Amanaganti Shiva Kumar

PROFESSIONAL SUMMARY:

- Having 9.5**+ Years** of experience in all the phases of **Software Development Life Cycle** (**SDLC**) including **Requirement Analysis**, **Design**, **Implementation** and **Product Testing** in both the Web based and Enterprise applications with emphasis on Object Oriented, Java/J2EE and Client Server technologies.
- Strong experience with Software methodologies like **Agile, Waterfall** and **Test-Driven Development**.
- Experience in Front-End **UI** technologies like **HTML5, CSS3, Bootstrap, JavaScript, jQuery, Ajax, AngularJS, React JS.**
- Having Good experience in developing and migrating existing application into cloud-based platforms like **Amazon Web Services (AWS).**
- Hands on experience working with various Amazon Web Services like **EC2, SQS, S3, Elastic Search** and **Dynamo DB** through **AWS Console** and **API Integration.**
- Hands on experience in developing and solid understanding of **Micro-Services**, **Service Oriented Architecture (SOA).**
- Experience with **Java 1.8 features** like **parallel streams** and **filters** through **lambda expressions**.
- Experience in developing web-based enterprise applications using **Java, J2EE, Servlets, JSP, EJB, JDBC, Hibernate, Spring IOC, Spring AOP, Spring MVC, Spring Web Flow, Spring Boot, Spring Security, Spring Batch, Spring Integration, Web Services (SOAP and REST) and ORM frameworks like Hibernate.**
- Experience with **design patterns** like **Singleton, Factory Pattern, Prototype, Business Delegate, Session Facade, Service locator, Observer, Decorator, Front Controller patterns.**
- Expert in **Core** Java with strong understanding of **Garbage Collector, Collections, Multithreading, Event handling, Exception handling** and **Generics.**
- Strong experience in developing and consuming **SOAP** and **Restful** based **Web Services.**
- Migrated the production **PostgreSQL** schema to the new **AWS RDS** Aurora instance.
- Expertise in **XML** technologies like **DTD, XSD, XSLT** and various parsers like **DOM, SAX and JAXB.**
- Expertise in using **JDBC API** and **Hibernate** for connection and access to databases.
- Strong Experience in database design using **PL/SQL** to write Stored Procedures, Functions, Triggers, Indexers and proficiency in writing complex queries, using **Oracle, IDM DB2, SQL Server, MySQL, PostgreSQL, MongoDB** and **Cassandra.**
- Proficient in writing complex queries using SQL Joins, Stored Procedures, Views and Indexes.
- Experience with **JUnit, Mockito and Log4j** in writing unit tests and determining application functionality.
- Hands on experience working with various Application servers like **JBOSS**, **Web Sphere** and **Apache Tomcat.**
- Experience in different **IDE**s such as **Eclipse, RAD**, **Spring Tool Suite (STS), and IntelliJ.**
- Having measurable experience on application building tools like **Maven** and **ANT.**
- Experience in development, deployment and troubleshooting Web based and enterprise-based applications on **Windows, UNIX** and **Linux platforms.**
- Experience in using code repository tools like **Tortoise SVN, GitHub and Visual Source Safe.**
- Experience with adhering to strict **coding standards** in a team development environment.
- Strong communication and analytical skills and a demonstrated ability to handle multiple tasks as well as to work independently or in a team environment.
- A team player and self-motivator possessing excellent **analytical**, **communication**, **problem-solving**, decision-making and Organizational skills.

TECHNICAL SKILLS:

Programming Languages	C, C++, Java, J2EE, SQL and PL/SQL.
Platforms	Windows, Linux and Unix.
J2SE/J2EE Technologies	Java, J2EE, Servlets, JSP, JMS, JavaBeans, JSTL, JSF, Struts, EJB, Spring, Hibernate, JTA, JNDI, JPA JMS, Web Services SOAP (JAX-RPC, JAX-WS), Restful (JAX-RS), WSDL and UDDI.
Web Technologies	HTML, CSS, JavaScript, jQuery, Ajax, Angular6/7, React JS and Bootstrap,
XML Technologies	XML, XSL, XPATH, DOM, SAX, DTD, XQuery, XSD and XSLT.

Cloud Technologies	Amazon Web Services
Databases	Oracle, My SQL, SQL Server, IBM DB2, Mongo DB and Dynamo DB.
Web/Application Servers	Apache Tomcat, IBM Web Sphere, Web logic and JBOSS.
Build Tools	ANT, Maven and Jenkins
IDE / Tools	Eclipse, IntelliJ, Spring Tool Suite (STS) and RAD.
Testing Tools/ Others	Selenium, JUnit, Mockito, Soap UI, Log4j.
Version Control	Tortoise SVN, CVS and GIT.
Methodologies	Agile, Waterfall and Test-Driven Development

PROFESSIONAL EXPERIENCE:

Client: Paychex, Inc. Rochester, New York **Dec’2024 - Till Date**
Role: Java Developer
Description: **Paychex** has a product which is **AQG** (Automated Quality Gataes), going beyond build time to enable automated quality inspections and enforcement at any point in the SDLC pipeline, which allows greater flexibility in configuration, monitoring and better performance. The platforms host test automation, security scanning platforms like **SAST** and **SCA** to prevent vulnerabilities in code and prevent risks associated to user solutions.

Roles &Responsibilities:

- Actively involved in Analysis, Design, Development, System Testing and User Acceptance Testing. Successfully followed Sprint planning and estimation for user stories in **Agile Environment**.
- Developed the application using **Java17/21** and implemented its features like **lambdas Expressions, Record, Time API, Streams, functional interfaces, collectors, default methods, type interfaces, for each**
- Developed **RESTfulAPIs** using **Kotlin**, **Java17/21**, and the **Spring** framework.
- Created diff type of **vulnerabilities** like critical, high, medium, low and created endpoints based on the requirement using **Java**, **Kotlinspring**.
- Used **Kafka** for and created diff topics like **SAST** flow tracking, **SAST** vulnerability topic, with connecting to **AQG**service.
- Wrote decoupled and testable code using Mockito and JUnit for unit testing in Java and Kotlin.
- Designed and developed **Micro Services** business components, **Restful service** endpoints using **Spring Boot**.
- Successfully spun up the Satellite service to validate **Kafka** topic connectivity with the target service.
- Building a Command line tool to interact with **RESTfulAPI** using in **Golang**.
- Utilized **Kafka** as a **messagequeue** for reliable, asynchronous communication between **Microservices**.
- Executed manual functional and comprehensive **regression** test scripts to test the **NCFAST** system.
- Used **Microservices** with **SpringBoot** based **servicesREST** and **ApacheKafkamessage** brokers.
- Migrated **SCAscans** to **SAST** scans by transitioning from on-premise infrastructure to **Checkmarx**, leveraging **Java**, **Spring** and **Kotlin** technologies.
- Deployed the application and conducted testing across multiple environments—including **N2A**, **Preflight** and **Production**—using **OpenShift**.
- Deployed **Docker** contained **Spring boot** micro services into **AWS EC2** container service using **AWS** admin console and used lambda to run applications in **AWS** console.
- Design and Develop, Maintain and Execute Release Regression Automation Scripts.
- Created pods and configured services in **RedHatOpenShift**, aligning deployment with available data centers such as **ODC** (**Omaha**), **WDC** (**Webster**) and **HDC** (**Henrietta**).
- Managed **Amazon Web Services (AWS)** like **EC2, S3 bucket, ELB, Auto- Scaling, AMI, IAM** through **AWS Console** and **API Integration**.
- Authored various types of test cases—including **TAF**, **PXT**, **sanitychecks** and **JUnit** tests—to validate each functionality.

- Transitioned legacy **TAF** test suites to **PXT framework** using **Golang**, improving test reliability and integration with modern **CI**/**CD pipelines**.
- Deployed, Scaled, Configured, wrote manifest file for various **Microservices** in **AWS**.
- Developed the test cases with **JUnit** and **Mockito** to test the application functionalities and **Log4j** for logging purposes
- Developed custom **endpoints** using **Go** (**Golang**) to support application-specific functionality.
- Stored data in **MongoDB** and accessed it using **SQL queries** through **DLPHX** (**Toad DB**).
- Created comprehensive wiki pages documenting each developed task and functionality, and presented them to stakeholders through live Demos.
- Used Docker to generate various types of summary reports in **build.txt** files using Golang**.**
- Provided on-call and production support on a rotational basis, addressing customer and end-user issues by delivering effective solutions.
- Responsible for **Continuous Integration (CI)** and **Continuous Delivery (CD)** process implementation using **Jenkins** on **Kubernetes** environment along with **UNIX** Shell scripts to automate routine jobs.
- Used **Git** to maintain the version of the files and took the responsibility to do the code merges from branch to master and creating new branch when new feature implementation starts and **IntelliJ** as **IDE**
- Collaborated with a cross-platform team to support and integrate the application across diverse environments.

Environment: Agile, Java 21, Spring, Spring Boot, RESTful services, Micro Services, Open Shift, Docker, Kubernetes, DLPHX, Mongo DB, PXT, TAF, JUnit, Jenkins, Git, Checkmarx, Kotlin, Red Hat, Go-Lang, Jenkins, UNIX, AWS, Mockito, SQL and IntelliJ.

Client: ADP, Alpharetta, GA **March 2022 – Nov 2024**
Role: Java Developer
Description: ADP has 4-5 versions of time management software and building Tau to be the Next Gen version of Time Management. The product is to publish schedules and look at the hourly employees they have available based on policies and PTO. To accelerate the manager\'s ability to schedule from doing manually to automatically and helps the employees find more easily who they can swap schedules with.

Roles &Responsibilities:

- Involved in all phases of the **Software development life cycle (SDLC)** using **Agile Methodology.**
- Developed User Interface using **HTML5, CSS3, Bootstrap, JQuery** and **Angular JS.**
- Created responsive web design using **Bootstrap** and Consumed **Restful Web Services** with **Angular 4/6 $httpservices.**
- Designed and Implemented the application using various **Spring Framework** modules **Spring IOC, Spring MVC, Spring AOP, Spring Security, Spring Boot, Spring Web Flow, Spring Data, Spring Cloud, Spring Integration** and **Hibernate.**
- Used **Spring IOC** for dependency injection and make the application loosely coupled and **Spring AOP** for logging purposes.
- Implemented the authentication, authorization and access-control features by using **Spring Security** and **LDAP.**
- Implemented **Restful** and **SOAP** based web services using **JAX-RS** and **JAX-WS** respectively.
- Utilized **Java 11** features like **Lambda** expressions for collection evaluation and comparing the data, **Stream API** for Bulk data operations on Collections which would increase the performance of the Application, Parallel operations on collections for effective sorting mechanisms and to change the sequential stream flow of threads while session establishment.
- **Micro Services** were being used as a strategy to gradually replace a core monolithic application while still adding business features.
- Designed and developed **Micro Services** business components and **Restful service** endpoints using **Spring Boot.**
- Implemented **Restful** Web Services to retrieve data from client side using **Micro Services** architecture.
- Developed **Micro Service** to provide **Restful API** utilizing **Spring Boot** with **Spring MVC.**
- Managed **Amazon Web Services (AWS)** like **EC2, S3 bucket, ELB, Auto- Scaling, AMI, IAM** through **AWS Console** and **API Integration.**
- Used **Docker** to containerize the Services and **APIs** to run on **EC2** instances and Implemented the build stage to build the **Micro Service** and push the **Docker Container** image to the private Docker registry.
- Worked with **Docker** and **Kubernetes** to deploy **Microservices** in modern containers to increase Isolation

- 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 multithreading in programming to improve overall performance using **Singleton** design pattern in Hibernate Utility class.
- Developed the test cases with **JUnit** and **Mockito** framework to test the application functionalities and **Log4j** for logging purposes.
- Used **Gradle** as a build automation tool and deployed the application on **Web Sphere Application Server.**
- Worked on developing **Restful** endpoints to cache application specific data in in-memory data clusters like **REDIS** and exposed them with **Restful** endpoints.
- Responsible for **Continuous Integration (CI)** and **Continuous Delivery (CD)** process implementation using **Jenkins** along with **UNIX** Shell scripts to automate routine jobs.
- Used **Git** to maintain the version of the files and took the responsibility to do the code merges from branch to master and creating new branch when new feature implementation starts and **IntelliJ** as **IDE.**

Environment: Java 11, J2EE, HTML5, CSS3, JQuery, Angular 4/6, Spring IOC, Spring MVC, Spring AOP, Spring Security, Spring Boot, Spring Data, RESTful services, SOAP, Micro Services Architecture, Amazon Web Services, Docker, Kubernetes, Hibernate, JDBC, Oracle, PostgreSQL, Mongo DB, JUnit, Mockito, Gradle, Redis, Web Sphere Application Server, Jenkins, Git and IntelliJ.

Client: New York State Department of Environmental Conservation, NY **Feb'2021–Feb'2022**
Role: Full Stack Java Developer
Description: The Department of Environmental Conservation(DEC), in conjunction with the Department of Health(DOH), is to create and maintain a geographic information system and associated data storage and data analytics for purposes of sharing all relevant water quality and drinking water quality data among state and local agencies and analyzing and solving water quality problems statewide. The goal is to collaboratively develop a high-level approach to create a data management system envisioned in the enacted legislation

Roles &Responsibilities:

- Involved in various phases of **Software Development Life Cycle (SDLC)** as requirement gathering, modeling, analysis, architecture design & development and worked with the business team to review the requirements.
- Developed User interface using **HTML5, CSS3, Bootstrap, jQuery, JavaScript, Ajax, ExpressJS, NodeJS** and **ReactJS.**
- Created responsive web design using **Bootstrap** and Consumed **Restful Web Services** with **ReactJS $HTTP services.**
- Developed complex **Multithreaded** Java applications for processing large volumes of data to achieve Multitasking
- Developed server-side presentation layer using **Struts MVC** Framework.
- Used **Java 1.8Lambda** expressions along with **Streams** for creating internal iteration and performing chain operations such as **Filter, Map, Collect on a collection**.
- Used **Spring MVC** framework at the Web tier level to isolate each layer of the application so that complexity of integration will be reduced, and maintenance will be very easy.
- Used **Spring AOP** for solving crosscutting concerns like keeping customer log data and transactions details etc.
- Used **Spring Batch** for processing large amount of data like transaction management, job processing, resource management and logging.
- Secured the API's by implementing **Oauth2** token-based authentication/authorization scheme using **spring security**.
- Developed the persistence layer using **Hibernate Framework** by configuring the various mappings in hibernate files and created **DAO** layer.
- Involved in writing **SQL** and **Stored Procedures** for handling complex queries with help of **TOAD** and access them through Java Programs from **Oracle Database**.
- Developed **REST** API's using **Spring MVC** and **Spring boot**, hosted all **micro services** on **AWS** and used **Elastic Beanstalk** to setup application and configured environments.
- Designed and developed **Micro Services Architecture** to divide application into business components using **Spring Boot**.
- Worked with **JUnit** Regression Test Framework to implement unit testing, to accelerate programming speed and to increase the quality of code.
- Worked on setting up **Maven** scripts to build, package, and deploy application code to the target **Apache Tomcat** Servers and worked on continuous integration servers like **Jenkins**.
- Developed a production-level cloud-based architecture in **AWS**, including creating machine Images like **AMI**.

- Involved in creating **EC2** instances and installed required configurations and applications on it and created **S3** buckets for storing object level data into it.
- Involved in creating resources stack on **AWS EC2** using **Cloud Formation** by writing **Template** files.
- Used **Docker** for creating **Docker images** for launching containers on **AWS EC2** and Expert in using **Docker** for Environment provision solution.
- Worked on integrating **Spring** with the **Elasticsearch** and developed API's to read and publish data from **Elasticsearch** cluster using native Java transport client as well as **REST** client.
- Worked on developing **Restful** endpoints to cache application specific data in in-memory data clusters like **REDIS** and exposed them with **Restful** endpoints.
- Used **Swagger** specification for documenting **REST** API's which exposes description of a **RESTFUL** web services to different clients.
- Worked in **Agile TDD** environment using SDLC tools like **SVN, Jira, Jenkins**, **Maven** for build packaging and deployments of the application components.
- Used **Spring Tool Suite (STS)** as an IDE for Application Development

Environment: Agile Methodology, Java 1.8, HTML5, CSS3, JavaScript, React JS, STS, J2EE, Hibernate, Spring framework modules, JSP, Oracle, MVC, Jira, Jenkins, JUnit, Maven, JAX-RS, Log4J, Apache Tomcat, Spring Boot, Micro services, AWS, Elastic Bean stack, Spring Batch, Docker, Elastic search, STS, Tortoise SVN.

Client: Ally Bank,Detroit, MI **Jan'2020–Dec'2020**
Role: Full Stack Java Developer
Description: Ally Bank offers a broad array of financial, banking services etc. The system developed provides personal savings details of Ally bank Customers. It offers a broad array of financial clients and as per the users account status offers him to take up the new schemes enrolling with the bank. Upon the user's request it generates the reports of user details and submits it to the user. Advertises other mutual benefits and keeps track of the contact details of the users and updates them on an event if requested.

Roles &Responsibilities:

- Participated in all the phases of **SDLC** including Requirements gathering, Design, Analysis of the Customer Specifications, Development and Customization of the application.
- Adapted **Agile Methodology** from specification, **Analysis**, **Design**, **System integration**, **Testing**, **Deployment** and **Maintenance**.
- Involved in daily standup meetings to provide regular updates on individual tasks and progress of user stories.
- Developed UI components using **Angular JS**, **Node JS**, **JavaScript**, **JSP**, **HTML5** tags, **CSS3**, **AJAX** and **jQuery.**
- Used **RAD** as IDE, configured and deployed the application onto **Tomcat** application server using **Maven** build scripts to automate the build and deployment process.
- Developed user-friendly interface to seamlessly combine the new module with existing login system by using **Angular JS.**
- Good Expertise in **Stream** of API for **Bulk Data** Operations on Collections and API improvements in **Java 1.8.**
- Maintained **Interface** compatibility and **concurrency** in the project using **Java 1.8** new features like **Lambdaexpressions**,**Default**, **Static methods** and **Concurrency** API.
- Designed and developed various modules of the application with **J2EE** design architecture and frameworks like **Spring MVC** architecture and Spring Bean Factory using **IOC**, **AOP** concepts.
- Used **REST** methodology using **HTTP Handlers** in developing **Webservice** replication interface for setting up data feeds for content transportation in **XML** and **JSON** formats.
- Developed **REST** API's using **Spring MVC** and **Spring boot**, hosted all **micro services** on **AWS** and used **Elastic Beanstalk** to setup application and configured environments.
- Designed and developed **Micro services**, using Java, **Spring**, **and REST**.
- Involved in integrating **JAX-RS** into **Spring Boot** application.
- Implemented the various services using **Micro Services** architecture in which the services work independently.
- Experience in **Docker** engine and Docker Machine environments, to deploy the **micro services**-oriented environments for scalable applications.
- Achieved synchronization of multiple threads through **Multithreading** and **Exception Handling.**
- Developed **interfaces/APIs** in Java to interact with various backend systems.
- Created web services using **Spring @Rest Controller** to return **JSON** to the frontend.

- Involved in writing **Spring Configuration** XML, file that contains declarations and business Classes are wired-up to the frontend managed beans using **Spring IOC** pattern.
- Implemented the application using **Spring IOC**, **Spring MVC Framework**, **Spring Batch** and handled the security using **Spring Security.**
- Used **Multithreading** in programming to improve overall performance using **Singleton** design pattern.
- Implemented and maintained monitoring and alerting of production and corporate servers such as **EC2** and storage such as **S3** buckets using **AWS.**
- Create and configured the continuous delivery pipelines for deploying **Micro services** using **Jenkins** CI server.
- Used **AmazonS3** to backup database instances periodically to save snapshots of data.
- Implemented service methods and SQL queries, **PL/SQL** to interact with the **Oracle** DBMS and worked with **Cassandra.**
- Deployed **Docker** contained Spring boot micro services into **AWS EC2 container** service using **AWS admin console** and used **lambda** to run applications in **AWS console**.
- Worked on **Elastic search** to convert raw data such as log files or message files into internal documents and stored them in a basic data structure like a JSON object.
- Implemented **log4j** logging framework by creating various **Splunk** monitoring tools dashboards for easy t racking and debugging.
- Used **JIRA** extensively to log all issues and track development status.
- Developed unit testing framework using **JUnit** test cases for continuous integration testing.
- Developed the build scripts using **Maven.**
- Project maintenance is done in **Git, Bitbucket** and related documentation in **Confluence**.

Environment: Agile methodology, Core Java, Java/J2EE, JDK 1.8, Spring MVC, Spring Boot, Spring Batch, Spring Security, Tomcat, Agile methodology, RESTful Web Services, JSP, MySQL, Cassandra, HTML5, CSS3, JavaScript, jQuery, AJAX, RabbitMQ, Elastic Search, AWS EC2, S3, Micro Services, Bamboo, Bitbucket, Angular JS, Node JS, Bootstrap, Splunk, Maven, Oracle, SQL, PL/SQL, JIRA, RAD, Log4j, JUnit, Git.

Client: ChangeHealthcare, Chicago, IL **April'2019 – Dec'2019**
Role: Full Stack Developer
Description: Post Pay (audit and Recovery services) helps to maximize savings by identifying and recovering lost dollars due to improper payments. Equiclaim identifies inaccurate repricing, improper coding, and misapplied reimbursements, and recovering over payments. Every claim that is paid by the payer will be retrospectively audited for possibility of identifying savings under the perspective of incorrect DRG Coding (DRG service), incorrect Hospital Billing (HBA service), RXI audits (RXI service) and high cost medical equipment (CCA service).

Roles &Responsibilities:

- Involved in the complete **SDLC (software development life cycle)** of the application from requirement analysis to testing.
- Followed **Agile Methodology** in analyze, define, and document the application, which will support functional and business requirements. Coordinate these efforts with Functional Architects.
- Extensively worked on **Responsive Web Design (RWD)** page development **using HTML5, CSS3, JQuery, JavaScript, AngularJS, and JSON.**
- Used **Java 1.8** features in developing the code like **Lambda** expressions, creating resource classes, fetching documents from database.
- **Designed, Configured** and **deployed Amazon Web Services (AWS)** for a multitude of applications utilizing the **Amazon cloud formation.**
- Experienced in setting up **AmazonEC2** instances, virtual private cloud (VPCs), and security groups. Setting up databases in **AWS** using **RDS**, storage using **S3** bucket and configuring instance backups to **S3** bucket.
- Implemented the application using **Spring IOC, Spring MVC Framework, Spring Batch, Spring Boot** and handled the security using **Spring Security.**
- Developed **RESTful API** for assessment indicators module and token based authentication for **RESTful** services using **Spring Security**.
- Involved in building database Model, APIs and Views utilizing **Python**, to build an interactive web based solution.
- Used **Jersey** to implement **Restful Web Service** and used **XML** form to transfer the data.

- Developed the application using **J2EE Design Patterns** like Business **Delegate, Factory, Singleton, Session Facade, Service Locator** and **DAO.**
- Worked and involved in deployment of core platform technologies, techniques, and web app frameworks such as **spring, JBOSS, Tomcat, JSON, XML, HTML5, and Web Services**.
- Used **Spring MVC module** to develop **MVC Architecture**.
- Worked on **Apache Camel, RabbitMQ** for sending messages over queue.
- Deployed the Application into **Docker** container and made easily accessible at runtime using Cloud Foundry and other cloud services like **AWS**, **Netflix Eureka**.
- Used **Spring Security and OAuth2.0** for **Authentication and Authorization** of the application.
- Implemented **Spring AOP** for declarative transaction management.
- Developed **AWSLambda** function to send a request for internal service end points through **API Gateway** using Apache HTTP Client.
- Managed code versioning with **GitHub** and deployment to staging and production servers
- Used **Rest Controllers** to replace the existing **operations layer**, which acts as bridge between **business layer** and the **UI**.
- Used **Jenkins** as build management tool for continuous integration process.
- Configured **pom.xml** to deploy the project using **Maven.**
- Provided Technical support for production environments resolving the **issues, analyzing** the **defects,** providing and implementing the solution defects.

Environment: Agile Methodology, Java1.8, Lambda, J2EE, HTML5, CSS3, JavaScript, JQuery, AJAX, AngularJS, Bootstrap, JSON, JSP, AWS, Micro Services, Oracle, RabbitMQ, Spring AOP, Hibernate, Spring Cloud, Spring MVC, Spring JDBC, JDBC, Web-Services, SOA (Service-oriented) Architecture, Redis, REST, JAX-RS, Jersey, JUnit, JAX-B, WebSphere, JIRA, Maven, GIT, RAD, Jenkins, MongoDB, Cassandra,

Client: Cognizant Technology Solutions, Chennai, India **Dec'2017 – Dec'2018**
Role: Java Developer
Roles &Responsibilities

- Worked on Java as a developer for the company entity management system with Spring MVC and Clients Itension Framework.
- Implemented business logic using JAVA, SPRING, HIBERNATE.
- Developed SQL queries using My SQL and SQL server for Impoter operation to import the data from external sources into DB according to the requirement.
- Worked on JDBC for connecting, updating and process the database of the Employee portal account.
- Writing the JUNIT testcases for the implemented code and checked that it covers all the requirements.
- Using Polarion software, able to track the defects, fix them accordingly by making necessary changes and fix the bugs.

Environment: Java, J2EE, JSP, Servlets, JDBC, HTML, CSS, JavaScript, AJAX, Eclipse, JUnit, MVC, Apache Tomcat and MySQL.

Client: TekStrom Inc., Bangalore, India **Oct'2015 – Nov'2017**
Role: Jr. Java Developer
Description: The objective of this project is to implement a web based front end system which can be used by the employees, managers and higher-level management having specific role permissions for effective processing of day to day business activities in the company like leave management, project management and resource management etc.

Roles &Responsibilities:

- Developed the application using **Struts** Framework that leverages classical Model View Controller (MVC) architecture.
- Designed the user interfaces using **JSP**s, developed custom tags, and used **JSTL Taglib**.
- Developed various java business classes for handling different functions.
- Developed controller classes using **Struts** and tiles Api
- Involved in Documentation and Use case design using UML modeling include development of Class diagrams, Sequence diagrams, and Use case Transaction diagrams.
- Participated in design and code reviews

- Developed User Interface using **AJAX in JSP**and also performed client-side validation
- Developed **JUnit** test cases for all the developed modules. Used SVN as version control

Environment: Java, J2EE, JSP, Struts, JNDI, DB2, HTML, XML, DOM, SAX, ANT, AJAX, Rational Rose, Eclipse, SOAP, Apache Tomcat, Oracle, LOG4J, SVN.