**A hexagon with text and a blue and white hexagon

Description automatically generatedNithish japala**

(314) 375-6075 | [japalanithish@gmail.com](mailto:japalanithish@gmail.com)

[LinkedIn](https://www.linkedin.com/in/nithish-japala-28b366339/) | [GitHub Portfolio](https://poojithg14.github.io/poojithgavini.github.io/) | [Medium](https://medium.com/@japalanithish)

**PROFESSIONAL Summary**

* **Highly skilled Java Full Stack** **Developer** with over 9 years of experience in developing, testing, and deploying enterprise-grade applications using **Java, Spring Boot, Angular, ReactJS,** and **cloud platforms,** ensuring end-to-end application lifecycle management with a focus on performance, security, and scalability.
* Developed dynamic and responsive user interfacesusing **Angular,** focusing on creating modular components, implementing **Angular Material**, and optimizing application performance with **Lazy Loading** and **Routing** mechanisms for efficient navigation and reduced load times.
* Built highly interactive frontends using **ReactJS**, **leveraging React Suspense, React Router,** and **Hooks** to manage state, optimize rendering, and ensure seamless navigation across application workflows.
* Implemented state management with **Redux** and **RxJS** Observables, enabling real-time updates, API integration, and consistent data flow across application components.
* Integrated secure authentication and authorization using **OAuth 2.0, JWT,** ensuring role-based access control and compliance with **HIPAA** standards.
* Designed and implemented scalable **RESTful APIs using Spring Boot** and **Java 17/11/8**, leveraging modern Java features such as **Records**, **Sealed Classes, lambda expressions, parallel streams, var keyword**
* Built modular microservices with **Spring Boot,** ensuring fault isolation, scalability, and efficient API communication via API Gateways for secure routing, monitoring, and traffic control.
* Designed **batch processing** workflows using **Spring Batch** for scheduled reporting, analytics exports, and periodic data cleanup tasks.
* Worked extensively with **PostgreSQL, Oracle, MySQL, MongoDB, Cassandra , and Couchbase** optimizing database performance, query execution, and ensuring data consistency across complex transactions.
* Implemented advanced caching strategies using **Redis**, improving API latency, enhancing data retrieval efficiency, and managing distributed cache systems effectively.
* Integrated **RabbitMQ** for real-time data synchronization and asynchronous message processing, enabling efficient communication between services.
* Leveraged **Kafka** forevent-driven architecture, ensuring scalable, high-throughput, and fault-tolerant message streaming pipelines.
* Experience in **Amazon Web Services** (Amazon EC2, Amazon S3, Amazon RDS, Amazon Elastic Load Balancing, Amazon SQS, AWS Identity and Access Management, AWS Cloud Watch, Amazon EBS, and Amazon Cloud Front).
* Managed scalable deployments on **Google Cloud Platform (GCP)**, leveraging GCP Compute Engine for backend services and GCP API Gateway for secure routing and traffic control.
* Experienced in leveraging Azure cloud services for application deployment, management, and scalability. Proficient in implementing **Azure Service Bus** for efficient messaging**, Azure Kubernetes Service (AKS)** for container orchestration, and **Azure Blob Storage** for storing data.
* Established **CI/CD pipelines with Jenkins, streamlining code builds, testing,** and **deployment** workflows to ensure **continuous delivery** and **code quality.**
* Containerized backend and frontend services using **Docker** and orchestrated them with **Kubernetes clusters**, enabling consistent builds, seamless deployment, auto-scaling, and optimized resource management across environments.
* Monitored application health and performance metrics using **Dynatrace** and **SonarQube**, proactively identifying **bottlenecks** and addressing performance issues.
* Implemented unit tests using **JUnit** and **Mockito** for backend reliability and error handling, while conducting front-end integration testing with **Selenium** to ensure functional workflows and cross-browser compatibility.
* Conducted thorough **code reviews**, applied secure coding standards, and performed vulnerability scans to mitigate security risks proactively.

**Tools and technologies**

* Programming languages: Java (17/11), J2EE 1.8, Python,JavaScript
* Technologies & Frameworks: Spring Boot, Spring MVC, Spring Security, Hibernate, JDBC, JSP, JSTL, Node.js, Angular, ReactJS, Redux, RxJS, Material-UI, Bootstrap.
* Database: Redis, Azure Blob Storage, MongoDB,Cassandra,Oracle, MySQL, MS SQL PL/SQL, S3
* Web Services: HTML5, CSS3, JavaScript, Angular, React JS, SOAP, REST
* Messaging: Azure Service Bus, RabbitMQ, Kafka.
* CI/CD & DevOps: Jenkins, JIRA, Docker, Kubernetes.
* Version Control System: Git, Perforce & SVN.
* IDE used: Visual Studio, Rider, Eclipse, NetBeans.
* Tools: SonarQube, Robo Manager, PuTTY, WinSCP, Postman.
* Monitoring Tools: Dynatrace, AWS CloudWatch, Splunk
* Cloud Technologies: AWS, Microsoft Azure, GCP
* Operating Systems: Windows and Linux/Unix.

**WORK Experience**

* **Full Stack Java Developer – Enqbator, Troy, MI Jan 2023 – Aug 2024  
  Project: Event Sponsorship Portal (ESP)**

The Event Sponsorship Portal is a comprehensive platform designed to empower event sponsors with enhanced visibility, advanced analytics, and seamless interaction capabilities on the conference website. It provides tools for sponsor branding, dynamic advertisement placement, and actionable insights to track engagement and return on investment (ROI).

**Roles and responsibilities**:

* + Developed dynamic and responsive user interfaces using **ReactJS and Material-UI**, delivering an intuitive and user-friendly experience for sponsor profile management and advertisement placements.
  + Built reusable React components and effectively **utilized React Hooks** (useState, useEffect, useContext, useReducer**)** to create modular, maintainable, and dynamic UI functionalities across various sponsor dashboards.
  + Implemented **React Router** for seamless client-side routing, enabling dynamic navigation between sponsor dashboards, campaign analytics, and engagement insights, enhancing user experience with efficient routing and URL management.
  + Optimized application performance using **React Lazy Loading and Suspense**, reducing initial page load times for data-heavy components and ensuring smoother rendering of analytics sections.
  + Enhanced state management with **Redux**, centralizing sponsor campaign data, authentication details, and global application state, ensuring predictable updates, better debugging, and streamlined communication between components.
  + Built scalable and secure **RESTful APIs using Spring Boot**, managing sponsor profiles, advertisement placements, and analytics data with adherence to REST standards for consistent communication between frontend and backend services.
  + Designed **Microservices Architecture using Spring Boot**, enabling independent deployments, fault isolation, and horizontal scalability for better system resilience.
  + Utilized **Java 17's Records and Sealed Classes** for defining immutable data structures and enforcing strict hierarchies in sponsor-related entities, enhancing code readability and maintainability.
  + Implemented **OAuth 2.0** and **JWT-based authentication** and authorization, ensuring secure, role-based access control and protection of sensitive sponsor data.
  + Automated scheduled tasks and reporting workflows using **Spring Batch**, optimizing analytics exports, periodic data cleanup, and scheduled report generation.
  + Integrated **Apache Kafka** for asynchronous message streaming, enabling efficient communication between microservices, ensuring reliable event-driven updates for campaign statuses, logs, and notifications.
  + Utilized **MongoDB** for flexible, schema-less data storage, efficiently handling sponsor campaign metadata and storing engagement logs without strict schema enforcement.
  + Managed relational data storage using **PostgreSQL**, optimizing complex queries, implementing stored procedures, and ensuring data integrity for campaign performance metrics and sponsor engagement records.
  + Deployed microservices on **AWS EC2** with **ELB** for scalability, used **S3** for secure asset storage, and **EBS** for persistent storage.
  + Managed access with **IAM**, enabled reliable queuing with **SQS**, monitored performance via **CloudWatch**, and optimized delivery using **CloudFront** for seamless operations.
  + Established **CI/CD pipelines using Jenkins**, streamlining automated builds, continuous testing, vulnerability scans, and deployment workflows to ensure faster release cycles.
  + Managed source code repositories on **GitHub**, enabling version control, effective collaboration, and structured pull request reviews.
  + Developed robust unit and integration tests using **JUnit** and **Mockito**, ensuring backend services were reliable, met functional requirements, and had minimal regression errors.
  + Followed the **Agile Scrum methodology**, actively participating in daily stand-ups, sprint planning, retrospectives, and managing tasks through **JIRA** for effective project tracking and team collaboration.
  + Embraced **Test-Driven Development (TDD)** principles, writing tests before implementation to ensure functionality validation at every stage of development.
* **Full Stack Java Developer – Gold Coast Health Plan, Camarillo, CA  April 2021 – December 2022**

**Project: Patient Health Management System**  
 The Patient Health Management System (PHMS) is an integrated platform designed to streamline healthcare workflows, improve patient engagement, and enhance operational efficiency. The platform enables secure access to patient records, manages appointment scheduling, automates insurance claims processing, and facilitates secure communication between healthcare providers and patients. It leverages a robust backend for data processing and a user-friendly frontend interface for seamless interaction.

**Roles and responsibilities:**

* + Engaged in all stages of the **Software Development Life Cycle (SDLC),** including requirement gathering, design, analysis, development, testing, and deployment, ensuring security best practices at every phase.
  + Developed various screens for the front end using **ReactJS** and used reusable components to promote code modularity and maintainability.
  + **Utilized React Hooks (useState, useEffect, useContext)** to efficiently manage local component states, side effects, and global data sharing across nested components.
  + **Implemented dynamic form validations using React Hook Form**, ensuring accurate patient data entry, field-level validations, and clear error messages for appointment and claims workflows.
  + Integrated **Axios** for handling HTTP requests, ensuring secure and efficient communication between the frontend and backend, and enabling reliable data retrieval for patient records, appointments, and insurance claims.
  + Implemented **Redux** for centralized state management, ensuring predictable data flow and efficient handling of authentication tokens, appointment data, and patient records, enhancing overall application stability and responsiveness.
  + Architected and implemented a microservice-based architecture to enhance scalability, maintainability, and deployment flexibility.
  + Developed secure **RESTful APIs using Spring Boot,** adhering to best practices for secure API design and communication protocols.
  + Utilized **Java 11** features, including the modern HTTP Client API for efficient API communication and advanced garbage collection optimizations, resulting in cleaner, more efficient, and high-performance backend code.
  + Implemented Spring Security with **OAuth 2.0** to enforce fine-grained access controls and secure sensitive patient data endpoints.
  + Configured **MongoDB** with Spring Data for seamless integration with healthcare data repositories, ensuring encryption at rest and secure data management.
  + Designed and optimized **SQL queries, PL/SQL stored procedures**, and functions to facilitate secure data retrieval, validation, and manipulation in Oracle databases.
  + Configured **AWS IAM roles and S3 bucket policies**, enabling **fine-grained access control** and **secure data storage** compliant with healthcare standards.
  + Deployed containerized **Spring Boot** applications on **AWS ECS** and **ECR**, ensuring auto-scaling, load balancing, and secure storage of sensitive documents in **AWS S3** with **KMS encryption**.
  + Performed continuous integration and deployment (CI/CD) using **Jenkins**, automating code analysis, vulnerability scanning, and ensuring deployment pipelines adhered to security compliance guidelines.
  + Monitored application performance and security metrics using **Dynatrace** and **SonarQube**, ensuring vulnerabilities and performance bottlenecks were proactively addressed.
  + Followed Agile Scrum practices, actively participating in sprint planning, backlog grooming, and daily stand-ups, and tracked progress using **JIRA.**
  + Integrated **SLF4J** for consistent and efficient logging across backend services, ensuring clear monitoring, debugging, and traceability of application workflows.
  + Conducted **code reviews** and implemented best practices for secure coding, identifying potential vulnerabilities, and applying patches proactively.
  + Created detailed technical **documentation** for system architecture, API contracts, and deployment configurations, ensuring transparency and future maintainability.
* **Software Engineer – Pinnacle Financial Partners, Nashville, TN Jan 2020 – Apr 2021**

**Project: Document Management and Approval Workflow.**

The Document Management and Approval Workflow System is an internal platform designed to streamline the creation, approval, and storage of critical documents, including contracts, compliance reports, and internal communications. Built a system that enables secure document storage, efficient approval workflows, version control, and real-time notifications. It ensures seamless collaboration, transparency, and accountability throughout the document lifecycle, while automated workflows and audit trails reduce manual intervention and improve compliance adherence across the organization.

**Roles and responsibilities:**

* + Used **AngularJS** and its library functions for client-side logical implementation across the entire application.
  + Implemented **Angular routing** and **lazy loading** to optimize navigation and reduce initial load times, improving overall application efficiency.
  + Designed and developed custom **Angular** components and services, applying best practices for modularity and maintainability.
  + Designed and implemented a Java engine and API to enable direct communication between **front-end JavaScript** and **AngularJS** with server-side Java methods.
  + Developed and deployed **RESTful APIs** using Spring Boot, enabling seamless integration with front-end applications and enhancing overall system efficiency.
  + Utilized **Spring Boot’s** **auto-configuration and dependency injection** features to simplify application setup and streamline development workflows.
  + Used the **Spring framework** for **injecting DAO** and Bean objects with component auto-wiring, while also designing and developing batch jobs using the Spring Batch architecture for efficient task execution and data processing.
  + Implemented microservices architecture with **Spring Boot**, designing scalable and resilient services with robust communication and fault-tolerance mechanisms.
  + Integrated **Kafka** for real-time event-driven communication, enabling document approval triggers, live status updates, and reliable asynchronous messaging between services for efficient document processing.
  + Optimized database performance using **Cassandra** for large-scale data and Oracle for transactional data, ensuring seamless integration with Java microservices via Spring Data.
  + Utilized **Azure services** for scalable application deployment, ensuring high availability and reliability.
  + Implemented **Azure Blob Storage** for durable document storage and **Azure Key Vault** for securely managing application secrets, certificates, and keys.
  + Deployed Spring Boot-based microservices and **Docker** containers in an internal cloud server.
  + Used **Agile SDLC** methodology with Scrum as a framework for requirement gathering, analysis, design, development, and testing while upgrading the project.
* **Software Engineer - Office Depot, Boca Raton, FL April 2017 – Dec 2019**

**Project: Supplier Management Portal**

The Supplier Management Portal is an internal platform designed to streamline supplier data management, contract handling, and compliance documentation. The portal ensures secure document uploads, automated approval workflows, and vendor performance tracking. Real-time notifications via RabbitMQ keep stakeholders updated on approvals and deadlines, while AWS S3 securely manages document storage and retrieval. This platform enhances supplier collaboration, improves transparency, and reduces manual overhead, ensuring efficient and compliant supplier operations.

**Roles and responsibilities:**

* + Developed dynamic and responsive user interfaces using **Angular** and **Angular Material**, ensuring seamless navigation and consistent layouts across supplier dashboards, contract forms, and compliance tracking modules.
  + Implemented state management using **RxJS** Observables, enabling real-time updates, efficient API communication, and consistent data flow between components for supplier records and approval workflows.
  + Utilized **Angular Routing** for efficient client-side navigation, enabling smooth transitions between modules like supplier onboarding, document uploads, and performance analytics.
  + Built dynamic forms using **Angular Reactive Forms**, implementing advanced validation rules, error handling, and dynamic form controls for supplier registration and compliance documentation.
  + Used the features of the **Spring Core layer**, **Spring ORM** layer, and **Spring DAO** support layer to develop the application.
  + Built scalable **RESTful APIs** using Java 11 and Spring Boot, handling business logic, sponsor management workflows, and campaign analytics data efficiently.
  + Developed controllers, repositories, service modules, and form beans with validations, and persisted them in the database using **JDBC** and **Hibernate**, ensuring seamless database connectivity through Hibernate configuration files.
  + Integrated **Redis** for in-memory caching of frequently accessed data, reducing database query latency and improving backend response times.
  + Configured **Google Cloud API Gateway** to securely route API traffic, enforce access policies, enable throttling, and monitor API performance.
  + Deployed backend services on **Google Cloud Platform (GCP**) using Compute Engine and App Engine, ensuring high availability and scalability.
  + Participated in daily stand-ups, sprint planning, and retrospectives following Agile Scrum methodology, tracking tasks using **JIRA**.
  + Written **Maven scripts** to build the entire module and deploy it on the WebLogic application Server.
  + Implemented the **JUnit framework** to write test cases for different modules and resolved the test findings.
* **Associate Software Engineer – KLA Tencor - Chennai July 2015 – April 2017**

**Project: Internal Workflow Automation Tool**

The Internal Workflow Automation Tool streamlines internal processes such as material procurement, equipment maintenance, and approval workflows by automating repetitive tasks and enabling seamless collaboration across departments. It supports multi-level approval workflows, real-time notifications, and detailed audit trails, ensuring transparency and accountability at every stage. By minimizing manual intervention and standardizing workflows, the tool accelerates approval cycles, reduces errors, and improves overall operational efficiency within the organization.

**Roles and responsibilities:**

* Developed **Java APIs** to facilitate seamless communication with Java Beans, ensuring modular design and efficient backend operations.
* Designed and implemented **PL/SQL stored procedures** and triggers to optimize database operations, ensuring transactional integrity and performance.
* Enhanced product features by optimizing **SQL queries** and refining database schemas for improved efficiency and scalability.
* Gathered and analyzed **user requirements**, translating them into actionable technical specifications for effective design and development.
* Performed comprehensive **testing,** including functional, integration, system, and validation phases, to ensure software reliability and performance.
* Created **JUnit test cases** for regression testing and integrated them into the Maven build process for automated validation.
* Implemented the **Log4J logging framework** to enable efficient monitoring, debugging, and performance tracking of applications.
* Reviewed **code** and **technical documentation**, ensuring adherence to coding standards and clarity in project artifacts.
* Utilized **Maven** as a build management tool to automate dependency management, build processes, and artifact generation efficiently.

**Education**

* Saint Louis University, St. Louis, MO January 2023 - Dec 2024
  + - Master of Science in Computer Science

**Certifications**

* AWS Certified Solutions Architect – Associate