**Name: Pavan Krishna**

**Sr. Java Full Stack Developer**

**Email: pavankbf502@gmail.com  
 Phone: 4085480491  
 LinkedIn:** [**https://www.linkedin.com/in/pavan-krishna-tirutullai-05501a194/**](https://www.linkedin.com/in/pavan-krishna-tirutullai-05501a194/)

**PROFESSIONAL SUMMARY:**

* **10+ 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.
* Strong experience in developing applications using **Java/J2EE technologies** like Servlets, Struts, accustom tags, Java Beans, EJB, Spring, Spring Boot, JDBC, Hibernate, JBPM, JMS, JSF.
* Good 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.
* Experienced in Object Relational Mapping (ORM) using **Hibernate 4/5** and **Java Persistence API** Standard (JPA) for database mapping and data retrieval facilities.
* Expertise in **UI technologies** like XML, XML Spy, HTML, HTML5, DHTML, XHTML, CSS, CSS3, XSLT, JSON, jQuery, AJAX, JavaScript, AngularJS, NodeJS, ReactJS and BackboneJS.
* Experience in **XML technologies** such as DTD/Schemas, DOM, SAX, JAX-B, and XSLT.
* Good command and knowledge in working with **Core Java Concepts** like Collection frameworks, Interfaces, Synchronization, Serialization, Thread pools, JavaBeans, Externalization, Generics and Annotations.
* Hands-on experience on application development for **Cloud platforms** using technologies like Java/J2EE, Spring Boot, Spring Cloud, Microservices.
* Skillful in implementing **Microservices architecture** by utilizing Spring Boot, Spring Cloud, RESTful API, AWS for cloud-native infrastructure, Docker for containers, Redis and Nginx for middleware.
* 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.
* Expertise in implementing client-server business applications using **Microservices**.
* Expert in **Spring Security** for Authentication and Authorization of REST services using OIDC on top of **OAuth**.
* 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**.
* Experience in creation of **Docker** containers and Docker consoles for managing the application life cycle.
* Good experience in implementing role-based authentication and authorization using **SiteMinder and Spring Security**.
* Experience in using **Hibernate** for mapping Java classes with databases and using Criteria API and HQL.
* 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, and MySQL.
* Hands-on experience in implementing, designing and developing **NoSQL Databases** like **MongoDB, Apache Cassandra and CouchDB**.
* Proficient in using **AJAX** for implementing dynamic Webpages where the content was fetched via API calls and updated the **DOM (JSON Parsing)**.
* Experience in using **JDBC** and **ORM tools** like **JPA, Hibernate** for persisting data to different relational databases.
* Strong experience in building **REST APIs** in **Java** and **NodeJS** using spring boot and Express’s respectively.
* 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 in using **Jenkins** for **CI/CD processes** 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.
* Extensive experience with project build/test/deployment tools like **Maven and Ant**.
* Experience in using **Code Repository Tools** - Tortoise SVN, CVS, GitHub, and Visual Source Safe.
* Experience working with **Log4j** for logging and used **JUnit and Mockito** for unit testing and integration testing.
* Expertise in using **JIRA** for bug tracking and project management.

|  |  |
| --- | --- |
| **Programming Languages** | Java 11/8/7, J2SE, J2EE, C, C++, PL/SQL |
| **Java/J2EE Technologies** | JSP, JSTL, JSF, Struts, Hibernate, JNDI, Servlets, JDBC, EJB, Bootstrap |
| **Web Technologies** | HTML5, XML, CSS3, JavaScript. JQuery, AngularJS, NodeJS, ReactJS, BackboneJS, Ajax,  JSON |
| **Spring Frameworks** | Spring MVC, IOC, AOP, JTA, Spring Dashboard, Spring Boot, Spring Security |
| **Microservices** | Spring REST, Spring Netflix, Spring Zuul Proxy |
| **Cloud Environments** | AWS, Spring Cloud, Netflix Eureka, Pivotal Cloud Foundry |
| **Web Services** | Restful API, Microservices, JAX-RS, JAX\_WS, JAX-RPC, SOAP, WSDL |
| **IDE** | Eclipse, Net Beans, Visual Studio, STS, Brackets, IntelliJ, RAD |
| **Version Control** | GIT, CVS, SVN |
| **Testing Tools** | JUnit, Mockito, Jira, Log4j, SOAP UI, Selenium, Cucumber, TestNG |
| **Build Tools** | Gradle, ANT, Maven, Jenkins |
| **Databases** | Oracle 9i/10g/11g, DB2, MySQL 5.1, SQL Server, NoSQL, Cassandra, Mongo DB |

**TECHNICAL SKILLS:**

**PROFESSIONAL EXPERIENCE:**

**MasterCard, St Louis, MO Apr 2022 to Present**

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

* Implemented **Java 11** features for the application development.
* Developed **UI pages** using HTML5, CSS3, ReactJS, NodeJS and JavaScript and used Bootstrap for creating responsive Web pages.
* 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 **Apache Kafka** for reliable and asynchronous exchange of important information such as payment status report.
* Designed and implemented multi-threaded Java applications to optimize performance in high-concurrency systems, reducing response time by 30%.
* Utilized **Java 11/17** features like Records, Streams, and Functional Interfaces for clean and efficient coding practices.
* Applied **SOLID** and **OOP** principles to develop modular and scalable backend component.
* Created Spring Boot with Micro Services utilizing RESTful API.
* Consumed **REST Web Services** for getting the credit card information from a third party application.
* Developed Restful Web services for other systems to interact with our system and added WS-Security using XWSS to secure the Web services.
* Integrated **MongoDB** for storing unstructured data and leveraged Spring Data MongoDB for seamless access.
* Built cloud-native Java apps and deployed to **AWS (EC2, S3, RDS, Lambda**) and Azure App Services with CI/CD.
* Used Docker to containerize **Spring Boot** applications and orchestrated deployments using Kubernetes.
* Implemented blue-green and canary deployments to minimize downtime during releases.
* Developed and deployed over 15 microservices using **Spring Boot**, **Spring Cloud**, and **Netflix OSS** stack (Eureka, Zuul, Ribbon).
* Implemented REST APIs using **Spring MVC** with **HATEOAS** and integrated Swagger/OpenAPI for **API** documentation.
* Configured circuit breakers and service resilience with **Resilience4j** and **Hystrix**.
* Implemented the associated business modules integration using **Spring and Hibernate** data mapping.
* 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.
* 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.
* Used **Spring Batch** which uses **Spring IOC** dependencies, this is useful for developing batch processing.
* Implemented **Hibernate** Template to great extent making use of HQL, Named Parameters, Named Queries and interfaces such as Expression, Query, and Criteria.
* Used **Node Package Manage (NPM)** for configuring the modules and libraries required for the application.
* 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.
* Migrated SOAP web service resources to Spring RESTful web services resources using **Spring REST API** with Spring Boot.
* Used **J2EE design patterns** like Adapter, Service locator, Session facade, Singleton, Factory and DAO.
* Developed the User Interface of the application using HTML5, CSS3, ReactJS, NodeJS, JavaScript, AJAX, JSON, and Bootstrap with Oracle database as the backend.
* Implemented Angular observables and listeners (RxJS reactive libraries) to simplify the making asynchronous calls to backend **Rest APIs** and used **Swagger** UI for API documentation.
* Authenticated users and provided the security to RESTful Web Services using OAuth 2.0.
* Developed unit and integration tests for existing microservices using **JUnit and Docker** environments.
* Involved in **CI/CD process** using **GIT**, **Jenkins** job creation, Maven build create Docker image and deployment in

**AWS** environment.

* Wrote **SQL queries** using PL/SQL, stored procedures and enhanced performance by running explain plans.
* Deployed the application on **Apache Tomcat** Application Server.
* Developed and coded J2EE Components with **JSP, Java Beans**, and **Business Objects** with Hibernate.
* Actively used the defect tracking tool **Jira** to create and track the defects during QA phase of the project.
* Used **JMeter** to test performance of SOAP protocol.

**Environment:** Java 11, J2EE, AWS Cloud, AWS EC2, AWS Lambda, AWS Security Groups, Spring Cloud, Spring Boot, Spring Batch, Spring Security, Spring IOC, Spring Core, AOP, Hibernate, HTML 5, CSS 3, Bootstrap, Core Java, Angular JS, Java Swing, SOAP/Rest, Web Services, Eclipse, JIRA 5.x/6.x, Microservices, JMS, SQL, NoSQL (MongoDB), Drools, Git, Apache JMeter, JQuery, JUnit, Docker, Maven, Jenkins, Selenium

**Signify Health, Dallas, TX Feb 2020 to Mar 2022**

**Java Full Stack Developer 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.
* 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.
* Used **J2EE patterns** like Singleton, Service Locator, Session Façade and Business Delegate during development.
* Developed UI using **HTML, CSS and JQuery** to display the data on the client side.
* Wrote SPA using **NodeJS** as the server side, **Gulp, Webpack (ES6), Restful** web services plus Ajax and AngularJS.
* Used **Backbone JS** to optimize in-browser use, to load the module and to improve the Speed. Front-end development by utilizing **JavaScript 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.**
* Designed and deployed **AWS Solutions** using EC2, S3, EBS, Elastic Load balancer, auto scaling groups.
* 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-architected monolithic service into microservices based architecture using **Spring Boot and Pivotal Cloud Foundry**.
* Automated build and deployment pipelines using **Jenkins**, **GitLab CI**, and **GitHub Actions** for faster release cycles.
* Managed application configuration and secrets using **Spring Cloud Config Server** and **AWS Secrets Manager**.
* Monitored production apps using **Prometheus**, **Grafana**, and **ELK Stack**.
* Achieved 90%+ unit test coverage with **JUnit 5** and **Mockito**, ensuring robust and bug-free code.
* Performed integration testing with **TestContainers** and end-to-end testing using **Selenium** and **Postman/Newman**.
* Employed TDD and BDD methodologies in Agile sprints for better code reliability.
* Involved in **Spring Configuration** XML files that contain declarations and another dependent object declaration.
* Developed **Microservices** with **Spring Eureka** to retrieve API routes for the entire cluster. Using this strategy each microservice in a cluster can be load balanced and exposed through one API gateway.
* Utilized **AWS Lambda** platform to upload data into AWS S3 buckets and to trigger other Lambda functions developed drag and drop document editing.
* Worked on AWS for storage and content delivery and application deployment services.
* Implemented **Data Access Object (DAO)** persistence layer and Object-Relational (O/R) mapping using Hibernate.
* Used **Spring Transactions** in the application and spring beans to develop Session beans for the business logic.
* Used **Spring MVC** Framework IOC design pattern to maintain 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.
* Maintained 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.
* Used **NoSQL MongoDB** database to provide recommendations on Performance tuning for applications.
* 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 **Selenium** Data Driven framework for testing the application and **Jenkins** (Continuous Integration Tool) for continuous delivery pipeline.
* Used **Maven** as the Build tool and used **GIT** as assistant tools to access the code repository.
* Implemented the logging using **Log4j** which is used to log both User Interface and Domain Level Messages.

**Environment:** Java 11, J2EE, Spring, Hibernate, AWS, Pivotal Cloud Foundry (PCF), Spring MVC, Spring Boot, Spring Eureka, AJAX, JMS 1.1, XML, SOAP, WSDL, REST, JDBC, JAX-WS 2.2, JavaScript 1.8.2, AngularJS, NodeJS, UML, HTML5, JAXB

2.2, SVN, Log4J, JMeter, Grunt, Webpack, Microservices, WebSphere 8.5, Application Server, Eclipse, Oracle 11g, GIT, Maven, Jenkins, Selenium, TestNG

**Discover Financial Services, Riverwoods, IL Jan 2018 to Jan 2020 Full Stack Java Developer**

**Responsibilities:**

* Implemented **Spring MVC** controllers, configured using Spring MVC annotations and configuration files.
* Implemented Spring framework **Bean Factory & Application Context** for bean initialization and configurations.
* Used Spring AOP 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 **AWS (Amazon Web Services) services** such as EC2, S3, RDS and Elastic Beanstalk for building the whole app on cloud infrastructure.
* Used **AWS SDK** for connection to Amazon **S3 buckets** as it is used as the object storage service to store and retrieve the media files related to the application.
* Developed **API** for using **AWS Lambda** to manage the servers and run the code in the AWS.
* Used **AWS CodeCommit, CodeDeploy, and CodePipeline** for implementing CI/CD.
* Unified build and test environments and provide efficient deployment method by **Docker** across different services. Combined it with **Jenkins pipeline** for CI/CD purpose.
* Optimized database access using **JPA/Hibernate** with custom native queries, achieving a 25% reduction in query execution time.
* Designed and normalized relational database schemas (**MySQL, PostgreSQL**) for enterprise applications.
* Integrated Java backends with front-end frameworks (**React.js/Angular**) via **REST** and WebSockets for real-time updates.
* Worked with **CORS, JSON** serialization/deserialization, and API versioning for seamless frontend-backend communication.
* Managed complex build configurations using **Maven** and **Gradle** with custom plugin integrations.
* Used **Git** (GitHub, Bitbucket) for version control and pull request code reviews in Agile teams.
* Used **Spring Core Annotations** for Spring Dependency Injection and Spring MVC for building REST APIs.
* Worked on RESTful Web Services for transferring JSON data between two interfaces.
* Configured **Spring Boot Security** with **OAuth2** to grant permissions for different level resources.
* Used **Hibernate** as an ORM tool to interact with the database.
* Involved in development of **POJO classes** and writing Hibernate query language (HQL) queries.
* Performed asynchronous messaging using **JMS** and created Queue for point to point messaging.
* Developed the web interface using **HTML5, CSS3, JavaScript, JQuery, ReactJS, and Bootstrap**.
* Developed various screens for the front end using **ReactJS** and used various predefined components from **NPM (Node Package Manager) and Redux Library**.
* 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.
* Performed unit testing using **JUnit** and created test cases using **Mockito** framework.
* Used **Log4j** logging mechanism for logging. Used **SVN** for versioning and concurrent access to the project files.
* Resolved Spring framework **artifacts** and code build & deployment with **Maven**.
* Used **JIRA** for defect and issue tracking and used **Jenkins** for continuous integration/continuous deployment.

**Environment:** Java 8, J2EE, Spring MVC, Spring Security, OAuth2, Spring AOP, Hibernate 3.0, AWS, Docker, HTML, CSS, JavaScript, ReactJS, Redux, NPM (Node Package Manager), JMS, SOAP UI, RESTful API, Axis2, Maven, Jenkins, DB2, Selenium, Log4j, SVN, JIRA, Apache Tomcat 6.0, JUnit, Mockito

**GGK Tech, Hyderabad, India Jul 2015 to Aug2017**

**Java Developer 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.
* Implemented **RESTful API** using **Spring MVC** to do the backend data transfers.
* 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.
* Used **Spring Security** with **JWT** for the RESTful API to provide application security.
* 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.
* Set 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**.
* Established database connectivity using **JDBC** and mapping the objects into **MySQL** database.
* Interacted with application testers to review system defects and provided comprehensive fixes.
* Addressed defects identified in DEV, QA, UAT phases and tracked them using **JIRA**.

**Environment:** Java 8, J2EE, Spring MVC, Spring DAO, Spring Security, Spring IOC, HTML, CSS, JSF, JQuery, JavaScript, AJAX, JWT, JDBC, SOAP, SOA, JMS, JPA, JAXB, Hibernate, EJB, MySQL, JBoss 6.0, Ant, CVS, Junit

**Menlo Technologies, Hyderabad, India Aug 2013 to Jun 2015 Java/J2EE Developer**

**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.
* Wrote 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**.
* Used **JSP** as the view, **Servlet** as the controller and **Java Beans** as the model to implement the Spring MVC to handle HTTP requests.
* Created the Registry Resources in WSO2 Governance Registry which would become a single resource for the registered schemas.
* Used **XML technologies** like XPath, XSTL for preparing reports.
* Developed dynamic and browser compatible pages using **HTML, CSS, JQuery, Angular JS and JavaScript**.
* Developed the login portal using framework AngularJS along with HTML5, CSS3.
* Developed WSO2 Carbon based Data Service layer to enable **CRUD operations** on Meta Content Repository.
* Developed **Dispatcher-Servlet** with web.xml files for proper routing applications.
* Invoked **JavaMailSender API** for automatically sending email functions.
* Deployed the application on **WebLogic Application Server**.
* Used **JSON and XML** to develop REST web services which are utilized by internal call in web and mobile applications.
* Monitored the error logs using **Log4j** and fixing the problems.
* Wrote **JUnit** test cases for unit testing of the application. Wrote **ANT scripts** for building application artifacts. **Environment:** Java 7, J2EE, JSP, HTML, CSS, JavaScript, Struts 2.0, EJB 2.1, Servlets, Java Beans, WSO2, JSON, XML, JDBC, JNDI, JMS API, JUnit, Log4j, WebLogic 7.0, Rational Rose