**NITHIN PONUGOTI**

**Sr. Java Full Stack Developer**

**Email:** [nithinponugoti90@gmail.com](mailto:nithinponugoti90@gmail.com)

**Phone:** 682-237-9195

**PROFESSIONAL SUMMARY:**

* Over **8+ years** of prevalent software experience in **design, development** and **deployment** of **web-based Client-Server** business applications using **OOP, Java/J2EE technologies.**
* Experience working with different **SDLC** (Software Development Life Cycle) methodologies like **AGILE (SCRUM), TDD (Test Driven Development)** and **Waterfall Models**.
* Proficient in applying design patterns like **MVC, Singleton, Session Facade, Service Locator, Visitor, Observer, Decorator, Front Controller, Data Access Object.**
* Experience in developing applications using **Java/J2EE technologies like Servlets, Struts, accustom tags, Java Beans, EJB, Spring, Spring Boot, JDBC, Hibernate, JBPM, JMS, JSF.**
* Experience on Spring Framework - **Spring** **Core, Spring DAO, Spring IOC, Spring MVC, Spring ORM, Spring AOP, Spring Boot**, **Spring Web flow** and hands on experience in working with **Spring Batch** for monitoring, logging into the application, tracing, application support, transaction and resource management.
* Expertise in UI technologies like **XML, XML Spy, HTML, HTML5, DHTML, XHTML, CSS, CSS3, PHP, XSLT, JSON, jQuery, AJAX, JavaScript, ReactJS, BackboneJS.**
* Developed modules using Angular,Node,React, Bootstrap, JavaScript, Ajax, query, CSS3 and HTML5.
* Good command and knowledge in working with core java concepts like **Collection frameworks**, **Interfaces, Synchronization,** **Serialization, Thread pools**, **JavaBeans**, **Externalization**, **Generics** and **Annotations**.
* Good command on **Java 8** concepts like **Lambda, Functional Interfaces, Filters** and **Streams.**
* Knowledge of application development for Cloud platforms using technologies like **Java/J2EE, Spring Boot, Spring Cloud, Microservices.**
* Expertise in implementing client-server business applications using **Microservices**.
* Experience in configuration and deployment of multi-tier applications using different web servers like **Apache Tomcat, WebSphere, Apache Axis, Apache HTTP Server, JBoss** and **WebLogic.**
* Hands-on experience in implementation and developing using **RDBMS** like **MySQL, DB2, Oracle.**
* Experience in database design and development of **PL/SQL** stored procedures, and packages in relational databases like **Oracle 9i / 10g / 11g, SQL Server 2005/2008, DB2, MySQL** and also good hands on experience in implementing, designing and developing **NoSQL** Databases like **MongoDB, Apache Cassandra** and **CouchDB.**
* Expertise in developing **SOA** **(Service Oriented Architecture), REST** and **J2EE** Web Services based on **SOAP, WSDL, UDDI, JAX-WS, JAXB, JAX-RPC, JAX-RS** using tools like **Apache Axis.**
* Experience on **AWS** platforms **VPC, EC2, ELB, SQ3,** and **EBS, ROUTE 53, Dynamo DB CLI** and **SDK’s/API tools** and good knowledge on **AWS** tools like **S3, Lambda, Kinesis, Cloud Watch, SQS.**
* Extensive hands on experience in **Pivotal Cloud Foundry (PCF)** and other **PaaS** platforms.
* Experience in AWS infrastructure with **AWS** **cloud** environments such as **Jenkins, Chef** and **Puppet**.
* Experience in using **Jenkins** for **CI/CD** process and creating **Docker** images and containers.
* Experience in working with different IDE’s like **Eclipse, Rational Application Developer (RAD), NetBeans, IntelliJ, STS (Spring Tool Suite), Visual Studio, Dream Viewer.**
* Experience with project build/test/deployment tools like **Maven** and **Ant.**
* Experience in using Automation tools like **Selenium** and **Cucumber**.
* Experience in using code repository tools - **Tortoise SVN, CVS, GitHub**, and **Visual Source Safe**.
* Worked with **Log4j** for logging and used **JUnit** and **Mockito** for unit testing and integration testing.

**TECHNICAL SKILLS:**

|  |  |
| --- | --- |
| **Programming Languages** | Java 8.0/7.0/6.0/5.0, J2SE, J2EE, C, C++, PL/SQL. |
| **Databases** | Oracle 9i/10g/11g, DB2, MySQL 5.1, SQL Server, NoSQL, Cassandra,  Mongo DB. |
| **Java/J2EE Technologies** | JSP, JSTL, JSF, Struts, Hibernate, Spring Security, JNDI. |
| **Technologies** | Servlets, JDBC, EJB, NodeJS, ReactJS, Bootstrap. |
| **Web Technologies** | HTML5, XML, CSS3, JavaScript. JQuery, Angular2/4, Ajax, JSON |
| **Cloud Environments** | AWS, Netflix Eureka, Pivotal Cloud Foundry. |
| **Web services** | Restful API, Microservices, JAX-RS, JAX\_WS, JAX-RPC, SOAP, WSDL |
| **Spring Frameworks** | Spring MVC, IOC, AOP, JTA, Spring Dashboard, Spring Boot, Spring  Microservices, Spring REST, Spring Netflix, Spring Zuul Proxy. |
| **IDE** | Eclipse, Net Beans, Visual Studio, STS, Brackets, IntelliJ, RAD. |
| **Version Control** | Git, CVS, SVN. |
| **Testing tool** | JUnit, Mockito |
| **Build Tools** | Gradle, ANT, Maven, Jenkins. |

**PROFESSIONAL EXPERIENCE:**

**Client: PNC Bank - Pittsburgh, PA Duration: Jul 2016 to Present**

**Role: Sr. Java Full Stack Developer.**

**Description:**

PNC Bank is an American multinational financial bank with operations around the world. Loan Application

Processing System of PNC Bank to keep track of all loan applications received at various branches till the applications are approved or rejected. The system provides two different user interfaces, one for the customer service department to enter information online. The second user interface is used by the loan processing officer to check the applications, the details of the customer so that loan can be granted.

**Roles & Responsibilities:**

* Involved in development and testing of Streamlets module. Actively involved in Analysis, Design, Development, System Testing and User Acceptance Testing. Followed **Agile** methodology in the **Scrum** Cycle model.
* Implemented **Java8** features like **Lambda expressions, Streams, filters, pipelines, Optional Interfaces etc.**
* Developed **UI** pages using **HTML5, CSS3, JavaScript** and used **Boot Strap** for creating responsive Web pages.
* Implemented UI components using **Angular 2.0** dependency Injection, Models, data binding and controllers.
* Developed custom **Angular directives**, **components**, **services**, t**emplates** and integrated with **Spring forms**.
* Used **J2EE** design patterns like **Adapter, Service locator, Session facade, Singleton, Factory and DAO**.
* Used **Executor Service** framework to implement multithreading in programming to improve overall performance.
* Used **JMS** for reliable and asynchronous exchange of important information such as payment status report.
* Developed classes using **Core java/Executor Service** **(multithreading, concurrency, memory management).**
* Involved in developing **UI** pages using **Angular 2.0** and calling the **Rest API’s** from backend.
* Created **Spring Boot** with Micro Services utilizing **RESTful API**.
* Used NodeJS for doing non-blocking design operation for server-side code.
* Built REST by building Nodesin the back-end to handle requests sent from the query Ajax calls.
* Consumed **REST** Web Services for getting the credit card information from third party application using **EME**.
* Developed **Restful Web services** for other systems to interact with our system and added **WS-Security** using **XWSS** to secure the **Web services**.
* Implemented the associated business modules integration using **Spring** and **Hibernate** data mapping.
* Implemented **Hibernate** **Template** to great extent making use of **HQL**, **Named Parameters, Named Queries** and interfaces such as **Expression, Query, Criteria.**
* Developed **Session Beans** for Transactions like fetching the data and updating the customer information.
* Worked on **Drools Rules Engine** and **Expectation Rule Engine** for writing Business Rule Validation.
* Utilized **Spring Core, AOP** and **WS-security** to build robust business operations with transactional support.
* Implemented **Spring** **Cloud Config Server, Routing** and **Filtering** to **Microservices** using **Netflix Zuul.**
* Extensively used **Spring Batch** which uses **Spring IOC** dependencies, this is useful for developing batch processing.
* Developed unit and integration tests for existing **microservices** using **JUnit** and **Docker** environments.
* Developed database objects-**Stored Procedures, Tables, Views, Sequences, Cursors, Packages and Triggers.**
* Wrote **SQL** queries using **PL/SQL**, stored procedures and enhanced performance by running explain plans.
* Involved in **CI/CD** process using **GIT**, **Jenkins** job creation, **Maven** build Create **Docker** image and deployment in **AWS** environment.
* Deployed **Spring Boot** based **microservices** **Docker** and **Amazon EC2** container using **AWS** admin console.
* Developed **API** for using **AWS** Lambda to manage the servers and run the code in the **AWS**.
* Created detailed **AWS** Security groups which behaved as virtual firewalls that controlled the traffic allowed reaching one or more **AWS** **EC2** instances.
* Worked on creation of custom **Docker** container images, tagging and pushing the images.
* Worked on creating the **Docker** containers and Docker consoles for managing the application life cycle.
* Used **Git** as Version Control System for the application. Used **JMeter** to test performance of **SOAP** protocol.
* **GUI** design and development using **Java Swing, JFC** Used for open source web testing.
* Developed and tested **extraction, transformation, job** and **load** **(ETL)** processes.
* Performed cross browser testing by using **Selenium**Grid on multiple browsers and on different platforms.
* Oversaw all aspects of testing **JUnit, FIT,** and **SIT,** **End to End, Regression,** **User acceptance** and **Performance.**

**Environment: Java 1.7/1.8, J2EE, HTML 5, CSS 3, Boot Strap, Spring Boot, Hibernate 4.x, Core Java, Angular JS2, SOAP/Rest, Web Services, Eclipse RCP SASS, WSDL, JIRA 5.x/6.x, Microservices, JMS, Node.js, SQL, NOSQL(MongoDB), ETL, Grunt, AWS, Spring Cloud, Spring batch, Git, Apache JMeter, JQuery, JUnit, Docker, Maven, Jenkins.**

**Client: Citi Bank, Tampa, FL Duration: Apr 2015 to Jun 2016**

**Role: Sr. Java/J2EE Full Stack Developer.**

**Description:**

Service Desk is a primary IT service intended to provide a Single Point of Contact ("SPOC") to meet the communication needs of both Users and IT employees. The application is implemented as central point of contact for handling customer, user and other issues.

**Responsibilities:**

* Involved in all the phases of software development lifecycle **(SDLC)** including Requirements Collection, Design & Analysis of the Customer Specifications, Development, Deployment and Customization of the Application.
* Developed **UI** using **HTML, CSS** and **JQuery** to display the data on the client side. Wrote **SPA** using **Node JS** as the server side, **GULP, WEBPACK (ES6), Restful web services** plus **AJAX** and **Angular JS**.
* Used **Backbone** **JS** to optimize in-browser use, to load the module and to improve the Speed. Front-end development by utilizing **Java script** frameworks such as **JQuery, NodeJS** and **AngularJS**.
* Developed a dashboard based on **Ajax** **calls** with Business Customers Count, Flags, Real-time graphs for Analytical
* Reporting using **NodeJs** and Wrote code to fetch data from web services using **JQuery AJAX** for asynchronous communication via **JSON** response and updating the modified data to display in **UI** pages.
* Called web services with **REST** **Architecture** to specify constraints through **Angular JS** services to get **XML** files.
* Involved in implementation of enterprise application communication and integration with **Web Services** and **Legacy** Systems using **REST** (Using **Axis** and **Jersey** frameworks).
* Used various **Core** **Java** concepts such as **Multi-Threading, Exception Handling, Collections framework** extensively to store, retrieve and manipulate data objects of application in the business logic layer.
* Involved in Spring Configuration **XML** files that contain declarations and another dependent object declaration.
* Used **J2EE** patterns like Singleton, Service Locator, Session Façade and Business Delegate during development.
* Developed **micro services** with **Spring Eureka** to retrieve **API** routes for the entire cluster. Using this strategy each micro service in a cluster can be load balanced and exposed through one **API gateway**.
* Used **Spring Transactions** in the application and spring beans to develop Session beans for the business logic.
* **Spring MVC** Framework **IOC** design pattern is used to have relationships between application components. It is also used to separate the application configuration and dependency specification from the actual application code.
* Implemented **spring** service layer with dependency wiring, transaction, **DAO**, Annotations for Dependency Injection.
* Used **Microservices** with **Spring Boot** services interacting through a combination of **REST** and **Spring Boot**.
* Maintaining quality **Microservices** code and developing a **RESTful API** to interact with the microservices.
* Designed and developed Application based on **Spring** **Framework, Spring MVC** and **spring templates**, Used Spring Framework for **Dependency Injection** and integrated with **Hibernate DAOs** for **ORM** mapping.
* Composed and executed **MongoDB** document scripts to insert, update **NoSQL** database, automating data changes, reporting and wrote **MongoDB** client to receive data from **AMC** for Companies.
* Used **NoSQL MongoDB** database to provide recommendations on Performance tuning for applications.
* Used **Hibernate** in persistence layer for mapping an object-oriented domain model to a relational database.
* Experience in designing and deploying **AWS** Solutions using **EC2, S3, EBS,** Elastic Load balancer, auto scaling groups.
* Created detailed **AWS** Security groups which behaved as virtual firewalls that controlled the traffic allowed reaching one or more **AWS** **EC2** instances.
* Worked on **Pivotal Cloud Foundry** to host Application which used to role-based access controls, applying, enforcing roles and permissions to ensure that users can access for which they have been granted access.
* Re-architect monolithic service into **microservices** based architecture using **Spring boot** and pivotal cloud foundry.
* Used **Selenium** Data Driven framework for testing the application and integrating it with **Maven**, **TestNG** [Unit Testing tool] and Jenkins [Continuous Integration Tool] for continuous delivery pipeline.
* Used **Maven** as the Build tool. Experienced in using assistant tools such as **GIT** to access the code repository.
* Implemented the logging using **log4j** which is used to log both User Interface and Domain Level Messages.

**Environment: J2EE 1.7/1.8, spring 4.0, Hibernate 4.0, AJAX, JMS 1.1, XML, SOAP, WSDL, REST, JDBC, JAX-WS 2.2, JavaScript 1.8.2, AngularJS, NodeJS, Spring Eureka, UML, HTML5, JAXB 2.2, SVN, Log4J, PCF, JMeter, Microservices, WebSphere 8.5, Application Server, Eclipse, Oracle 11g, GIT, Maven, Jenkins.**

**Client:** **Cerner Corporation, Kansas City, MO Duration: Feb 2014 – Mar 2015**

**Role: Java Developer**

**Description:**

Cerner Corporation is one of the largest providers of Healthcare Information Technology solutions, services, devices and hardware. Cerner solutions enable physicians, nurses, and other authorized users to share data and streamline processes across an entire organization. Cerner&#39;s comprehensive portfolio of solutions is designed on a common blueprint known as Cerner Millennium. Cerner Millennium solutions combine technology with knowledge to deliver vital data needed for real-time decision-making.

**Responsibilities:**

* Worked on **Agile** methodology including team-based design and code reviews.
* Actively involved in Requirement Analysis, designing, development and testing of the entire application.
* Developed the web interface using **HTML5, CSS3, JavaScript, JQuery, ReactJS**, and **Bootstrap**.
* Develop various screens for the front end using **ReactJS** and used various predefined components from **NPM** (Node Package Manager) and redux library.
* Implemented **Spring MVC** controllers, configured using Spring **MVC** annotations and configuration files.
* Implemented spring framework Bean Factory & Application Context for bean initialization and configurations.
* Spring **AOP** is used for logging, auditing, and security and transaction management to distinguish business logic from the crosscutting concerns.
* Implemented the back-end functionality of the modules using **Singleton** and **Session Facade** design patterns.
* Used **Hibernate** as an **ORM** tool to interact with the database.
* Involved in development of **POJO** classes and writing **Hibernate query language (HQL)** queries.
* Asynchronous messaging is done using **JMS** and created Queue for point to point messaging.
* Developed **REST** web services using **Spring** **MVC** to extract client related data from database and used **SOAP UI** for testing these **RESTFUL API** web services.
* Implemented Service Oriented Architecture by developing and consuming **RESTFUL** services based on **JAX-RS**.
* Implemented Java Classes to use **JAXP** parser to create Java objects, able to modify the data in the response.
* Worked with **DB2** and Wrote **SQL** Queries, stored Procedures, triggers, and integrated **SQL** Queries into **DAO**.
* Deployed the application on **Apache** **Tomcat** Application Server.
* Used **Log4j** logging mechanism for logging.
* Used **SVN** for versioning and concurrent access to the project files.
* Resolving spring framework artifacts and code build & deployment with **Maven**.
* Used **JIRA** for defect and issue tracking and Used **Jenkins** for continuous integration/continuous deployment.
* Automated regression test cases using **Selenium**.

**Environment: Java 6, J2EE, Spring MVC 3.0, Hibernate 3.0, HTML, CSS, Node.js, JavaScript, React JS, JMS, SOAP, Axis2, Maven, Jenkins, DB2, Selenium, Log4j, SVN, JIRA, Tomcat 6.0.**

**Client: 1800Flowers, New York, NY Duration: Nov 2012 – Jan 2014**

**Role: Java Developer**

**Description:**

This Application is a telephonic order capture and order fulfillment system developed to shop flowers, bouquets and gifts items online. This system downloads orders from other systems, processes and fulfills the orders. In Atlas systems, agent’s takes order via telephone, customer service call and process order in queues. This system helps business people to maintain inventory, manage promotions, florist and vendor details, generate reports for business people and financial people.

**Responsibilities:**

* Involved in Software Development Life Cycle **(SDLC)** - design, coding, testing, and documentation using Rational Unified Process **(RUP)** for analysis and design of application.
* Developed UI using **HTML, JSF**, **JQuery** and **JavaScript**, **AJAX** in few applications.
* Developed **CSS** sheets for developing user aesthetic UI components.
* Developed Web applications in **MVC** architecture using **Spring** framework.
* Implemented logging in the application using **Spring** Framework **AOP** module to know the application status. Used Spring **DAO** module for performing **CRUD** operations on database.
* Used Spring **IOC** design pattern to provide relation between application components. Separated application configuration and dependency specification from actual application code.
* Developed **SOAP** Web service modules based on **SOA** architecture to consume services.
* Called external applications using **Web Services** for delivering notifications.
* Used **JAXB** parser for parsing the valid **XML** files as defined by the **XSD**.
* Used **JMS** API to access the queues, read and write the messages for processing.
* Worked on **JUnit** for creating test cases for all the Business Rules and the application code.
* Experienced with setting up the basic project setup from scratch using **Spring-Hibernate** based design.
* Worked on **JPA** as persistence layer, involving **EJB 3.1** annotations to synchronize with database through **XML**.
* Deployed the application in **JBoss** Application Server. Used **ANT** scripts to fetch, build, and deploy application to development environment and used **CVS** as version control.
* Interacted with application testers to review system defects and provided comprehensive fixes.

**Environment: Java 6, J2EE, HTML, CSS, JSF, JavaScript, AJAX, SOAP, SOA, JMS, Hibernate, EJB, JBoss 6.0, Ant, CVS, JAXB.**

**Client: State Street Corporation, Boston, MA Duration: Sep 2011– Oct 2012**

**Role: Java Developer**

**Description:** Developed an application which provides users a web interface to the State Street Accounting application

and to view the available fund data in various formats. It also validates various transactions and activities and generates

alert messages. Users can log on to the system to research and resolve the alerts.

**Responsibilities:**

* Participated in all phases of the project development life cycle like design, development, testing, enhancements and production support.
* Used Rational Rose for the Use Case Diagrams, Object Diagrams, Class Diagrams and Sequence Diagrams to represent the detailed design phase.
* Developed front end using **JSP, HTML, CSS** and client-side validations using **JavaScript**.
* Implemented **MVC** architecture and developed action classes for the entire application using **Struts** framework.
* Implemented **Singleton**, **Service Locator** design patterns in **MVC** framework and developed command, delegate, model action script classes to interact with the backend.
* Involved in writing the exception and validation classes using **Struts** validation rules.
* Designed and developed reusable Java classes for business logic functionality using services like **JDBC** and **JNDI**.
* Involved in creating the Registry Resources in **WSO2** Governance Registry which would become a single resource for the registered schemas.
* Developed **WSO2** Carbon based Data Service layer to enable CRUD operations on Meta Content Repository.
* Deployed the application on **WebLogic** application server.
* Monitored the error logs using **Log4j** and fixing the problems.
* Designed database Tables, View, Indexes and created triggers for optimized data access.
* Written **JUnit** test cases for unit testing of the application.
* Written **ANT** scripts for building application artifacts.

**Environment: Java 6, J2EE, JSP, HTML, CSS, JavaScript, Struts 2.0, EJB 2.1, WSO2, JDBC, JNDI, JUnit, Log4j, WebLogic 7.0**

**Client: Jayam Solutions - Hyderabad, India** **Duration: Jun 2009 – Aug 2011**

**Role: Jr. Java Developer**

**Description:**

The aim of the project is to implement a web based front end system which can be used by the employees, managers and management having their specific role permissions for effective processing of day to day life business activities in the company like leave management, project management and resource management etc.

**Responsibilities:**

* Involved in all phases of the development life cycle (**SDLC**) involving in requirement Collection. Design and analysis of the customer specifications of the application using **TDD** Methodology,
* Implemented struts framework using **MVC** design pattern and front controller pattern
* Designed user interface by creating web pages for each component using Struts web **MVC** and Action Servlet as front controller in this application.
* Developed Action Class components for performing business process execution and involved in configuration of struts configuration specific xml file
* Developed **SCM** by using the **JSP/HTML** for each user interface. JavaScript, Servlets standard validations are used as the controllers for business logic and using **JDBC**, **XML** parsing techniques using **MVC**
* Designed mailing application to send notifications to the Oracle users for intimating to start the next workflow.
* Designed and implemented **AJAX** server calls to dynamically retrieve shipping information.
* Worked on **Oracle** in writing complex queries in the business logic layer. Used joins, debugging the SCM related packages and customization.
* Involved in testing the application using **JUnit**.
* Designed, coded and tested business logic for software applications.

**Environment**: **Core Java, JDK, J2EE, JSP, MySQL, Ant Build, JDBC, Struts, HTML, XML, Java Script, JUnit, Ajax, TDD Methodology, Apache Tomcat, Eclipse.**