**A blue hexagon with white text

AI-generated content may be incorrect.**

Anil Kumar. M

+1(417)-379-6320 / [anilkumarm2195@gmail.com](mailto:anilkumarm2195@gmail.com)

[LinkedIn](http://www.linkedin.com/in/anil-kumar-m-0439b7356) |Portfolio |[Medium](https://medium.com/@anilkumarm2195)

**Summary**

* Over 8 years of experience building dynamic and responsive user interfaces with **React** and **Angular**. Focused on creating efficient, reusable components and ensuring smooth user interactions across web applications by leveraging modern techniques like **React Hooks**, **React Router**, **Angular Material**, and **Lazy Loading** for optimized performance.
* Expertise in building robust and scalable **microservices** using **Spring Boot**, designing and implementing business logic that scales efficiently while ensuring high performance, maintainability, and fault tolerance. Experienced with **Spring Security** and **OAuth 2.0** to secure API endpoints.
* In-depth knowledge of designing and developing **microservices** with **Spring Boot** and implementing **RESTful APIs** to ensure fault isolation and horizontal scalability. Focused on decoupling services to provide a flexible and scalable system.
* Utilized **Kafka** for real-time, event-driven communication between microservices, enabling high-throughput, fault-tolerant, and scalable **data streaming**. Integrated Kafka for effective event processing and improved data synchronization across distributed systems.
* Implemented **RabbitMQ** for managing asynchronous communication between services, optimizing resource utilization and ensuring that the system remains responsive during peak loads by managing background tasks and messaging queues effectively.
* Extensive experience working with **SQL databases** such as **PostgreSQL**, **MySQL**, and **Oracle**. Optimized complex queries, ensuring high-performance data retrieval, transaction management, and ensuring consistency in relational data systems.
* Proficient in working with **NoSQL databases** like **MongoDB**, **Cassandra**, and **Couchbase**, designing flexible data models and ensuring efficient handling of large-scale, unstructured data for high-performance and scalable applications.
* Managed cloud infrastructure using **AWS** services such as **EC2**, **Lambda**, **S3**, **RDS**, **CloudWatch**, and **API Gateway**. Leveraged cloud-based services to enhance scalability, security, and cost-effectiveness of applications, optimizing cloud resources for better operational efficiency.
* Implemented and maintained **CI/CD pipelines** using **Jenkins** to automate build, test, and deployment processes. Ensured continuous integration and continuous delivery for faster release cycles, higher quality, and better project velocity.
* Proficient in using **GitHub** for version control, managing repositories, implementing branching strategies, and collaborating with teams on code review and versioning, ensuring the integrity of code and smooth deployment cycles.
* Expertise in writing **unit tests** using **JUnit** and **Mockito** to ensure the reliability and stability of backend services. Additionally, implemented **Selenium** for automated UI testing to verify the front-end behavior and ensure cross-browser compatibility.
* Set up logging and monitoring systems using **ELK Stack** (Elasticsearch, Logstash, Kibana) to track application performance and errors, proactively addressing issues, optimizing system performance, and ensuring smooth application health.
* Actively participated in **Agile Scrum** practices, working in **sprints**, attending daily stand-ups, and ensuring continuous delivery with **Test-Driven Development (TDD)**. Focused on improving code quality and meeting sprint goals through iterative development and feedback.
* Implemented **security best practices** including **JWT authentication**, **OAuth 2.0**, and **Spring Security** for secure and compliant application development. Ensured data encryption, secure access management, and overall application security in compliance with industry standards.
* Integrated **Kafka** for real-time **data streaming** and event processing, enabling seamless data flow across distributed microservices, which improved real-time analytics and decision-making.
* Implemented **Redis** for caching frequently accessed data, reducing database load, improving response times, and enhancing application performance, especially during high user traffic and peak periods.

**Tools and Technologies:**

* **Programming Languages**: Java (17/11/8), JavaScript, TypeScript
* **Technologies & Frameworks:** Spring Boot, Spring Security, Spring MVC, Hibernate, JPA, Angular, ReactJS, Redux, RxJS, Angular Material, Bootstrap, Node.js
* **Database**: PostgreSQL, MySQL, Oracle, MongoDB, Cassandra, Couchbase, Redis
* **Web Services:** HTML5, CSS3, JavaScript, Angular, ReactJS, SOAP, RESTful APIs
* **Messaging**: Kafka, RabbitMQ, ActiveMQ
* **CI/CD & DevOps:** Jenkins, GitLab, Docker, Kubernetes
* **Version Control Systems:** Git (GitHub), SVN
* **IDE Used**: IntelliJ IDEA, Visual Studio Code, Eclipse, NetBeans
* **Tools:** SonarQube, Postman, GitHub, Maven, Gradle, JIRA, Swagger
* **Monitoring Tools:** Dynatrace, AWS CloudWatch, ELK Stack (Elasticsearch, Logstash, Kibana), Prometheus, Grafana
* **Cloud Technologies**: AWS (EC2, S3, Lambda, RDS, CloudWatch), Microsoft Azure, Google Cloud Platform (GCP)
* **Operating Systems:** Windows, Linux (Ubuntu, CentOS)

**Professional Experience:**

**Client: Abbott Inc, Minneapolis, Minnesota**

**Role: Advanced Software Developer**

**Duration: January 2024 – Present**

**Project: Medi Stream: Real-Time Health Data Pipeline**

The Medical Data Streaming Services platform is a high-performance, cloud-based system designed to process and analyze real-time medical data from healthcare devices. It enables secure, scalable, and low-latency data streaming, ensuring that critical patient information is processed, stored, and made accessible to healthcare professionals and patients in real time.

**Roles & Responsibilities:**

* Developed dynamic and responsive user interfaces using **Angular** and **Material-UI**, delivering intuitive and interactive dashboards for real-time medical data visualization.
* Built reusable **Angular components** and leveraged **RxJS Observables** for handling asynchronous health data streams efficiently.
* Implemented **Angular Router** for seamless navigation across patient monitoring dashboards, analytics reports, and data-streaming insights.
* Optimized application performance with **Lazy Loading** and **Change Detection Strategies**, ensuring smooth rendering of high-frequency real-time data.
* Managed global state with **NgRx Store**, ensuring consistent data updates across different application modules while improving debugging and maintainability.
* Designed and developed **scalable and resilient microservices** using **Spring Boot**, handling real-time medical data processing and event-driven operations.
* Implemented **Spring Cloud Gateway** for API routing, load balancing, and securing endpoints with **OAuth 2.0 and JWT authentication**.
* Developed **RESTful APIs** for seamless interaction between frontend, data processing layers, and external healthcare providers.
* Optimized system performance by utilizing **Spring WebFlux (Reactive Programming)** for non-blocking event processing.
* Automated batch processing and periodic data aggregation using **Spring Batch**, improving data consistency and processing efficiency.
* Integrated **Apache Kafka for real-time streaming of medical sensor data**, ensuring low-latency processing and fault-tolerant data transmission.
* Configured **Kafka Producers, Consumers, and Streams API,** enabling efficient message distribution across microservices.
* Implemented **Kafka Connect** to ingest data from various sources, including IoT medical devices and third-party healthcare systems.
* Ensured reliability using **Kafka Schema Registry (Avro format)** for structured and backward-compatible message serialization.
* Utilized **PostgreSQL** for structured data storage, optimizing complex queries and ensuring ACID compliance for patient records.
* Implemented **MongoDB** for flexible and schema-less storage of real-time streaming logs, unstructured telemetry data, and analytics insights.
* Designed **optimized indexing strategies** for efficient querying of time-series medical data, reducing latency in fetching patient reports.
* Deployed microservices on **AWS EC2 with Auto Scaling and Elastic Load Balancer (ELB),** ensuring high availability and performance.
* Stored large-scale health data securely using **AWS S3**, enabling seamless integration with data analytics pipelines.
* Managed infrastructure provisioning using **AWS CloudFormation** and monitored system performance via AWS CloudWatch.
* Secured application access with **AWS IAM roles and policies**, ensuring compliance with healthcare data protection regulations.
* Enabled event-driven workflows with **AWS Lambda** for on-demand processing of patient alerts and notifications.
* Established **CI/CD pipelines in Jenkins**, automating builds, tests, and deployments to reduce time-to-market.
* Implemented **GitLab CI/CD workflows** for continuous integration, ensuring a streamlined release process across multiple environments.
* Managed source code repositories using **GitHub**, ensuring proper version control, branching strategies, and structured pull request reviews.
* Conducted **code reviews and merge requests**, maintaining high code quality and collaboration across teams.
* Developed unit and integration tests using **JUnit and Mockito**, ensuring backend services met functional requirements with minimal regression issues.
* Implemented logging and monitoring using **ELK Stack (Elasticsearch, Logstash, Kibana)**, enabling real-time tracking of system health and anomalies.
* Followed **Agile Scrum methodology**, actively participating in daily stand-ups, sprint planning, retrospectives, and backlog grooming sessions.
* Embraced **Test-Driven Development (TDD)**, writing tests before implementation to ensure feature reliability from the ground up.
* Worked closely with cross-functional teams, including data scientists, security experts, and DevOps engineers, to ensure seamless integration and system scalability.

**Environment:** Java 17, Spring Core 6, Spring Boot 3, Hibernate 6, RESTful APIs, GraphQL, Spring Batch 5, AOP, AWS (EC2, S3, RDS, Lambda, Auto Scaling, CloudWatch, CloudTrail, CloudFormation, CodeDeploy, CodeBuild), Angular 16, TypeScript 5, HTML5, CSS3, Bootstrap 5, Apache Kafka 3, RabbitMQ 3.12, PostgreSQL 15, MongoDB 7, Apache Tomcat 10, Jenkins 2.440, Swagger 3 (OpenAPI), Spring Security 6, Ansible 2.15, XSLT, JavaScript (ESNext), Agile, Scrum, CI/CD (Jenkins, GitHub Actions, GitLab CI/CD).

**Client: State of Missouri, Springfield, Missouri**

**Role: Senior Java Developer**

**Duration: March 2021 - December 2023**

**Project: Missouri Corrections Integrated System**

​The Missouri Corrections Integrated System (MOCIS) is an initiative by the Missouri Department of Corrections (MODOC) aimed at modernizing and integrating the state's correctional facilities' operations. MOCIS focuses on integrating various modules and services within the corrections system to streamline processes and enhance service delivery. The initiative involves collaboration among MODOC, the Office of Administration, and external vendors to implement comprehensive solutions.

**Roles & Responsibilities:**

* Developed dynamic, responsive, and user-friendly interfaces using **Angular 12/13**, leveraging **Angular CLI** and **Angular Services** to enhance the experience for correctional officers, administrators, and other stakeholders interacting with the corrections system.
* **Utilized Angular Reactive Forms and Validations** in conjunction with **ngRx** (state management) to streamline state handling and make the application more maintainable. Used **Angular Material** and **Bootstrap 5** for a consistent design system that ensures a uniform user interface (UI) across all correctional system modules.
* **Implemented lazy loading** for efficient resource management, ensuring smooth loading times for large datasets such as inmate records, parole histories, and institutional details.
* **Developed and integrated RESTful APIs** using **Spring Boot**, adhering to industry standards for **REST API design** and **documentation using Swagger** for easy integration and testing.
* **Backend development powered by Spring Boot** microservices architecture. Each microservice handled distinct business logic, such as **inmate management**, **parole hearings**, and **case progression**, ensuring independent deployments, fault isolation, and improved system resilience.
* **Implemented robust security protocols** using **OAuth 2.0** and **JWT authentication** to protect sensitive information within the system. Introduced **role-based access control** to ensure correctional officers, administrators, and external stakeholders only had access to authorized data.
* **Deployed backend microservices on AWS EC2** instances, ensuring high availability and reliability. Applied **AWS Auto Scaling** to manage traffic spikes, providing optimal resource allocation and minimizing costs.
* **Data storage managed with AWS RDS (PostgreSQL)**, ensuring **transactional consistency** and **data integrity** for structured records like inmate profiles, legal histories, and rehabilitation reports. **AWS S3** used to securely store large files, such as images, legal documents, and videos.
* **Integrated AWS CloudWatch** for performance monitoring and **real-time alerts** for system anomalies. Monitored key metrics such as response times, error rates, and system throughput.
* **AWS Lambda** used to implement **serverless functions** for sending notifications related to parole status updates and case file reviews, optimizing system efficiency.
* **Integrated Apache Kafka** for handling **real-time event-driven processing** for critical data updates. Kafka Streams were used to process updates to inmate records or changes in parole status.
* **MongoDB** utilized for schema-less storage of unstructured data, such as logs and alerts, providing flexibility and scalability for corrections data storage.
* **Collaborated with cross-functional teams** (product managers, UX designers, data scientists, and QA engineers) using **Agile Scrum methodology**. Actively participated in **daily stand-ups**, **sprint planning**, and **retrospectives** to ensure timely delivery of milestones and project objectives.
* Managed tasks using **JIRA**, tracked progress, and resolved issues in bi-weekly sprints, ensuring high project visibility and stakeholder engagement.
* **Implemented unit and integration tests** using **JUnit** and **Mockito**, ensuring backend services were reliable and met functional requirements. Tests were part of the CI/CD pipeline, ensuring that the codebase remained stable.
* **Adopted Test-Driven Development (TDD)** principles, ensuring that new features were tested and validated at each stage before implementation.
* **Automated build, test, and deployment workflows** using **Jenkins**, enabling **continuous integration (CI)** and **continuous delivery (CD)** for faster release cycles and efficient delivery of new features.
* **Collaborated with external agencies** through **API integration** to ensure interoperability with other government systems, enhancing data exchange and improving operational efficiency.
* **Implemented detailed logging** using **AWS CloudWatch** and **Log4j**, tracking system activities, error logs, and operational events for easier debugging and performance monitoring.
* **Managed version control** with **GitHub**, ensuring organized source code management. Used **feature branches** for new feature development and **peer code reviews** to maintain high-quality code standards.
* Provided **ongoing system maintenance and support**, ensuring the platform remained stable, secure, and responsive to the evolving needs of the Missouri Department of Corrections.

**Environment:** Core Java, Spring Framework, Spring Boot, RESTful APIs, Microservices, Dependency Injection (DI), Inversion of Control (IoC), AOP, Angular 12/13, TypeScript, HTML5, CSS3, Angular CLI, Angular Material, Bootstrap 5, Java 11, Spring Boot 2.5/2.6, Spring Security, Spring Data JPA, Spring Cloud, JWT Authentication, OAuth 2.0, Apache Kafka, AWS EC2, AWS Lambda, AWS RDS (PostgreSQL), AWS S3, AWS Auto Scaling, AWS CloudWatch, Jenkins, Terraform, Gradle, Docker, Kubernetes, Git, GitHub, JUnit, Mockito, JEST, TestNG, TDD, ELK Stack (Elasticsearch, Logstash, Kibana), Apache Kafka, Amazon SNS/SQS, OAuth 2.0, JWT Authentication, JIRA, Scrum.

**Client: Cipher Health, NYC NY**

**Role: Java Full Stack Developer**

**Duration: October 2018 – February 2021**

**Project : Patient Health**

**PatientConnect** is a comprehensive patient engagement platform designed to streamline communication between healthcare providers and patients. The application facilitates appointment scheduling, real-time health monitoring, secure messaging, and personalized health reminders, aiming to enhance patient satisfaction and improve health outcomes.

**Roles & Responsibilities:**

* Developed dynamic and responsive user interfaces using **Angular 12**, **TypeScript**, **HTML5**, and **CSS3** to deliver an intuitive, user-friendly experience for patients, healthcare providers, and administrators. Ensured mobile responsiveness for accessibility across all device types.
* Implemented interactive forms and workflows for **appointment scheduling**, **medical history updates**, and **real-time health monitoring** features, streamlining data collection and improving patient experience.
* Enhanced user experience by utilizing **Reactive Forms** and **Angular Services** for smooth validation, error handling, and real-time dynamic field management.
* Integrated **AJAX** calls using **Angular HTTPClient** to asynchronously fetch, update, and display data, ensuring a seamless experience during data-intensive operations.
* Developed backend services using **Java 11** and **Spring Boot 2.5** to create high-performance **RESTful APIs** for managing patient data, appointments, and health metrics.
* Utilized **Spring Data JPA** for data persistence, optimizing **CRUD** operations on relational databases such as **PostgreSQL**, ensuring efficient data retrieval.
* Integrated **Spring Security** with **JWT Authentication** and **OAuth 2.0** for secure, role-based access control to sensitive healthcare data.
* Designed and implemented **Spring AOP** to handle cross-cutting concerns such as logging, security checks, and transaction management.
* Implemented **WebSocket** for real-time communication, enabling live chat, appointment reminders, and emergency notifications between patients and healthcare providers.
* Managed patient data storage using **AWS RDS** with **PostgreSQL**, ensuring high availability and fault tolerance with automatic backups and replication.
* Integrated **AWS S3** for scalable and secure file storage solutions, managing patient documents, medical images, and lab reports.
* Optimized SQL queries and database performance by implementing indexing, stored procedures, and a normalized schema for patient records.
* Deployed microservices on **AWS EC2** instances with **Elastic Load Balancer (ELB)** to ensure scalability and availability during peak demand.
* Utilized **AWS Auto Scaling** to dynamically adjust the number of running instances based on system load for cost-effective resource management.
* Monitored performance with **AWS CloudWatch**, setting up custom metrics and alarms to detect performance issues and mitigate them promptly.
* Configured **CI/CD pipelines** using **Jenkins** for automating build, testing, and deployment processes, leading to faster release cycles.
* Automated deployment using **Docker** containers and **Kubernetes** for container orchestration, ensuring consistency across environments and efficient management of microservices.
* Set up deployment workflows with **GitHub Actions**, streamlining code integration and reducing manual intervention.
* Participated actively in **Agile Scrum** ceremonies, using **JIRA** for task and bug management to ensure project tracking and transparency.
* Followed **Test-Driven Development (TDD)** principles, writing tests before feature implementation to ensure code reliability and minimal defects.
* Used **JUnit** and **Mockito** for backend unit testing, ensuring backend services were thoroughly tested, and **JEST** for frontend testing of Angular components.
* Integrated **OAuth 2.0** and **JWT** for secure, token-based user authentication, protecting user data.
* Worked with compliance teams to ensure adherence to **HIPAA** regulations, including implementing encryption and secure data storage practices.
* Utilized **AWS IAM (Identity and Access Management)** to manage secure access to AWS resources and services.
* Leveraged the **ELK Stack** (Elasticsearch, Logstash, Kibana) for real-time logging and performance monitoring, enabling proactive issue tracking.
* Set up **Prometheus** and **Grafana** for system monitoring and visualization of performance metrics such as CPU, memory, and network utilization.
* Integrated **Sentry** for real-time error tracking, enabling quick resolution of issues that affected application performance or user experience.
* Collaborated with **product managers** and stakeholders to refine feature requirements and ensure the platform met the needs of healthcare providers and patients.
* Conducted **code reviews** and **pair programming** sessions, promoting knowledge sharing, ensuring high code quality, and fostering best practices.
* Communicated technical challenges and updates to non-technical stakeholders, ensuring alignment with business goals and timely resolution of blockers.

**Environment:** Angular 12, TypeScript, HTML5, CSS3, Java 11, Spring Boot 2.5, RESTful APIs, Spring Data JPA, PostgreSQL, AWS RDS, AWS S3, AWS EC2, AWS Lambda, AWS CloudWatch, AWS Auto Scaling, Jenkins, Docker, Kubernetes, GitHub Actions, JUnit, Mockito, JEST, ELK Stack, Prometheus, Grafana, OAuth 2.0, JWT, HIPAA compliance, Scrum, JIRA.

**Client: Credit Suisse, NYC, NY**

**Role: Software Engineer**

**Duration: August 2016 – September 2018**

**Project: Client Trading Technology Platform**

The Client Trading Technology Platform is a high-performance, real-time trading platform designed to empower institutional traders and clients to execute and manage trades seamlessly across a wide range of financial instruments. With an emphasis on scalability, security, and speed, this platform integrates real-time market data, trade execution, and client portfolio management, delivering a robust solution for complex financial environments. The platform leverages Angular for the frontend and Java-based backend services to ensure a responsive, intuitive, and highly secure user experience.

**Roles & Responsibilities:**

* Developed dynamic, responsive user interfaces for the trading platform using **Angular**, ensuring a smooth and user-friendly experience for traders and clients across devices.
* Utilized **Angular services** and **RxJS** to manage real-time data streams, enabling asynchronous data handling for market feeds, trade execution statuses, and client portfolio updates, ensuring seamless user experience.
* Designed and implemented **RESTful APIs** using **Java 17** and **Spring Boot**, ensuring secure, efficient communication between the frontend and backend services for real-time trade processing and market data analysis.
* Developed **microservices** architecture using **Spring Boot** to ensure a scalable, modular approach to platform components, including trade execution, risk management, and order processing services.
* Integrated **Apache Kafka** for event-driven data streaming, enabling real-time updates on trade status, market alerts, and execution logs across distributed services.
* Employed **Spring Security** and **OAuth 2.0** with **JWT** to implement secure authentication and authorization for users, ensuring the integrity and confidentiality of financial transactions and sensitive client data.
* Worked with **PostgreSQL** to handle the storage and querying of trade data, market information, and client portfolios, optimizing database interactions to ensure performance and scalability.
* Utilized **Spring Data JPA** for seamless integration between Java objects and relational database entities, improving data management and reducing boilerplate code for CRUD operations.
* Deployed microservices on **AWS EC2** instances, using **Elastic Load Balancer (ELB)** to ensure high availability, and implemented **Auto Scaling** to automatically adjust resources based on platform demand during peak trading periods.
* Leveraged **AWS S3** for secure and scalable storage of trade logs, historical data, and market reports, ensuring easy access and compliance with data retention policies.
* Integrated **Redis** for caching frequently queried data, improving the platform's performance by reducing latency during high-frequency trading activities.
* Automated build, test, and deployment workflows using **Jenkins** and **GitLab**, streamlining the CI/CD pipeline and ensuring consistent, fast, and reliable deployment cycles for new features and updates.
* Conducted unit and integration testing with **JUnit** and **Mockito**, ensuring the reliability and accuracy of backend services, particularly trade execution and market data processing components.
* Collaborated with **cross-functional teams**, including business analysts, product owners, and QA engineers, to define feature requirements, refine technical specifications, and ensure alignment with business and compliance needs.
* Participated in **Agile Scrum** processes, including sprint planning, daily stand-ups, and retrospectives, contributing to project tracking and visibility using **JIRA**.
* Adhered to **Test-Driven Development (TDD)** principles, writing comprehensive tests before implementation to ensure code quality and reduce defects in the final product.

**Environment:** Angular, Java 11, Spring Boot 2.5, Microservices, Apache Kafka, PostgreSQL, AWS EC2, S3, RDS, Lambda, Jenkins, GitHub, OAuth 2.0, JWT, Docker, Kubernetes, Elasticsearch, Kibana, Prometheus, Grafana, JUnit, Mockito, Agile, Scrum, CI/CD.

**Client: WalkingTree Technologies, Hyderabad IN**

**Role: Associate Java Software Engineer**

**Duration: July 2014 – June 2015**

**Project: Automated Workflow Management System**

The Automated Workflow Management System is an internal solution developed to automate and streamline various business processes, ensuring more efficient task management, seamless collaboration, and faster decision-making. The system offers tools for automating approvals, task assignments, notifications, and status tracking, which significantly reduces manual work and optimizes workflow processes.

**Roles & Responsibilities:**

* **Provided ongoing support** for the **Automated Workflow Management System**, ensuring smooth operational flow and assisting internal teams with any technical issues.
* **Backend services** were built using **Java 11** and **Spring Boot**, and I contributed to resolving backend issues, ensuring system reliability.
* **Assisted in managing PostgreSQL**, ensuring smooth database operations, optimizing queries, and maintaining data integrity within the system.
* **Integrated and supported Kafka for real-time data streaming**, enabling efficient messaging and data updates across the system’s components.
* **Utilized AWS services** such as **EC2**, **RDS**, and **S3** to manage cloud infrastructure, ensuring high availability and scalability.
* **Monitored application performance** using **AWS CloudWatch** and implemented necessary optimizations to ensure system reliability.
* **Assisted in maintaining CI/CD pipelines** with **Jenkins** to automate deployments, reducing the risk of errors and enhancing deployment efficiency.
* **Collaborated on unit testing** using **JUnit**, ensuring that backend services met functional requirements and had minimal regression errors.
* **Worked in an Agile environment**, participating in Scrum ceremonies like sprint planning, daily stand-ups, and retrospectives for effective project tracking and collaboration.
* **Implemented logging and monitoring solutions** using tools like **ELK Stack**, ensuring better visibility into system behavior and proactive troubleshooting.

**Environment:** Java, Struts, JSP, Servlets, JDBC, Hibernate, Eclipse, GIT, Ant, log4j, SQL, XML, JSTL, Eclipse IDE, Struts Tag Libraries, JavaScript, SOAP, REST.

**Education:**

Master of Science in Computer Science, Missouri State University, Springfield, Missouri

Bachelor of Science in Computer Science, Jawaharlal Nehru Technological University, Hyderabad, India.