Dinesh K

Senior Software Engineer

dineshk42224@outlook.com | +1 469-453-6462

Professional Summary

- Senior Software Engineer with **10+ years of experience** designing and building high-performance, cloud-native applications across **healthcare**, **retail**, **e-commerce**, and **financial sectors**, delivering measurable improvements in scalability, operational efficiency, and user engagement.
- Strong expertise in **Frontend development** using **React.js**, **Angular**, **JavaScript**, **TypeScript**, **HTML5**, **CSS3**, **NgRx, RxJS**, and **Redux**, delivering responsive and accessible user interfaces for enterprise platforms.
- Skilled in Backend development with **Java (Spring Boot, J2EE, Jakarta EE)**, **Node.js (Express.js)**, and **Python (FastAPI, Flask)**, designing robust **RESTful** and **GraphQL APIs** to support microservices-based architectures and real-time integrations.
- Proficient in managing data-centric applications using **SQL databases** (**MySQL**, **PostgreSQL**, **Oracle**), **NoSQL databases** (**MongoDB**, **DynamoDB**), and **Snowflake, Amazon Redshift**, with an emphasis on performance, scalability, and reliability.
- Experienced in implementing **caching strategies** using **Redis** and **Memcached**, improving response times and reducing backend load under peak user traffic.
- Hands-on experience with DevOps practices and CI/CD pipelines using **GitHub Actions**, **Jenkins**, **GitLab CI**, **Docker**, **Terraform**, and **Ansible**, enabling secure, automated deployments across environments.
- Proven success deploying distributed systems on **AWS, GCP** and **Azure,** supporting hybrid and multi-cloud strategies with infrastructure-as-code.
- Advanced knowledge of **messaging systems** such as **Apache Kafka**, **RabbitMQ**, and **Amazon SQS**, enabling reliable, event-driven communication between loosely coupled services.
- Applied modern **monitoring and observability** practices using **CloudWatch**, **Splunk**, **ELK Stack**, **Azure Monitor**, **Prometheus**, and **Grafana**, ensuring proactive issue detection and SLA adherence.
- Delivered data-driven platforms by engineering **ETL pipelines** with **AWS Glue**, **Apache Airflow**, and **Snowflake/Redshift**, and integrating **real-time analytics** dashboards with **Amazon QuickSight** and **Tableau**.
- Contributed to AI/ML initiatives using **Amazon SageMaker**, **Bedrock**, and **RAG architectures**, enabling predictive stock forecasting and natural language querying in operational workflows.
- Enabled healthcare data interoperability by integrating **FHIR** and **HL7v2** standards into backend APIs and vendor communication workflows within regulated lab environments.
- Focused on **testing** with tools like **JUnit**, **Jasmine**, **Jest**, **Mocha**, **Mockito**, and **Postman**, ensuring maintainable, high-quality codebases across services and reducing regression risks.
- Strong understanding of **security best practices**, including **OAuth2**, **JWT**, **RBAC**, **SSO**, and **API hardening** to maintain data privacy and system integrity.
- Familiar with industry compliance standards including **HIPAA**, **SOC 2**, and internal audit controls for secure application development and data protection in regulated environments.
- Adept in **Agile environments (Scrum, SAFe**), collaborating with cross-functional teams to translate domain-specific requirements into scalable technical solutions with tangible business outcomes.

Technical Skills

- **Frontend:** Angular, React.js, Redux, NgRx, RxJS, HTML5, CSS3, JavaScript, TypeScript, Bootstrap, Tailwind CSS, D3.js, JSP
- **Backend & APIs**: Java, J2EE, Spring Boot, Spring MVC, Jakarta EE, Node.js, Express.js, Python (FastAPI, Flask), REST APIs, GraphQL, Microservices, FHIR, HL7v2
- **Databases:** PostgreSQL, MySQL, OracleDB, MongoDB, DynamoDB, Snowflake, Amazon Redshift
- **Data Engineering & Visualization:** Apache Airflow, AWS Glue, ETL Pipelines, Amazon QuickSight, Tableau
- **AI/ML:** AWS SageMaker, AWS Bedrock, GenAI, LLM, RAG, Predictive Analytics, Forecast Modeling
- **Caching:** Redis, Memcached

- **Messaging & Event Streaming:** Apache Kafka, RabbitMQ, Amazon SQS/SNS
- **DevOps & CI/CD:** GitHub Actions, GitLab CI, Jenkins, Docker, Terraform, Ansible, Kubernetes (EKS, AKS, GKE), Helm, Shell Scripting
- **Cloud Platforms:**
    - **AWS:** Lambda, API Gateway, Step Functions, S3, EKS, SageMaker, Bedrock, CloudWatch
    - **Azure:** Azure Functions, App Services, Azure DevOps, API Management, Blob Storage, Key Vault
    - **GCP:** GCP Cloud Functions, GCP Storage, Cloud Logging, Secret Manager
- **Monitoring & Logging:** Prometheus, Grafana, CloudWatch, Splunk, ELK Stack, Azure Monitor, GCP Logging
- **Security & Compliance:** Spring Security, OAuth2, JWT, RBAC, SSO, Resilience4j, HIPAA, SOC 2, POPIA, API Gateway Policies
- **Testing & Quality:** JUnit, Mockito, Jest, Mocha, Chai, Karma, Jasmine, Postman, Swagger, OpenAPI
- **Version Control & Collaboration:** Git, GitHub, Bitbucket, Confluence, JIRA

Certifications

- Microsoft Certified: Azure AI Engineer Associate | Credential: 7D91AF06EB235B8F
- AWS Certified Solutions Architect – Associate | Credential: 352f2d1915f8454b98c0442f74d694a1

Work Experience

Senior Software Engineer | Labcorp Jan 2023 – Present

Burlington, North Carolina

Project Description:

Developed and optimized IT-driven inventory management solutions to streamline Labcorp's clinical laboratory operations. Focused on enhancing real-time tracking and automation of lab supplies, reducing stockouts and improving operational efficiency. Enabled data-driven decision-making through integrated analytics, predictive models, and workflow automation, supporting critical lab operations across multiple locations.

Responsibilities:

- Built highly scalable **Spring Boot (Java)** microservices, containerized with **Docker** and deployed on **Kubernetes (EKS)**, enabling real-time inventory updates, expiry validation, and stock threshold monitoring across distributed lab environments.
- Developed **Node.js (Express.js)** and **Python (FastAPI)** microservices with **FHIR** and **HL7v2** standards, deployed on **AWS Lambda** to automate vendor workflows and reduce procurement delays by 25%.
- Built dynamic **Angular** interfaces using **NgRx** for state management, **RxJS** for real-time updates, and **D3.js** to visualize live stock data, helping lab technicians track usage trends, reduce discrepancies, and improve visibility by 20%.
- Implemented **expiry detection alerts in Angular** using **RxJS observables, Angular HTTP Clients**, and modular services to fetch real-time stock data via **GraphQL**, enabling UI-level warnings based on configurable thresholds and driving proactive inventory decisions.
- Standardized build processes using **npm** for **Angular UI** components, **Node.js** services and **Gradle** for **Spring Boot** services, streamlining dependency management.
- Designed and implemented **GraphQL** and **REST APIs** using **Spring Boot**, with support for **FHIR** resource structures and **HL7v2** message translation, unifying data access for **Angular** dashboards, analytics pipelines, and admin interfaces while optimizing logic with **Java Streams** and **Optionals**.
- Tuned batch workflows in **Spring Boot** using **lazy-loading**, **Redis**, and **asynchronous execution**, reducing inventory reconciliation time by over 40%.
- Used **Redis** as an in-memory cache for frequently accessed records, reducing **DynamoDB** and **PostgreSQL** read pressure and improving SLA compliance by 60%.

- Streamlined stock lifecycle events using **Apache Kafka** and **Spring Boot** ensuring low-latency updates, seamless message delivery, and resilient inter-service communication across supply chain endpoints.
- Developed scalable **ETL pipelines** using **AWS Glue**, transforming data from **DynamoDB** and **PostgreSQL** into **Redshift** for analytics, model training, and compliance audits.
- Built predictive analytics models in **Amazon SageMaker** and **Bedrock**, trained on historical data from **Redshift**, helping cut urgent stockouts and unplanned orders by 25%.
- Integrated **LLM** powered **insight query service** using **Claude** via **AWS Bedrock** and a **Redshift-backed RAG architecture**, enabling lab staff to query inventory trends and anomalies using natural language.
- Enhanced **Kafka** alerting with **GenAI** summarization via **Bedrock**, enabling rich notifications with inventory, predictions, and vendor delays, boosting proactive reorders by 20%.
- Built secure CI/CD automation using **GitHub Actions**, integrating workflows directly with **GitHub repositories** to enforce code quality via **linting, Docker image scans**, **unit tests**, and **blue-green deployments**.
- Managed microservice builds with **Docker**, orchestrated deployment pipelines on **EKS**, and maintained rollback consistency across QA, staging, and production environments.
- Defined **Terraform** and **Ansible**-based infrastructure-as-code templates to provision **Lambda**, **S3**, **EKS**, **Redshift**, **SageMaker**, and **Secrets Manager**, ensuring reproducible, compliant multi-region environments and consistent configuration across cloud and container nodes.
- Used **AWS Step Functions** with **Lambda**, **SNS**, and **S3** to automate reorder approvals, vendor coordination, and exception alerts, reducing manual escalations by 30%.
- Built real-time dashboards with **Amazon QuickSight** on top of **Redshift**, visualizing lab-specific usage trends, reorder thresholds, and supply risk forecasts for data-driven decisions.
- Secured API traffic using **OAuth2**, **JWT**, and **RBAC**, integrated with **API Gateway** and **Spring Security**, ensuring HIPAA-compliant access control and request validation.
- Partnered with lab ops, procurement, and compliance teams to align architecture with **HIPAA**, **SOC 2**, and audit requirements, enabling secure, transparent automation workflows.
- Implemented observability using **Spring Boot Actuator**, **Prometheus**, and **Splunk**, exposing live service health and reducing incident detection time by over 40%.
- Engineered fault-tolerant workflows using **Kafka**, **Redis**, and **Step Functions**, incorporating **Resilience4j** for retries, circuit breakers, and error isolation to lower failure rates by 15%.
- Achieved over **90%+ unit test coverage** with **JUnit**, **Mockito**, and **Jasmine**, and enforced PR-level gates using GitHub Actions pipelines.
- Documented architectural flows and service dependencies (**Spring Boot, Kafka, SageMaker, Redshift**), supporting successful security audits and **SOC 2** compliance readiness.
- Led cross-team technical sessions on topics like secure API development, containerized deployment, observability patterns, and best practices using **Spring Boot**, **Docker**, and **AWS-native tools**.
- Participated actively in **Agile ceremonies**, collaborating with stakeholders across sprints to ensure timely, high-quality delivery of lab inventory features and platform enhancements.

Environment: Java, Spring Boot, Jakarta EE, Node.js, Express.js, Python, FastAPI, Flask, GraphQL APIs, RESTful APIs, FHIR, HL7v2, Angular, NgRx, RxJS, D3.js, JavaScript, npm, Gradle, PostgreSQL, DynamoDB, Redshift, Redis, Apache Kafka, AWS Lambda, AWS Glue, SageMaker, Bedrock, Claude, RAG, S3, API Gateway, AWS Step Functions, SNS, Docker, Kubernetes (EKS), Terraform, Ansible, GitHub Actions, Prometheus, Splunk, CloudWatch, Amazon QuickSight, OAuth2, JWT, RBAC, Spring Security, Resilience4j, JUnit, Mockito, Jasmine, Agile

Senior Software Engineer | Albertsons Jan 2019 – Dec 2022

Dallas, TX

Project Description:

Redesigned Albertsons’ retail operations platform to replace outdated systems and improve how stores manage inventory, pricing, and promotions. The new solution helped store teams track stock levels more accurately, respond faster to low-stock alerts, and roll out pricing changes across locations in real time. It also supported better coordination between central office teams and stores, enabling faster decision-making and improved customer experience on the retail floor.

Responsibilities:

- Migrated legacy monolithic applications into distributed **Spring Boot (Java 11)** and **Node.js** microservices using **Docker** containers, deployed across **Azure Kubernetes Service (AKS)** and **Google Kubernetes Engine (GKE)** to improve release velocity and cross-cloud resilience for 2,000+ store locations.
- Developed and deployed lightweight **Node.js (Express.js)** services via **Azure Functions** and **Google Cloud Functions** to handle low-latency operations like digital flyer generation and receipt storage.
- Built reusable UI components in **React.js** using **React Hooks** and styled them with **Tailwind CSS** to deliver real-time inventory tracking, shelf capacity alerts, and price override interfaces for store employees.
- Deployed static frontend assets to **Azure CDN** and **GCP Cloud CDN**, reducing load times for **React** dashboards by 40% and improving access across distributed retail locations.
- Integrated **D3.js** charts into **React.js** dashboards to visualize promotions, inventory aging, and product turnover, helping store managers optimize shelf space.
- Standardized build and dependency workflows using **npm** for React component libraries and **Gradle** for Spring Boot microservices, enabling consistent packaging.
- Routed all APIs through **Azure API Management** and **Google Cloud API Gateway**, enabling access logging, quota enforcement, and secured routing between microservices and internal portals.
- Integrated **OAuth2**, **JWT**, and **SSO** for internal portals used by store and regional managers, improving login efficiency and reducing IT support tickets by 15%.
- Protected sensitive endpoints with **Spring Security** and centralized access controls, isolating stock movement APIs and pricing changes by user roles and minimizing data leakage risk.
- Developed secure **RESTful** and **GraphQL APIs**, implemented with **Spring WebFlux** and **Apollo Server**, enabling reactive client-server communication and reducing client-side data over-fetching by 30%.
- Applied **Mono** and **Flux** from **Project Reactor** within backend services to handle concurrent pricing requests, inventory syncs, and forecast queries without blocking I/O threads, increasing throughput under load.
- Used **Redis** to cache price rules, stock thresholds, and promotional flags, improving homepage load times by 55% during regional campaigns and reducing database round-trips.
- Streamed inventory update and promotional start events using **Apache Kafka**, enabling near real-time sync between store stock systems and regional dashboards.
- Designed **Kafka** consumer services to process inventory thresholds and feed directly into replenishment triggers, avoiding 20% of potential out-of-stock situations during peak sales.
- Managed credentials and service secrets using **Azure Key Vault** and **GCP Secret Manager**, supporting automated rotation and aligning with **SOC2** audit requirements.
- Used **Azure Blob Storage** and **GCP Cloud Storage** for secure archival of transactional receipts, promo assets, and pricing audit logs, ensuring high availability and regulatory compliance.
- Built a **ETL pipelines** using **Apache Airflow** to pull data from **OracleDB** and **MongoDB**, store it in **Azure Blob Storage** and **GCP Cloud Storage**, and load it into **Snowflake** for unified reporting.
- Configured **Tableau** and **Looker** dashboards using **Snowflake** data to surface out-of-stock trends, **ROI**, and replenishment KPIs, accelerating decision cycles.
- Established CI/CD pipelines in **GitLab CI**, incorporating **Docker image builds**, **Helm chart deployments**, static code checks, and automated testing, which reduced release errors by 25%.
- Built integration and unit test suites using **JUnit**, **Mockito**, **Mocha**, and **Jest**, achieving 90%+ coverage and cutting QA rework cycles in half over four major releases.
- Monitored performance with **Prometheus** and **Grafana**, while using **Azure Monitor** and **GCP Cloud Logging** for system logs, availability checks, and incident alerting.
- Authored and published **OpenAPI** and **GraphQL schema documentation** using **Swagger** and **Apollo Docs**, streamlining partner onboarding and reducing integration issues.
- Mentored engineers on **React patterns**, **Spring reactive programming**, **Kafka event streams**, and multi-cloud deployment practices, leading to a 25% improvement in pull request approval turnaround.
- Collaborated with product owners and merchandising leads in **Agile ceremonies** to align sprint goals with seasonal marketing strategies and stock management priorities.

Environment: Java, Spring Boot, Node.js, Express.js, React.js, Tailwind CSS, RESTful APIs, GraphQL, MongoDB, OracleDB, Redis, Snowflake, Apache Kafka, Apache Airflow, Docker, Kubernetes (AKS, GKE), Terraform, GitLab CI, Azure Functions, GCP Cloud Functions, Azure API Management, Google Cloud API Gateway, Azure Blob Storage, GCP Cloud Storage, Azure Key Vault, GCP Secret Manager, Spring Security, OAuth2, JWT, SSO, Prometheus, Grafana, Azure Monitor, GCP Logging, Tableau, Looker, JUnit, Mockito, Postman, Mocha, Jest, Swagger, Agile

Software Engineer | Sanlam Corporate May 2016 – Dec 2018

Hyderabad, TG, India

Project Description:

Developed a digital platform for Sanlam's new Group Risk and Employee Benefits Product Suite, enabling corporate clients to manage policies, track premium contributions, and process claims efficiently. Focused on delivering seamless self-service portals, automating premium calculations, and integrating with external actuarial and payment systems to support accurate and timely operations.

Responsibilities:

- Built responsive and reusable **Angular components** styled with **Bootstrap**, **SASS**, and **LESS**, enabling HR users to manage enrolments, contributions, and claims efficiently from self-service portals.
- Designed role-based dashboards using **Angular**, **D3.js**, and **REST APIs** to give actuaries, underwriters, and HR admins visibility into claims volume, premium trends, and member plan status.
- Developed modular **Spring Boot** services and configured business rules to automate premium billing workflows, reducing manual entry in group policy operations by **40%**.
- Collaborated with actuarial teams to align **Java**, **J2EE**, and **Jakarta EE**-based premium engines with risk-adjusted pricing logic, ensuring consistency across group plans, contribution tiers, and optional riders.
- Designed and secured **REST APIs** using **Spring Boot** and **Spring Security** integrating with actuarial systems and payroll platforms for synchronized premium deductions and member data validation.
- Implemented robust access control using **OAuth2**, **JWT**, and **SSO**, delivering a seamless login experience across employer, broker, and internal admin portals in alignment with **POPIA** compliance standards.
- Applied **Memcached** to cache read-heavy datasets like policy plans and eligibility rules, improving **Angular UI** responsiveness and backend throughput by approximately **15%**.
- Deployed **RabbitMQ** clusters on **Amazon EC2** with **Ansible**-driven configuration and secure **VPC** networking and auto-recovery policies to handle asynchronous messaging for premium calculations and claim lifecycle events, ensuring reliability during peak business hours.
- Utilized **PostgreSQL** for structured data such as group policies and premium history, and **MongoDB** for unstructured documents like claim forms and uploaded beneficiary records.
- Stored policy documents, claim evidence files, and exportable audit logs in **Amazon S3**, enabling cost-effective archival, versioning support, and easy retrieval for POPIA-aligned compliance audits.
- Established **CI/CD automation** with **Jenkins**, incorporating build pipelines for both **Java microservices** and **Angular apps**, supporting rollback and multi-environment deployments.
- Created **environment-specific configuration files** and managed deployment variables securely using **Jenkins** and **Git**, ensuring consistent behaviour across dev, QA, and production pipelines.
- Automated regression testing using **Karma**, **Jasmine**, and **Selenium**, ensuring consistent UI behaviour across complex, form-based workflows and dashboards.
- Developed backend unit tests using **JUnit** and **Mockito** for **Spring Boot (Java)** services, increasing confidence in calculation accuracy and maintaining backward compatibility in policy change cycles.
- Monitored application health using the **ELK Stack**, aggregating logs from **Spring Boot (Java)**, **RabbitMQ**, and **MongoDB**, supporting production debugging and issue traceability.
- Participated in **Agile** ceremonies including daily standups, sprint planning, and retrospectives, working closely with QA, product owners, and UI/UX teams to drive sprint outcomes.
- Documented **API contracts**, database schemas, and UI flows in **Confluence**, improving QA preparedness and enabling smoother onboarding for new developers.

Environment: Java, Spring Boot, J2EE, Jakarta EE, Angular, Bootstrap, SASS, LESS, D3.js, RESTful APIs, Spring Security, OAuth2, JWT, SSO, PostgreSQL, MongoDB, RabbitMQ, Memcached, Amazon EC2, Amazon S3, Ansible, Jenkins, Git, Karma, Jasmine, Selenium, JUnit, Mockito, ELK Stack, Agile, Confluence

Software Engineer | Snapdeal June 2015 – May 2016

Bengaluru, KA, India

Project Description:

The project focused on developing and maintaining Snapdeal's product catalogue management system, which handled onboarding, categorization, and searchability of products across the platform. Aimed at ensuring accurate product information, faster search results, and smooth browsing experience for millions of online shoppers. The platform also supported integration with seller portals to manage product listings, stock availability, and pricing updates in real-time.

Responsibilities:

- Built **responsive frontend interfaces** using **HTML5, CSS3, JavaScript, and jQuery** to render paginated product listings, dynamic filters, and quick-view modals for enhanced catalogue navigation on Snapdeal's e-commerce platform.
- Supported **Spring MVC backend development** by creating **REST APIs** to retrieve catalogue data from **MySQL** based on filters like category, price range, and availability.
- Queried **MySQL** to retrieve product metadata including title, price, brand, and stock count, enabling accurate filtering logic and powering UI components with real-time product states.
- Contributed to **backend logic** for product enrichment, metadata normalization, and content validation before pushing structured records to **Apache Solr** for fast and faceted search indexing.
- Wrote **Java batch jobs** using to periodically extract recently updated products from **MySQL** and index them into **Solr**, ensuring near real-time search availability without manual intervention.
- Configured **Log4j** for structured logging of API requests, exceptions, and service performance, enabling early debugging during staging/UAT cycles and supporting faster issue resolution.
- Wrote **JUnit test cases** to validate business logic for product uploads, pricing thresholds, and attribute enrichment, improving code quality and reducing defect rates during testing.
- Deployed **Java**, **J2EE**, and **Jakarta EE** WAR files to **Apache Tomcat** servers running on **Linux** staging environments using shell scripts that automated stop-cleanup-deploy-restart sequences.
- Monitored deployed builds by reviewing **HTTP access logs** and application logs to identify slow endpoints, API failures, or backend exceptions, collaborating with QA teams during UAT validation.
- Used **Git and Bitbucket** to manage code across feature branches, resolve merge conflicts during integration, and push clean commits tagged to corresponding **JIRA** user stories.
- Participated in **Agile sprint activities**, including stand-ups, sprint planning, backlog refinement, and task estimation, contributing to a structured, iterative delivery model guided by product leads and senior engineers.

Environment: Java, J2EE, Jakarta EE, Spring MVC, JSP, HTML5, CSS3, JavaScript, jQuery, RESTful APIs, MySQL, Apache Solr, Apache Tomcat, Log4j, Linux, JUnit, Git, Bitbucket, JIRA, Agile