



Java Full Stack Developer

Personal Information

Name: vinaykumar M
 Phone: *Not specified*
 Email: vinaykumarjava88@gmail.com
 Location: US-MO-Kansas City-64114

Last Modified: 2/26/2018

Last Activity: 2/26/2018

Experience

Total years experience: 48 Years, 2 Months

Job Categories: Information Technology (No experience)
 (No experience)

Work History

Company Name:	Cerner Corporation	(2 Years) August 2016 - Present
Job Title:	Full Stack Developer	
Company Name:	AOP	(48 Years) January 1970 - Present
Job Title:		
Company Name:	ACT	(2 Years) August 2014 - August 2016
Job Title:	Full Stack Developer	
Company Name:	Citibank, N.A	(1 Years) December 2013 - August 2014
Job Title:	Sr. Java Developer	
Company Name:	Amica Mutual Insurance	(0 Years) January 2013 - December 2013
Job Title:	Sr. Java Developer	

Education

School: Gandhi Institute Of Technology And Management University, India Graduation Date: June 2010
 Major: Bachelor Of Technology In Mechanical Engineering
 Degree: Bachelor's Degree

Additional Skills And Qualifications

Recent Job Title: Recent Wage:
 Languages Spoken: Managed Others:
 Security Clearance: No Military Experience:

Desired Position

Desired Employment Type: **** NOT FOUND ****
 Desired Travel: Desired commute:
 Desired Relocation:

Resume

VINAYKUMAR
 Java Full Stack Developer
vinaykumarjava88@gmail.com 712-587-7612

Professional Summary:

- * Around 8 years of experience as a Java J2EE Developer in Object Oriented Design, Development, implementation and maintenance of the web-based Client-Server business applications using Java/J2EE technologies.
- * Experience with Agile/Scrum and Waterfall development methodologies.
- * Experience in using the Ajax, JSON and JavaScript/Typescript for building the most interactive web pages.
- * Expertise in designing and developing the interactive UI using HTML5, XHTML, JavaScript, XML, CSS3 (SAAS, LESS), Bootstrap, DHTML, jQuery and Ajax.
- * Experience in using MVC and MVVM client-side JavaScript frameworks like Angular 4, Angular 2.0, Angular JS, Backbone JS, Handlebar JS, Ember JS.
- * Good Knowledge on Angular 2.0/4.0 components (ng Modules), Directive, Services (observable /promises) and Pipes.
- * Experience on Server-Side JavaScript frameworks like node.js, Express.js.

- * Experience in developing Single Page Application using MEAN (Mongo, Express, Angular and Node) Stack.
- * Experience on server-side view rendering Java content structures like React.JS and rendering pages on the server side utilizing Redux for accomplishing the unidirectional plan stream.
- * Experience in unit testing tools of JavaScript frameworks like Karma, Jasmine and Protractor for Angular, Mocha and Chai for React JS.
- * Good Experience in using Core Java features like Multithreading, Concurrency, JDBC, Transaction Management, Collections and Exceptional Handling.
- * Experience in Spring Framework modules like Spring MVC, Spring IOC, Spring AOP, Spring JTA, Spring Security, Spring IO, Spring REST, Spring Batch, Spring Data, Spring JDBC, Spring Boot with Thymeleaf template.
- * Experience in writing SQL queries in PL/SQL, Triggers and Stored Procedures in Relational Database Management Systems such as MySQL, Oracle 10/11g, MS SQL and DB2.
- * Experience in using NoSQL Databases like Mongo DB and Cassandra for handling the unstructured data.
- * Experience on the build tools like Maven, Ant and Gradle to automate the creation of executable applications from source code.
- * Experience in using the NoSQL frameworks like Mongoose Connector and Cassandra Node Driver.
- * Expertise in using the Persistence Frameworks like Hibernate, Java Persistence API (JPA) and iBATIS.
- .
- * Experience on the Security frameworks like OAuth 2.0, JWT token based authentication, Spring Security and Implemented Single Sign-On (SSO) using Site Minder on single/multiple cookie domains for Web applications.
- * Experience in designing and developing the Log4j Logger Encryption by subclassing Log4j ConsoleAppender.
- * Experience in developing the SOAP services on JAX-WS frameworks (Apache CXF, Axis-2) and using WSDL.
- .
- * Expertise in developing the Rest Services (URI based) on JAX-RS (Apache CXF, Jersey), WADL, Swagger respectively.
- * Experience in designing and developing the Microservices using the Spring Boot.
- * Experience on performing testing using JUnit and good knowledge on Mockito and TestNG.
- * Experience on performing Integration testing using Selenium and Cucumber (Behavior Driven Testing).
- * Experience in Java Messaging Service tools like ActiveMQ, Apache Kafka, RabbitMQ and IBM MQ.
- * Good Experience on the Web Servers like Apache Tomcat, Oracle WebLogic, Glassfish, IBM WebSphere, JBoss.
- * Experience on implementing the Swagger for dynamic REST API documentation.
- * Experience in Implementing the Spring Netflix Zuul API gateway and Eureka service registry for reverse proxy and Dynamic service discovery of API's, Ribbon for Load Balancing and Hystrix Circuit Breaker for Fault tolerance.
- * Good knowledge on the Drools Rule Engines which corresponds to validating BRMS (Business Rules Management System).
- * Experience in implementing the PCF (Pivotal Cloud Foundry) PaaS (Platform as a Service) services such as Pivotal Application Service, Pivotal Functional Service and Pivotal Container Service to continuously deliver applications on to the cloud.
- * Experience in using the various Amazon Web Services (AWS) like Amazon Elastic Load Balancing, EC2 for Virtual Servers, AWS Identity, EBS, S3 & Glacier for Storing Objects.
- * Experience on build tools like Jenkins & Bamboo for Continuous Delivery/ Continuous Integration.

Technical Skills:

Programming Languages C, C++, Java, UNIX Shell Script, PL/SQL
 Java/J2EE Technologies Spring, Servlets, EJB, JDBC, JSP, JPA, JTA, JNDI, JSTL, AWT, Applets
 Frameworks Spring 3.0, Spring Batch, Spring Security, Spring AOP, Spring Core, Spring IOC, iBatis, Struts 2.0/1.0, Hibernate 4.x/3.x
 Web Technologies HTML/HTML5, Bootstrap, CSS/CSS3, XML, AJAX
 Databases MySQL, Oracle, SQL Server, NoSQL (Mongo, Cassandra)
 JavaScript Frameworks JQuery, Angular 2.0/4.0, ReactJS, Express.js, Backbone.js, Node.js, Ember.js, Handlebars.js
 Web Services WSDL, SOAP, RESTful, Apache CXF, Axis-2, JAX-RS, JAX-WS.
 Messaging Services RabbitMQ, IBM MQ, Apache Kafka, ActiveMQ
 Tools
 Build Tools (Ant, Maven, Maven2),
 Version Control Tools (CVS, SVN, Clear Case, Surround),
 Logging (Log4j)
 Testing Tools (JUnit, Mockito, TestNG, Selenium, Cucumber,),
 Monitoring Tools (SPLUNK, GOMEX),
 Traffic Routing (BIG F5)

Cloud Amazon Web Services, Pivotal Cloud Foundry, Microsoft Azure
 Methodologies Agile, Scrum, Waterfall model

Education:

- * Bachelor of Technology in Mechanical Engineering from Gandhi Institute of Technology and Management University, India - 2010.

Work Experience:

Client: Cerner Corporation, Kansas City, Missouri Aug '16 to Present
 Role : Full Stack Developer

Description: Cerner Corporation is an American supplier of Health Information Technology solutions, services, devices and hardware. In Cerner I got an opportunity to work on Fetalink Monitor Device upgrade project. Fetalink Monitor is a Cerner produced device which monitors the patient vitals and sends that information for storage and we process the data and store it in our database. We have designed an User Interface where the nurse/doctor can view that information based on the Hospital/Patient.

Responsibilities:

- * Involved in developing a Micro Service Oriented Architecture Application by using SpringBoot , Angular 4.0, Node JS with Express JS and MySQL database.
- * The efficient styling for the web pages was obtained by using the CSS3(SASS) and HTML5 was used for the DOM representation of the application.
- * The \$http Service is used for making the Ajax Calls.
- * Twitter Bootstrap was used for creating the custom templates and obtaining the responsive design.
- * The Component based architecture provided by the Angular4.0 was used for creating and developing the Typescript components which are reusable and services for producing and consuming the Rest API's.
- * ImplementedTypescript for providing deep readability for tools through its optional static typing.
- * For managing the concurrency of the components, the ES6 generators were used.
- * Angular 4.0 route module was used for implementing the Imperative routing and guards and implemented the router component for navigation.
- * Webpack was used for bundling modules. Webpack takes modules with dependencies and generate static assets representing those modules.
- * Angular Unit testing was performed by writing the test cases on Jasmine and Karma was used for running those tests.
- * Implemented Java 8 features like parallelstreams and filters through lambdaexpressions to handle thesearching.
- * The annotation driven approach was implementedfor developing the application using Spring Boot.
- * Implemented Spring Boot for obtaining the production ready features such as Health checks and externalized configurations.
- * Developed the Restful service interface using Spring Boot to the underlying services API and implemented the RESTful web services.
- * Developed the Rest Services on the server side using the Node JS with Express JS.
- * Implemented modules into NodeJS to coordinate with plans and necessities.
- * Implemented Node JS as a server-side proxy for an event-driven, non-blocking I/O model tomake the application lightweight and efficient.
- * ImplementedExpress JS which respond to HTTP Requests and dynamically render HTML Pages based on passing arguments totemplates.
- * Used OAuth 2.0 protocol for authorization of server and application use Access Tokens for making API requests and implemented Site Minder for Single Sign On.
- * DevelopedApache Kafka services for sending and receiving the messages.
- * Implemented the Node Package managerto configure the correct versions of packages and for managing the dependencies.
- * ImplementedSwagger for making the automated documentation to JAR file for Clients.
- * Implemented MySQL for handling the structured data and Hibernate was used as Persistence layer.
- * Implemented the Spring Netflix Zuul API gateway and Eureka service registry for reverse proxy and Dynamic service discovery of API's, Ribbon for Load Balancing and Hystrix Circuit Breaker for Fault tolerance.
- * Used the Docker for deploying the dockerized application in the cloud and launched several AWS EC2 instance to deploy the application to AWS.
- * Used Jenkins for Continuous Integration and Continuous delivery.
- * Junitwas implemented for Unit testing and Cucumber (behavioral driven testing) was for Integration testing.
- * Involved in Application development which followed Agile Scrum Methodology.

Environment: Micro Service Oriented Architecture, Angular 4.0, CSS3 (SASS), HTML5, Bootstrap, Typescript, ES6, route module, router, Webpack, Jasmine, Karma, parallel streams, filters, lambda expressions, Spring Boot, RESTful web services, Node JS with Express JS, OAuth 2.0, Site Minder, Single Sign On, Apache Kafka, Node Package manager, Swagger, MySQL, Hibernate, Spring Netflix Zuul API gateway, Eureka service registry, Ribbon, Hystrix Circuit Breaker, Docker, AWS, Jenkins, Junit, Cucumber, Agile Scrum Methodology.

Client: ACT, Iowa City, Iowa Aug '14 to Aug '16
 Role : Full Stack Developer

Description: ACT assessment measures high school students' general educational development and

their capability to complete college-level work with the multiple choice tests covering four skill areas English, mathematics, reading, and science. I involved with a team to develop the application which collects and stores the students data and display the different questions from the various dumps according to the specific module.

Responsibilities:

- * Involved in developing the Single Page Micro Service Oriented Architecture Application using the Mongo Database, React with Redux, Angular 2.0 and Node JS with Express JS.
- * Involved in developing the efficient UI using HTML5 for DOM representation and CSS3-LESS for good styling of the application.
- * Used the Bootstrap for the custom templating of the web pages.
- * Involved in developing custom directives for making reusable components in Angular 2.0.
- * Used Angular 2.0 to create views to hook up models to the DOM and synchronize data with server as a Single Page Applications.
- * Developed the objects including the components using Typescript, ngModules, interfaces and classes using the Arrow function.
- * Implemented AngularJS Observables and listeners (RxJS reactive libraries) to simplify the making asynchronous calls to backend REST API's.
- * ES6 generators were used for maintaining the concurrency between the components.
- * Multiple views were rendered by using the React JS and implemented JSX for adding XML syntax to the javascript.
- * Implemented various screens for the front end using React.JS and used various predefined components from NPM (Node Package Manager) and redux library.
- * Implemented React JS components, Forms, Events, Keys, Router, Animations.
- * Implemented React with Redux for state-management and Router for the navigation.
- * Implemented Redux to cut some complexity corners by using functional composition.
- * Implemented Store-Redux for holding the whole state tree of application and Reducer-Redux for describing actions.
- * Reducers are implemented for specifying how the application state changes in response to the actions sent to the Store.
- * Implemented Webpack for bundling the modules and maintaining the libraries efficiently.
- * Angular testing was performed using the Protractor and React unit testing was performed using Mocha and Chai.
- * Developed the Rest Services on the server side using the Node JS with Express 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.
- * Developed Node.js Readable and Writable streams to process the data from the external source of the application.
- * Implemented modules into NodeJS to coordinate with plans and necessities.
- * Developing RESTful web services using Node JS and Express JS and used Node JS server to interact with the Rest services and database, hosted on multiple load balanced cloud instances.
- * Implemented Express JS which is fast, assertive, essential and moderate web framework of Node JS that helps manage a server and routes.
- * Implemented Express JS which respond to HTTP Requests and dynamically render HTML Pages based on passing arguments to templates.
- * Implemented Axios for calling the developed RESTful services that is for making the Ajax Calls.
- * Implemented the Node Package Manager build tool for managing the dependencies in the project.
- * OpenID Connect (OIDC) was implemented on the top of OAuth2.0 for authorizing the Rest Services and JWT for token based authentication.
- * Used Spring IOC for the life cycle management of the bean and Java Transaction API (JTA) for the service layer.
- * Developed some standalone Micro services using Spring Boot micro service architectural patterns.
- * Implemented Spring Boot properties like actuators and health checkers to enhance micro service implementations.
- * MongoDB was used for handling the unstructured data and used the Mongoose Connector for the connection.
- * MongoDB was used for heterogeneous collections of content types.
- * Implemented Log4J for logging the information and RabbitMQ for the service of Message broker.
- * Implemented Docker for creating the custom Docker container images.
- * Implemented the Docker for packaging the application in virtual container and then the contained application is deployed to Pivotal Cloud Foundry using CLI (Command Line Interface) tool where service is as Infrastructure.
- * Jenkins was implemented for the Continuous delivery and for Continuous Integration.
- * Used the Junit for unit testing and Selenium for the Integral testing of the application.
- * Implemented the Agile methodology for the efficient and effective completion of project.

Environment: Redux, Angular 2.0, HTML5, CSS3-LESS, Bootstrap, Typescript, ngModules, Arrow function, Angular JS observables, listeners, ES6, React JS, JSX, React.JS, NPM, React JS components, Forms, Events, Keys, Router, Animations, Store-Redux, Reducer-Redux, Webpack, Protractor, Mocha and Chai, Rest Services, Node JS with Express JS, Axios, OpenID Connect (OIDC), OAuth 2.0, JWT, Spring IOC, Java Transaction API (JTA), Spring Boot, Mongoose Connector, MongoDB, Log4J, RabbitMQ, Docker, Pivotal Cloud Foundry, Jenkins, Junit, Selenium and Agile methodology.

Client: Citibank, N.A, Dallas, TX Dec '13 to Aug '14

Role : Sr. Java Developer

Description:

Citi's Worlds Digital bank provides rich support and enhancement features of Internet banking and Mobilebanking. The project is responsible for application development of all NA browser based applications supporting Citicards, Citibank and Citi Mortgage clients. While supporting existing application architecture, the team embarks on new technologies including Responsive WebDesign(RWD), Cloud Infrastructure-as-a-Services(IaaS), Service Oriented Application services and Web API' fully on a path of an agile development technology.

Responsibilities:

- * Designed and developed a Service Oriented Architecture application using SOAP service, REST Service, Angular 1.6, jQuery, HTML5, CSS3, Spring MVC, Cassandra and Oracle Databases.
- * Developed dynamic web pages using HTML5, CSS3(SASS), jQuery and AngularJS 1.6.
- * Implemented jQuery for rendering the multiple views for the web page.
- * Implemented jQuery for auto completion of forms and user-validation functionalities.
- * Refactored and enhanced an existing AngularJS 1.6 application to follow strict MVC patterns for improved source code maintenance which resulted in 40% code reduction, ease of upgrade, scalability, ease of developer spins up and enhanced performance.
- * Implemented Angular JS - Factory for adding some logic before creating the required object.
- * Implemented Angular JS - Service provider function for creating a service object and then adding properties to it by using this keyword and returning the result by using this.
- * Implemented ng-model directive binds the value of HTML controls(input, select, textarea) to application data.
- * Developed responsive design by implementing the Bootstrap custom templates.
- * Bower was implemented for managing the dependencies and Gulp was implemented where files are piped between the tasks.
- * Implemented Spring for IOC (Inversion of Control) container.
- * Implemented Spring IOC for the life cycle management of a bean.
- * Implemented Spring Inversion of Control (IOC) to develop code for obtaining bean references in spring framework using annotations.
- * Separated the Transaction Management from the business code by implementing the Spring Transaction Management approach that is Declarative Transaction Management.
- * The Transactions were then managed by using XML based Configurations.
- * JAX WS API (AXIS-2 framework) was implemented for developing the SOAP services which are protocol specification for exchanging structured information in the implementation of Web Services.
- * Web Service Description Language (WSDL) was implemented for describing the functionality of a web service.
- * Implemented JAX RS API (Apache CXF) for developing RESTful services.
- * Developed a Restful Web Service for catering requests related to transactional data using Apache CXF.
- * Apache CXF is capable of working with Spring AOP interceptors applied to resource classes from Spring.
- * Implemented Spring Template for making HTTP REST calls (Client Side).
- * Bottle micro-framework implemented with RESTful service and used WADL (Web Application Description Language) for modelling the resources provided by the services and relationships between them.
- * Configured various interceptors for the transaction management, Security, Audit logging (database) and logging using Spring AOP.
- * JAX WS Security which is based on User Name and Password was implemented based on the client request and implemented OAuth for authorization.
- * Batch Programming was executed by using Spring Batch which provides reusable functions that are essential for job skip, restart and resource management and implementing Quartz Scheduler which allows to schedule the job by time interval.
- * Implemented Spring Batch for notifying other systems when errors occur via messaging by integrating Spring Integration.
- * Gradle was implemented for building the project and Gradle wrapper was used to obtain the wrapper like structure.
- * Cassandra was implemented for handling the large amount of unstructured data whereas Oracle database for handling the Relational data.
- * Implemented Hibernate for connection between the Spring and Oracle database and CassandraNodeDriver for the connection between the Spring and Cassandra database.
- * Implemented Hibernate Query Language (HQL) for writing queries independent of database.
- * Log4J was implemented for logging in the application.
- * Developed the several EC2 instances and used the EBS(Elastic Bean Stack) for deploying the application to the Amazon Web Service(AWS) cloud.
- * Implemented Java Multithreading in programming to improve overall performance using Singleton design pattern in Hibernate Utility class.
- * Implemented ActiveMQ as the messaging broker service.
- * JUnit was used for testing the unit cases and creating the Mock objects for the purpose of Behavior Driven Development and cucumber for Integral testing.

Environment: Angular 1.6, Factory, Service, ng-model, jQuery, HTML5, CSS3, Bower, Gulp, Spring, Spring IOC, Spring Transaction Management, JAX WS API (AXIS-2 framework), WSDL, JAX RS API (Apache CXF), Spring AOP, JAX WS Security, OAuth, Spring Batch, Quartz Scheduler, Gradle, Gradle

wrapper, Cassandra, Cassandra Node Driver, Oracle, Hibernate, HQL, Log4J, EC2, EBS, AWS, Multithreading, ActiveMQ, Junit and cucumber.

Client:Amica Mutual Insurance, Petaluma, CA Jan '13 to Dec '13
Role : Sr. Java Developer

Description:Amica Mutual Insurance offers auto and property insurance coverage and related services to employer groups, government sponsored plans, and individuals. The project was to develop a web application which is used to process auto and property (home-owner, renter, and condo) insurance. It allows users to run a quote to get their estimates for auto and property, and Purchase policy through online, and to recall quotes.

Responsibilities:

- * Involved in developing the Service Oriented Architecture Application using REST Services, Backbone JS, Handle Bar JS, Ember JS, jQuery, HTML, CSS, Bootstrap.
- * Involved in developing the dynamic web pages by using the HTML, CSS and Bootstrap.
- * Backbone JS and Ember JS were implemented for creating responsive user interface modules.
- * Developed and supported REST API and UI for enterprise-level data migration platforms using Ember JS.
- * Designed and implemented various UI features using Backbone JS, performed routing using Backbone Router.
- * UsedBackbone JSalong with jQuery to utilize its minimal set of data-structuring (models and collections) and userinterface (views and URLs) primitives for building a MVC work flow design and to render the views.
- * Implemented templating Java Scripts for directly rendering the templates by using the Handle bar JS.
- * Implemented Handlebar Java Script libraries for building clean templates based on mustache templating language.
- * Implemented JQuery in developing the Rich Internet Applications (RIA) with AJAX mechanism and JSON as a data exchange tool.
- * NPM was implemented for managing the dependencies.
- * ImplementedSpring IOC for the life cycle management of the beans.
- * Implemented Autowire(@Autowired) formarking a constructor, field, setter method or configurationmethod as to be Autowired by Spring's dependency injection facilities.
- * Spring MVC was used for developing a RESTful web services.
- * Developed web applications using Spring MVC to maintain loose coupling between the layers and injected thenecessary dependent components.
- * Implemented application security using Spring Security for user access management.
- * Implemented Spring AOP (Aspect-Oriented Programming) module to handle cross cutting concerns like transaction management, logging and profiling.
- * Spring Transaction Management is used for the isolation of the transaction ability in our program and to support declarative transaction in the application.
- * Restfulmicro web services were developed using JAX RS API(Apache CXF framework).
- * WADL (Web Application Description Language) was implemented to simplify the reuse of web services that are based on existing HTTP architecture of the web.
- * Maven was used as building tool and for managing the dependencies.
- * DB2was used for handling the Structured dataand iBatiswas used for establishing the connection with the database.
- * Used iBatisframework to automate the mapping between the SQL databases and the Objects in Java.
- * iBatis was used for querying the DB2 data.
- * Implemented IBM MQ for the messaging broker service.
- * Junit was implemented for running the unit test cases and Selenium was implemented for Integral testing.
- * BatchProgrammingwas executed by using SpringBatch which provides reusable functions that are essential for job skip, restart and resource management and implementing QuartzScheduler which allows to schedule the job by time interval.
- * Implemented Java Multi-threading part in back-end component, one thread will be running for each user, which serves that user.
- * The application was deployed to IBM Web Sphere Server.

Environment:Backbone JS, Backbone Router, Handle Bar JS, Ember JS, jQuery, JSON,HTML, CSS, Bootstrap, NPM, Spring IOC (Autowire), Spring MVC, Spring Security, Spring AOP (Aspect-Oriented Programming), Spring Transaction Management, JAX RS API (Apache CXF framework), WADL, Maven, DB2, iBatis, IBM MQ, Junit, Selenium, Spring Batch, Quartz Scheduler, Multi-threading, IBM Web Sphere.

Client:E-Mug Technologies, Hyderabad, India Nov '11 to Nov '12
Role : Associate Software Engineer

Description: E-mug Technologies is one of the finest companies in India developing projects in the Mechanical and Electrical departments. In E-mug I had the opportunity developing their website and creating connections with the Database for exchanging the Information.

Responsibilities:

- * Involved in developing the application using Spring, JSF, Soap Services and JavaScript.
- * Developed UI modifications for portal using XSLT, DHTML, CSS, XML and JavaScript.
- * Implemented JavaScript for client-side validations and for capturing the dynamic events.
- * The application is designed using J2EE design patterns and technologies based on MVC architecture.
- * Implemented Script Closures for associating some data(lexical environment) with a function that operates on that data.
- * Implemented Immediately Invoked Function Expressions (IIFE) for taking advantage of JavaScript's scope rules to funnel in certain global variables and avoid creating too many of our own globals.
- * JSF (Java Server Faces) with prime faceslibrary which was powerful MVC framework were implemented for building component-based UI.
- * Build web application using JSF Prime Faces by assembling reusable UI components in a page and wiring the client generated events to server-side event handlers.
- * Installed WebSphere Application Web Server for handling HTTP Request/Response. The request and response from the client are controlled using Session Tracking in JSP.
- * Developed EJB's and deployed them on IBM WebSphere Application server.
- * EJB Session Beans were used to process request from user interface using OSS.
- * Developed and implemented the MVC Architectural pattern using STRUTS framework including JSP, Servlets, and EJB.
- * Implemented SOAP services for exchanging structured information in the implementation of web services.
- * Implemented Spring Security for authentication and authorization for the application.
- * Used ANT build tool for compilation and generation of war file.
- * Loading data from Text Files to Database by using SQL Loader.
- * Implemented Hibernate for mapping the java classes with the database tables.
- * Hibernate is used for data query and retrieval facilities.
- * HQL (Hibernate Query Language) was used for writing the database independent queries.
- * Developed stored procedures, triggers in MySQL for lowering traffic between servers & clients.
- * Implemented Spring Batch for executing the batch programming using shell scripts and commands passed through Unix command line.
- * Performance tuning of JVM heap size, garbage collections, java stack and Native thread & production performance.

Environment: XSLT, DHTML, CSS, XML, Java Script, Script Closures, IIFE, JSF, prime faces, IBM WebSphere, EJB, OSS, STRUTS, SOAP services, Spring Security, ANT, SQL Loader, Hibernate, HQL, MySQL, Spring Batch, shell scripts, JVM and garbage collections.

Client: Clarion Ad media, Hyderabad, India June'10 to Nov '11

Role : Jr. Java Developer

Description: Clarion is a collaborative advertising agency for both media and web based. Providing web applications for both web advertising and providing applications for client to interact with their customers.

Responsibilities:

- * Involved in developing the Monolithic architecture application using EJB, Struts, jQuery, JSP by following the Waterfall Methodology.
- * Implemented the EJB and Struts Framework for developing the Form and Action Classes.
- * Prepared EJB deployment descriptors using XML and Used JAXB components for transferring the objects between the application and the database.
- * Dynamics from generation auto completion of forms and user-validation functionalities using jQuery.
- * Configured Struts-config.xml, Tiles-def.xml and properties file provided by Struts Framework for the implemented modules.
- * Implemented connectivity to databases using JDBC from servlets and JSP through Enterprise Java Beans (EJB).
- * Developed Web pages using JSP and implemented Struts Validation framework for form input validations.
- * Build PL/SQL functions, stored procedures, views with JDBC data source with connection pooling enabled.
- * Implemented JavaScript Library for simplifying the HTML document traversing, and event handling.
- * The application is deployed to JBoss Application Server.

Environment: EJB, Struts, XML, jQuery, JDBC, Servlets, JSP, PL/SQL, HTML, Java Script and JBoss.