**Pl Chandini P**

Email: chandini9608@gmail.com **Phone: - +1 3148848179**

|  |
| --- |
| **Professional Summary** |

* **Seven years of experienced IT planning** and development experience in the Telecom, Health Care Industries using **OOPS, and JAVA/J2EE** advancements. Experience in working with **Model View Controller (MVC),** design for web improvement utilizing **Spring MVC, JSP, JavaBeans** and **Servlets.**
* Good hands-on experience on Experience in building **REST webservices, Microservices with Spring boot** and **Gradle**, **Maven** **build tools** for web applications design, development, and deployment.
* Experienced in all stages of the Software Development Life Cycle (SDLC), including analysis, design, development, coding, support, integration, maintenance, and application enhancements after deployment.
* Experienced in using front end debugging tools like, Developer tools.
* Experience with **Core Java** concepts like **Collection Framework, Multi-threading, Generics, Serialization.**
* Experienced in **Unix**, **Linux** shell scripting to set and run Cron jobs and using build and deployment tools like **Maven**.
* Experienced in using test frameworks like **Junit**, **Selenium.**
* Experienced in testing web services using tools like **POST Man.**
* Experienced with **SQL database**s like **Oracle**, **SQL Server**, **MySQL**.
* Experienced in using IDE tools like **STS**, **Eclipse**, **IntelliJ.**
* Experience on ideas of Java 8 like Functional Interfaces and Lambda Expressions.
* Worked extensively with **Git, SVN, Jenkins, PHP Unit, PHP My Admin, JIRA, and Apache Web Server.**
* Experience in working with **Java 8** features like **Functional Interface, Lambda expressions, Java Stream API** and **Time API**.
* Experienced visualizing table relationships in a database using free license DB Visualizer tool.
* Experience in **DOM, React Props, Lifecycle methods, working with React States and Events.**
* Experienced in writing user manuals and documentation of web application flows and creating.
* Experience in working with **Agile (SCRUM)** improvement methodology.
* Experience in applying best **Design Patterns** and **Strategies** to enormous business applications to diminish tight coupling, enhance execution, developing and testing.
* Experience in creating Form Validation by jQuery.
* Proficiently designed, developed, and deployed enterprise applications on Microsoft Azure cloud platform, utilizing services like **Azure Virtual Machines, Azure App Service, and Azure SQL Database**.
* Implemented microservices architecture using **Azure Functions and Azure API Management,** resulting in improved scalability and efficient **API management**.
* Designed and configured **Azure Virtual Networks, Network Security Groups (NSGs), and Azure Firewall** to ensure a secure and isolated network environment for applications.
* Java-based enterprise applications have been successfully deployed and managed on the **AWS** cloud architecture using services like **Amazon EC2, Amazon RDS, and Amazon S3**.
* **AWS Lambda and API Gateway** were used to design and deploy a microservices architecture, which increased scalability and decreased operational cost.
* Developed various **Struts framework** Action classes for interfacing with the business logic form beans for presenting the data to **JSPs**.
* **React/Redux** was used to create the responsive **UI components** and to send events to the store.
* **React JS components, Forms, Events, Keys, Router, Animations, and the Flux idea** were used during development.
* **React JS** was used to implement the client-side interface, together with the **Redux library** and a number of prefabricated components from **NPM** (Node Package Manager).
* involved in developing **React components** for data visualization using **React with Redux and ES6.**
* **React HTTP** service was used to handle **GET, POST, PUT, and DELETE requests for API responses.**
* **React Forms** was used to create a variety of template- and data-driven form components, while **React router** was used to specify module-level routing for a variety of application components.
* Experienced in Creating Domain Model, Activity Diagrams, Use Case diagrams, class diagrams, sequence diagrams and Object Model.
* Experience in design and development of n-tier web applications using **Java** and its framework like, **Spring, Hibernate, Web Services** and **J2EE** standards.
* Experience in making and solid perception of **Micro Services** Architecture.
* Used Bootstrap for responsive design in conjunction with **AngularJS** for maximum scalability.
* Designed and implemented the enhanced development platform by doing POC using frameworks like **Angular 10**, **NodeJS, NPM**.
* **AWS EC2** instances were built and configured for use in setting up cloud clusters, and **S3** buckets were generated for our application.
* I designed affordable, fault-tolerant, and highly available systems using many **AWS instances**, security groups, elastic load balancing, and auto scaling.
* Experience in producing **RESTful services** using **JavaScript, jQuery and Angular10**.
* Experience in web servers like **Apache Tomcat** on entire **Software Development Life Cycle** (SDLC).
* Experience using **Hibernate** for mapping **Java classes** with **database** and using Criteria **API** and **Hibernate Query Language (HQL).**
* Hands on experience with writing test cases and developing the test cases with **JUnit** and **mocking frameworks** such as. **Mockito**
* Hands on experience with **SQL/No SQL** databases systems like **Oracle, MySQL, Mongo DB.**
* Hands on experience on Operating Systems like **Ubuntu, Linux, and Windows.**
* Excellent communication skills, interpersonal, hardworking, mentoring and a very good team player.
* Skilled in using version control and branch repository tools like Gitlab.
* Used web-based **GIT** repository manager with issue tracking features.
* Hands on Experience on code quality control tools like **SonarQube**.
* Extensively used Git for version controlling and regularly pushed the code to **GitHub, GitLab,Svn.**
* Experienced in using **JIRA** as bug tracking tools.

**Technical Skills**

|  |  |
| --- | --- |
| Languages | Java, J2EE |
| Enterprise Java | Java8, J2EE, Spring, JSP, Servlet, JDBC, JPA, JDBC, |
| Web Technologies | JavaScript, AngularJS, JS, XML, HTML5, CSS3, XHTML, XML Schema DHTML, JSON RPC, AJAX, jQuery, WebSocket, Knockout JS, React, React js, Angular JS & Angular 10, XSL, XSLT, SED, DTD. |
| Mark-up/XML Technologies | HTML5, CSS3, JavaScript, jQuery, Angular JS, Node JS, XML, AJAX, JSON |
| Tools & Framework | Spring, Spring MVC, Spring IOC, JUnit. Spring Boot, |
| Web services | RESTful, Soap Web services |
| Architectures | EJB, Multi-Tier Enterprise Application, JDBC, Extreme Programming (XP) and Continuous Integration (CI) |
| Web/App Servers | Apache Tomcat |
| Frameworks | Struts, JSF, Hibernate, Spring, Junit |
| Database | Oracle, SQL-Server, MySQL server, DB2, Mongo |
| IDE/ Tools | Eclipse, Spring Tool Suite (STS), IntelliJ, Visual studio |
| SDLC | Agile - Scrum, Waterfall |
| Testing Tools/ Others | JUnit, Selenium. |
| Version Control | GIT, Gitlab, Bit Bucket |
| OS & Environment | Windows Vista/XP/2000/NT, LINUX, IOS |
| Design Patterns | MVC, Front Controller, Singleton. |
| Cloud/CI-CD | Jenkins, Docker, Kubernetes, Cloud bees, AWS |
| Messaging System | KAFKA |

|  |
| --- |
| **Professional Experience:** |

**Client: Centene Corporation**

**Location: St. Louis (Remote) JAN’22 Til Date**

**Role: Full Stack Developer**

* Involved in **SDLC** cycle like Analysis, Designing, developing, Testing and deploying the Enterprise applications using **Java** and **Object-Oriented Design Patterns** based on business requirements.
* In meetings, there is daily interaction with the client's technical architecture with an emphasis on the functionality and application of code development.
* Implementing **new APIs** that connect with clients' legacy systems through **Restful Web services and Spring Boot**.
* With very little configuration, **Spring Boot** is used by experience to **build cloud Microservices and create Spring-based applications**.
* Interacted with external interfaces using **Kafka Messaging Services**.
* I have used **HTML5, CSS3, Bootstrap, and ReactJS** to design, construct, and test the web application
* **React/Redux** was used to create the responsive **UI components** and to send events to the store.
* **React JS components, Forms, Events, Keys, Router, Animations, and the Flux idea** were used during development.
* **React JS** was used to implement the client-side interface, together with the **Redux library** and several prefabricated components from **NPM** (Node Package Manager).
* involved in developing **React components** for data visualization using **React with Redux and ES6.**
* **React http** service was used to handle **GET, POST, PUT, and DELETE requests for API responses.**
* **React Forms** was used to create a variety of template- and data-driven form components, while **React router** was used to specify module-level routing for a variety of application components.
* Used **angular testbeds** and async utilities to fake async calls while developing unit test cases **using Jasmine, Karma, and Web Pack**.
* Performed **CRUD** operations on **Mongo DB** and created collections. Using **the Spring Mongo Template and Mongo Repository for persistence**.
* used **Junit and Mockito** to write the test cases for the application.
* Created a data pipeline using **Storm** and **Kafka** to store information in **HDFS**.
* **Kafka, Flume**, and real-time data loading for leaking information utilizing **Storm and Spark**.
* Different hive optimization methods, including **Dynamic Partitions, Buckets, Map Joins, and Parallel Executions, have been implemented.**
* I have updated multiple functional tests to ensure that they reflect the latest enhancements and changes made to the code.
* I have also created **test cases** that provide good functional and code coverage, which can be used to automate test suites.
* The project involves working with **AWS** to create server instances and using tools like **Tomcat and Swagger to deploy Microservices**.
* Java-based enterprise applications have been successfully deployed and managed on the AWS cloud architecture using services like **Amazon EC2, Amazon RDS, and Amazon S3**.
* **AWS Lambda and API Gateway** were used to design and deploy a microservices architecture, which increased scalability and decreased operational cost.
* Throughout the project, I have followed the **AGILE methodology with regular SCRUM meetings.**
* To maintain version control of the files, I have used **GitHub** and created new branches whenever a new feature implementation starts.
* Additionally, I have collaborated with the offshore team to transition requirements and provide the necessary inputs for the successful execution of the project

.

**Environment:** Java 1.8, Spring Boot, Kafka, IntelliJ, Microsoft Visual Studio, TypeScript, React Js, Git, Cucumber, SOAP, Jasper Reports, Jenkins, AWS, Oracle, Mongo DB, Flume, spark, Kubernetes.

**Client: At&t**

**Location:Banglore FEB’21– DEC’21**

**Role: Full Stack Developer & UI Developer**

**Responsibilities:**

* **Involvement in all stages of the project, from the analysis and requirement phase to project delivery**.
* After gathering all the customer requirements, I created the project's **System Requirement Specifications** (SRS) report.
* **Agile Methodology** was used to develop the functionalities.
* Implementing new **APIs using Restful Web services** and **Spring Boot** to communicate with clients' old systems and mobile applications.
* Used **Angular 10, Typescript, HTML5, CSS3, and Bootstrap** to create single-page applications.
* **Angular 10's** component-based architecture was used to create Typescript reusable components and services for consuming REST APIs.
* Knowledge of Angular APIs, such as **@angular/http HTTP** requests are made using @angular/common. Typical fundamental components like form validation and **@angular/router** **Routing** for the whole of our app.
* Used concepts for **Hibernate caching, Spring/Hibernate batch transactions, and Hibernate transaction management.**
* Expertise utilizing **Apache Kafka** to manage communications services.
* Using **Spring Cloud, the whole Microservices architecture** was designed, including the **Gateway**, **API** **Service, Core Service, Discovery Server, and Monitor Server.**
* Saved process as a service was built **Microservices** that are standardized: will take the name and parameters and output the database's output or an error.
* using the **Java API** to load data into a **Cassandra** cluster.
* **AWS EC2** instances were built and configured for use in setting up cloud clusters, and S3 buckets were generated for our application.
* I designed affordable**, fault-tolerant, and highly available systems using many AWS instances**, security groups, **elastic load balancing, and auto scaling**.
* **Created microservices-based applications utilizing the Spring Boot and Spring Integration frameworks**.
* developed a data pipeline using **Java Map, Flume, and Pig Reduce** to import client behavioral data and histories of financial transactions into **HDFS** for analysis.
* Utilize **Pivotal Cloud Foundry** (PCF) to deploy and manage microservices-based applications on cloud infrastructure.
* By sharing the **WSDL**, the **Web Services** were made available to the client apps.
* **DDL and DML SQL** scripts were created to create database objects in the **SQL** Server database.
* By putting the principles of normalization and **RDBMS** into practice, logical and physical data models were created.
* To speed up the application's response time, new tables, **PL/SQL** **stored procedures, functions, views**, constraints, triggers, and **SQL tuning** were created. Jenkins was used to load the JAR files from the database that were needed to start the program.
* To create an **JAR** file for deployment in application servers, the **Maven** building tool was used. It is also employed in the management of code dependencies in applications.

**Environment:** Eclipse, Web Sphere, Agile Methodology, SVN, Maven, Jenkins, Junit,J2EE, Java 1.7, spring Boot, Angular, Microservices, SQL Server, Hadoop, Scala, MapReduce, AWS.

**Client: FedEx**

**Location: Bangalore NOV’18 – FEB’21**

**Role: Java/UI Developer**

**Responsibilities:**

* Developed user interface using **Spring**, **JSP, HTML5, CSS3**, **JavaScript**, **jQuery, AJAX.**
* Designed user interface to display the customer Financial data known as financial module (for the customer to view the charges they paid or they yet to pay) **using Spring MVC framework, Hibernate ORM Module, XML, JSP and XSLT.**
* Participated in resolving any user interface related issues reported by client.
* Implemented load-balancing with NGINX to allow dozens of Node JS instances to handle thousands of concurrent users.
* Experience with Front-end JavaScript frameworks like Angular JS, Node JS. And with the creation of RESTful Web service.
* Developed **Multitier J2EE** design by utilizing **MVC architecture** and **hibernate.**
* Implemented functionality using **Servlets, JSP, HTML, Hibernate, Spring, Java Script and Web logic.**
* Built **RESTful** Web services.
* Worked on Collections Framework.
* Responsible of maintaining and developing server-side code with **JavaScript** runtime environment.
* Used Oracle database for backend testing and to test functionality.
* Designed and developed **Multithreaded** Java Swing clients for traded financial products.
* Implemented multithreading in backend **Java beans**.
* Responsible for building single-page-applications using **AngularJS2** development framework.
* Designed client-side application which navigates data web sites.
* Issued user defined queries and achieved result set retrieval.
* Extensively used **Hibernate/JPA** and **Spring-JDBC** in data access layer to access and update.
* For externalized configuration **spring cloud** is used.
* Involved in Developing the Application using **Spring MVC** Framework by implementing Controller, Service classes.
* Used **SoapUI** for testing Web services.
* Implemented the Project structure based on Spring **MVC pattern** using **spring boot**.
* Used **Jenkins** for continuous Integration.
* Used **Selenium** testing tool for TDD and BDD.
* Proficiently designed, developed, and deployed enterprise applications on **Microsoft Azure cloud platform, utilizing services like Azure Virtual Machines, Azure App Service, and Azure SQL Database**.
* Implemented microservices architecture using **Azure Functions and Azure API Management**, resulting in improved scalability and efficient API management.
* Designed and configured Azure Virtual Networks, Network Security Groups (NSGs), and Azure Firewall to ensure a secure and isolated network environment for applications.
* Participated in various deployment activities.
* Participated in Sanity testing after the deployment.
* Logically implemented **HTTP** (header cookies) constructors to accomplish site navigation.
* By using Stateless session **EJBs** developed Scalable applications.
* Developed logic for shell script to export oracle table’s data into flat files and performed unit testing by using **Junit**. Used **Log4j** for logging and automatic batch jobs.

**Environment:** Eclipse, Core Java, JSP, JSTL, Servlets, Spring, Spring Boot, Spring Cloud**,** Hibernate, AJAX Framework, MongoDB, UNIX Shell Scripts, XSL, XSLT, HTML5, CSS3, Angular JS, Node JS, UML, JavaScript, Soap UI, Jenkins, JAXP, XML, Log4j, JMS, Node JS, WebLogic.

**Client: AIG Life Insurance**

**Location: Hyderabad, India FEB’17 – OCT’18**

**Role: Java Developer**

**Responsibilities:**

* Participated in Iterative application development, stand-up meetings, and weekly Sprints are all included in the **Agile Scrum technique.**
* **Worked on different microservices like calling one microservice with other and integrating 2 microservices**.
* Worked on developing some changes to the existing microservices.
* Performed **unit testing.**
* Used Dependency Injection feature, provided by spring framework for **Bean Wiring**.
* Designed contract first Web Services using **Spring-WS** to communicate with Database Tables.
* Developed UI pages using **HTML, JavaScript, Angular JS, XML, JSP, CSS** as per the requirements and providing the client side using java script validations.
* Worked on **Restful Web Services** which enforced a stateless client server and support **JSON**.
* Worked on **Angular JS, controllers, views** and **JMS**.
* Developed and designed the service business process and data access to Oracle database and back end layers with **JAXB, JMS** and **XML**.
* Used **Gradle** as the application building tool.
* Implemented **JMS listeners** to connect to the **Tibco JMS queues** for producing and consuming **JMS.**
* Participated in the developmentof **FIX messages** processing engine.
* Designed and developed **JSP, Servlets** and deployed them on **JBOSS** Application Server.
* Used **Spring-JDBC** configuration to manage transactions across multiple database.
* Used **Spring ORM** Support for **Hibernate** to manage Hibernate Resources.
* Designed modules for patient and Reports.
* Used **Junit** and various Test Suites and Test Cases.
* Used **Log4j** to trace the flow of the application.
* Used **MySQL** which is an object relational database.
* Designed Hibernate persistence classes using **Hibernate API**.
* Developed code for **Procedures/Triggers/Functions** using **SQL Navigator** to perform operations on **Oracle 10g** database.
* Deployed application on **BEA Web Logic Application Server** to get efficient performance.

**Environment**: JDK 1.8, Spring-JDB, Spring-ws-core, SOAP, Spring, Hibernate, Maven, Log4j and JUnit, Eclipse, HTML, CSS, Angular JS, JavaScript’s, Oracle 11g, SVN

**Client: Genpact**

**Location: Hyderabad JUN’15 – JAN’17**

**Role: Java Developer**

**Responsibilities:**

* The application was created in an **agile** setting using the Scrum methodology.
* Built the architecture using the MVC design pattern of the well-liked **J2EE** Model 2 approach.
* Used the **Spring framework** to develop an application, using Spring capabilities like **Spring Dependency injection**, **Spring Security, and Spring Web flow with Winter MVC.**
* Utilized the accessible **REST Web Services** by using the **Spring Rest Template**.
* Creating code for the spring framework's bean references utilizing **annotation-based Inversion of Control** (IOC) or **Dependency Injection** (DI).
* Business components were created using **Java objects, and the Hibernate framework** was utilized to map the Java classes to the **database**.
* The development of **asynchronous** message-based point-to-point application communication using **JMS, MQ,** and MDB process message.
* created **Java/J2EE** code, business logic using the **Spring framework**, the **Hibernate framework,** and OOP ideas, and participated in peer code reviews.
* implemented responsive development using **CSS3** media queries and **Bootstrap**.
* Implementation of the authentication, authorization, and access-control elements was done using **Spring Security**.
* **Spring MVC** was used to publish Restful services, and the spring-rest template was used to consume those services.
* **RESTful** web services are created using **spring boot, JPA, Cloud Foundry, and Oracle** to implement functionality.
* Skilled in creating **sophisticated SQL queries** to pull data from numerous tables.
* Extensive SQL query execution to view data transactions that were successful.
* Data was extracted from a variety of sources, including **SQL Server, flat files, and Oracle.**
* **XML, XSLT, and XSD** are being developed for schema definitions.
* Integration and unit testing application evaluation.
* Coordinated with the testing team to monitor tickets for bug reporting and fixes using JIRA.
* Experience writing and running repeatable tests using unit testing tools like JUnit and test-driven software development.
* Used **GIT HUB** to manage release versions and maintain the component.
* War file for the application was built using Jenkins.
* Web services were tested using **SOAP UI and Sonar**.
* used **Log4j** to record performance logs, messages, and faults.

**Environment**: JDK 1.6, Spring Framework, Oracle, Eclipse 3.2, Maven, WebLogic 11g, SVN, JUnit, Log4j, NoSQL, REST, Hibernate 3.3.

|  |
| --- |
| **Education:** |

**Master of Science in Computer science and Engineering,** May 2023