed and optimized services using NodeJS and ES6 as interfaces to Cassandra NoSQL database.

- Redesigned and re-implemented services using NGINX, Node JS, Express JS, Mongo DB, and MySQL.
- Implemented load-balancing with NGINX to allow dozens of Node JS instances to handle thousands of concurrent users.
- Used Jetty and Tomcat for the deployment of the applications developed on Spring Boot using REST/SOAP-based web Services.
- Responsible in developing integration workflows using Mule ESB framework and implemented Data weave and content based routing using Mule ESB.
- Developed CI/CD system with Jenkins on Kubernetes environment, utilized Kubernetes and Docker for the runtime environment for the CI/CD system to build, Test and Deploy.
- Used Amazon Cloud Watch to monitor AWS services and Amazon Cloud Watch logs to monitor application and involved in writing Java API for Amazon Lambda to manage some of the AWS services.
- Used Amazon Elastic search service for application monitoring and full -text search.
- Working with GIT version control and HPALM SDLC lifecycle management of keep track of bugs and backlogs
- Developed business layer components using Spring & Hibernate, and GUI using JSP, GWT & XMI.
- Written Hibernate classes, DAO's to retrieve & store data, configured Hibernate files.
- Involved in bug fixing during the dev and production.

Environment: Java8, HTML5, CSS3, Bootstrap, AngularJS, React JavaScript, AJAX, JQuery, NodeJS, ExpressJS, REST/SOAP Web Services, Spring MVC, EJB, JSTL, JSP, S3, IAM, Route53, ECS, EC2, Jetty, Tomcat, JBoss, Kubernetes, Docker AWS SNS, AWS SQS, GWT, XMI, Amazon Cloud Watch, oracle, Hibernate, NoSQL, TDD, BDD, NoSQL, SQL, GIT.

CLIENT: CENTENE

2019 Nov - 2021 Dec

Role: Full Stack Java Developer

Responsibilities:

- Provided strategic direction to handle **Software Development life cycle** phases as in data gathering, design with Analysis, estimating the timelines, Code development and deployment for the project
- Implemented **Agile (Scrum)** Methodology that includes daily scrum meetings with the team, sprint planning, and estimating the tasks for the user stories.
- Worked on Core **Java** concepts such as **OOP Concepts, Generics, Multi-Threading, Collections Framework, Exception Handling, Garbage Collection.**
- Created **APIs** on the server side to perform **CRUD** operations using **Java 8.**
- Used **Java 8** features in developing the code such as Lambda expressions, creating resource classes, fetching documents from database.
- Developed UI pages and Performed Client-side programming using **HTML5, CSS3, Bootstrap4, AJAX , JSON, jQuery, React JS, Node JS.**
- Extensively used **NodeJS** on the front end and implemented data binding for front-end development for the current web application and used to structure **JavaScript** code to build endpoints.
- Used **jQuery** to traverse through a DOM tree and manipulated the nodes in the tree. Implemented the Drag and Drop functionality using **jQuery** framework.

- Used **React.js/Redux** to build User Interface, knowledge of **state store, middleware, action creator, reducer, and container**.
- Worked on **React JS Virtual Dom** and React views, rendering using components which contains additional components called custom HTML tags.
- Closely worked with Application using **React.JS** and **Node.JS** libraries NPM, Gulp directories to generate desired view and flux to root the URL's properly.
- Developed and performed test cases using **Jasmine** and **Karma** for front end unit testing and used the **Cucumber, Protractor** and **Http-server** for the End 2 End testing.
- Designed and developed an application using **Spring Framework** modules like **Spring IOC, Spring AOP, Spring MVC, Spring Security, Spring Batch and Spring Data** and used **YAML**, a **swagger** used to describe the API 's.
- Used **Spring Security** to provide authentication, authorization and access-control features for Application.
- Used **Spring JDBC** template to perform the database operations. Used **Spring Data Framework** to use the features of **Spring JDBC** and **Spring ORM** classes like **JDBC Template** and **Hibernate Template** to perform the database operations by connecting to Data sources available.
- Used **Hibernate, object/relational-mapping (ORM) solution** for mapping data representation from MVC model to Oracle Relational data model with a SQL-based schema.
- Traced connections to the Mongo database and monitoring utilization of resources for each process.
- Used **Restful web services** to communicate with **Mongo DB** and performed **CRUD** operations on Mongo DB using restful web services.
- Created **JSON** request and validated the response using **Postman** Interceptor and **Swagger UI**.
- In terms of the testing perspective **Postman, JMS** Messenger and **Swagger** are used to perform unit testing.
- Developed a data pipeline using **Kafka, HBase, Spark and Hive** to ingest, transform and analyze customer behavioral data.
- Involved in maintaining and updating the procedure for **ETL** process.
- Moving existing monolithic architecture to **microservices** and migrating them to **AWS** cloud.
- Used **AWS** Services platform to upload data into **AWSS3** buckets and creating **EC2** instances.
- Worked with **Splunk** to read the data and to collect logs, Developed Splunk reports that provide overview of the Application.
- Created plugins to extract data from multiple sources like **Apache Kafka**, Database and Messaging Queues.
- Designed and developed in interface framework to keep the application of the JBPM Interaction.
- Responsible for implementation of Java integration components for **JBPM** and **DROOLS**.
- Created **Maven** and **Gradle** builds to build and deploy **Spring Boot Microservices** to internal enterprise **Docker** registry.
- Setup and managed light weight Kubernetes K3s for self-healing and high availability
- Containerized applications using Docker file best practices, and integrated with different microservices
- Deployed Prometheus with Grafana to monitor the Kubernetes cluster and configured alerts firing when different conditions met
- Migrating Services from On-premise to Azure Cloud Environments. Collaborate with development and QA teams to maintain high-quality deployment
- Designed Client/Server telemetry adopting latest monitoring techniques.
- Worked on Continuous Integration CI/Continuous Delivery (CD) pipeline for Azure Cloud Services using CHEF.
- Integrated GIT into Jenkins to automate the code check-out process.
- Used **Jenkins** as a primary tool for implementing the **CI/CD** during code releases, worked with **CI/CD** pipe line to build and deploy transcoding application on **AWS**.
- Developed **Mockito** and **JUnit** test cases to improve code coverage.

Environment: Java8, Spring Boot, Spring Batch, Spring IOC, Spring AOP, Spring MVC, Spring DAO, Microservices, HTML5, JavaScript, CSS3, XML, ReactJS, Jasmine, Karma, Cucumber, Protractor, NoSQL, PL/SQL, MongoDB, Microservices, Postman, JMS, Swagger, Apache Kafka, Web Services, Hibernate, jQuery, AWS Services, Docker, Kubernetes, Splunk, Jenkins, Mockito, Junit, JIRA, Agile methodology, Maven, GIT, Log4j

CLIENT: NOVARTIS

2017 September - 2019 Nov

Role: Java Full stack Developer

Responsibility:

- Involved in the agile software development process' design and development phases.
- I created a unique, single-page, responsive Web application for the front end using Angular 4.0.
- Angular 4.0 Validations, Modules, Events, Expressions, Directives, Controllers, and Filters were used to create web pages.
- In order to create single-page applications, we used the React Redux architecture, ES6, Webpack, and grunt.
- Developed many front-end panels using ReactJS and a variety of pre-built components from **NPM** (Node Package Managed).
- Participated in the creation of a reusable **ReactJS** component for **DOM** manipulation.
- I spent a lot of time utilizing **CSS3** to arrange **DOM** pages.
- React.js and Node.js packages from **NPM**, gulp folders to create the necessary view, and flux to properly root the URLs were used in close collaboration with the application.
- Create a variety of front-end displays using **ReactJS**, pre-built elements from the **NPM** (Node Package Manager), and the Redux framework.
- I used NodeJS to implement the **HTTP REST API**, and **POSTMAN** to thoroughly test RESTful services.
- Using Spring Boot, I developed Java modules to implement business rules and workflow.
- Improved for Loops and Enums and developed utilizing Java 8 annotations and generics.
- Created documents with **MongoDB** embedded using Java code utilizing MongoDB spring data.
- Worked directly with the operations team to support data migration, monitoring, analysis, and fine tuning as well as the deployment of MongoDB apps.
- Replication between data centers can be used to guarantee the continuous availability of our MongoDB clusters, which are essential for our business.
- Well-versed in the key concepts of **Java**, including **JSP**, **Servlet**, Swing Serialization, and Collections.
- UI was designed and built for customer service modules and reports using **AngularJS**, **HTML5**, **CSS3**, **Bootstrap**, **JavaScript**, and **jQuery**.
- **AngularJS** was used to create the controllers and angular components.
- Used Node.js as a proxy for those connections to the database and other services.
- Utilizing Spring and Hibernate data mapping, I implemented the integration of the business components.
- Used the Spring Boot Actuator to configure an application's properties in various situations. spring rest controllers and services classes have been updated to enable the switch to the spring framework.
- In order to create the Discovery Server, Service, and Discovery Clients, Microservices were developed using Spring Boot, Spring Cloud, and Netflix Eureka. Apache Kafka was also incorporated as a message broker.
- We created microservices with Spring to obtain API routes for the entire cluster. By employing this technique, a single API gateway may be used to load and make accessible each microservice in a cluster.
- Built entity objects and accessed persistent SQL Server data using the ORM tool Hibernate.
- Experience utilizing persistence frameworks like JPA and Hibernate to bind Java classes to databases.

- Experience creating Docker images, using multiple images, and using Docker Hub, mostly for installing middleware and configuring domains.
- I implemented SQL queries, managed JDBC interfaces with data sources, and used SQL Server to access the database at various levels.
- Using Spring Boot and Spring MVC, I designed and implemented specialized message adapter components for Apache Kafka, allowing messages to pass across Microservices with a RESTful API.
- Employed Maven build scripts to compile and package the application, and implemented Log4j.
- I have used special design patterns such the singleton, business factory, and J2EE design patterns.
- Kafka was used to load the data onto the Hadoop file system and transfer it to the MongoDB database.
- Participating in the creation of an internal application using AngularJS, ReactJS, and NodeJS with an Oracle backend connection.
- Create a cluster using Docker Swarm and Kubernetes for Docker container management.
- Building and maintaining Docker container clusters on Google Cloud Platform using Kubernetes, Linux, Bash, GIT, and Docker (Google Cloud Platform).
- The CI/CD system's runtime environment for building, testing, and deploying applications used Kubernetes and Docker.
- Participated in creating a test environment using Docker containers and setting up the containers using Kubernetes.
- Utilizing JUnit, unit tested the Java modules.
- Developed JAX-RS, JSON, and documentation using the Swagger API
- Utilized Hibernate to implement object mapping.
- worked on enhancing change requests and addressing bugs.

Environment: jQuery, AJAX, AngularJS, JSP, AWS, Microservices, Spring AOP, Hibernate, Spring Cloud, Spring MVC, Spring JDBC, Web-Services, REST, JAX-RS, Jersey, JUnit, JAX-B, WebSphere, JIRA, Maven, GIT, Jenkins

CLIENT: APOLLO PHARMACY

2016 June - 2017 August

Role: Java/J2EE Developer

RESPONSIBILITIES:

- Helped to finish the technical design papers by taking part in the creation of functional definition documents and discussions with business users and the testing team.
- The Web Application was enhanced using Spring.
- Business logic and a JSP-based application were created using the Spring Framework plus Servlets.
- I documented the code using Java doc-style comments.
- Client-side validation was developed using the Struts Validate framework and JavaScript.
- Created unit test cases for a variety of modules and addressed test outcomes.
- SOAP was built using Web services in order to connect with other systems.
- In the project, JMS was utilized to transmit and receive messages on the queue.
- The WebLogic Application Server was used to write and deploy JSPs and Servlets.
- Automated build files were generated using Maven.
- Version control was handled via Subversion, and faults were logged by log4j.
- Numerous modules for the application were made using Java as the programming language.
- Worked on setting up web services based on SOAP to provide data to various internal applications.
- Had a hand in the growth of SOA design.
- I was a UNIX operating system developer.

- Using the Log4j and commons-logging frameworks, the application flow was recorded.
- The version control system utilized was GIT.
- SOAP UI is frequently employed to ad hoc test middleware components.
- Working together on testing and fixing bugs with the QA team.
- I developed build and deployment strategies for the QA, TRN, and PRD environments using Bamboo.
- Leadership in DevOps and AppOps for QA, TRN, and PRD deployments
- PL/SQL for Oracle was created. Stored methods and triggers
- Helping the production support team resolve troubleshooting problems
- Took part in the deployment and release management process.

Environment: Java, J2EE, Struts, JSP, JMS, Servlets, JavaScript, Hibernate, SOAP, Oracle, Windows, PL/SQL, WebLogic, Log4j, Maven, CVS, and Junit.