Kavya Jasti
Software Developer
kavyajasti0719@gmail.com | +1 (206)-278-7832 | LinkedIn

PROFESSIONAL SUMMARY

Software developer with 7 years of experience in building scalable web applications using Java/J2EE, Spring Boot, and modern JavaScript frameworks like React and Angular. Skilled in designing RESTful APIs, integrating cloud services (AWS), and working with databases such as MySQL, PostgreSQL, and MongoDB. Proficient in real-time data processing with Apache Kafka, implementing security with OAuth/JWT, and managing CI/CD pipelines using Jenkins and Docker. Passionate about delivering efficient, user-focused solutions and ensuring high-quality performance throughout the development lifecycle.

TECHNICAL SKILLS

Programming Languages	C, C++, C#, Java 8/11, SQL, JavaScript, Python
Java/J2EE Technologies	Servlets, JSP, JSTL, JDBC, JPA, EJB, JMS, JNDI, JavaBeans
Frameworks	Spring Boot, Spring MVC, Spring Data, Spring Batch, Spring IOC, Hibernate, Struts
Databases	Oracle, SQL Server, MySQL, MongoDB, Apache Cassandra
Web Technologies	HTML5, CSS3, Bootstrap, JavaScript, JSON, XML, RESTful APIs, SOAP
JavaScript Frameworks	Angular (12-16), React.js, Node.js, Vue.js, Next.js
Cloud Technologies	AWS (EC2, S3, RDS, VPC, IAM, ELB, Route53, SQS, SNS, ES, ECS, EKS), Docker, Azure, Google Cloud Platform (GCP)
Messaging Services	Apache Kafka, RabbitMQ
Architectural Patterns	Microservices, MVC, SOA, ORM
Scripting Languages	Shell Script, Bash
Application/Web Servers	Apache Tomcat, WebSphere, JBoss, WebLogic
IDEs	IntelliJ IDEA, Eclipse, NetBeans
Version Control	Git, GitHub
Build & CI/CD Tools	Maven, Jenkins
Testing Frameworks	JUnit, Selenium, Cucumber, Mockito, JMockit
Configuration Management	SVN, GitHub, Git, Jenkins
Additional Tools & Technologies	GraphQL, Kubernetes, Grafana, ELK Stack, Terraform, Postman, Log4J

PROFESSIONAL EXPERIENCE

Infinite Computer Solutions **August 2023-Present**
Full Stack Developer

Project Description: At Infinite Computer Solutions, I contributed to the development of a comprehensive system aimed at optimizing Electronic Medical Records (EMR) and Electronic Health Records (EHR) for

healthcare providers. This project aimed to streamline patient data management, improve clinical workflows, and enhance data accessibility and security. The goal was to enhance overall healthcare delivery by improving the efficiency of EMR/EHR systems, enabling providers to deliver better patient care while adhering to compliance standards.

Frontend Development:

- Developed a responsive, modern user interface using **Vue.js** and **Bootstrap**, optimizing the system for healthcare providers to efficiently access patient data and clinical records across different devices.
- Implemented **Vuex** for managing the application's state, ensuring consistent data flow between components for smooth handling of patient records and appointment details.
- Integrated **WebSocket's** to push real-time data updates on patient records, test results, and appointment statuses, providing healthcare professionals with live updates.
- Ensured the system is mobile-friendly with **Tailwind CSS**, allowing doctors and nurses to access the EMR/EHR system on smartphones and tablets, even in non-clinical settings.
- Used **Axioms** to connect the frontend with backend systems, enabling seamless API communication for fetching patient data, lab results, and prescriptions.
- Utilized **lazy loading** and **dynamic imports** to optimize load times, enhancing the user experience and performance of the application when managing large healthcare datasets.

Backend Development:

- Developed the backend using **Node.js** with the **Express** framework, ensuring high performance and scalability for healthcare data processing and real-time updates.
- Used **MongoDB**, a NoSQL database, to store unstructured patient data, enabling flexible data retrieval and fast access to patient health records and medical histories.
- Implemented **JWT** (JSON Web Tokens) for secure authentication, allowing role-based access control and ensuring only authorized healthcare professionals can access sensitive patient data.
- Implemented **Apache Kafka** to enable event-driven processing of patient data changes, such as appointment updates, lab results, and prescription refills, ensuring real-time synchronization across the healthcare platform.
- Leveraged **RabbitMQ** for messaging and queuing to handle patient record updates, appointment scheduling, and notifications across different services, ensuring reliable data delivery.
- Developed unit and integration tests using **Mocha** and **Chai** to ensure the backend services are robust and error-free in processing healthcare data and managing patient interactions.
- Architected backend services as independent microservices, packaged in **Docker** containers, enabling modular development and easy deployment for the EMR/EHR platform.

Cloud and Data Solutions:

- Deployed the application on **Google Cloud Platform (GCP)**, using **Compute Engine** for backend services and **Cloud SQL** for managed database storage, ensuring scalable and reliable infrastructure.
- Stored encrypted healthcare records in **Google Cloud Storage**, adhering to data privacy regulations like **HIPAA**, and providing highly available, durable storage for patient information.
- Integrated **Google Pub/Sub** for real-time messaging and event-driven data processing, enabling the system to handle patient data changes, such as prescription updates and test results, instantly.
- Configured **AWS Backup** for disaster recovery, ensuring that patient data was regularly backed up and recoverable in case of data loss or system failure, minimizing healthcare service disruptions.
- Used **Google Kubernetes Engine (GKE)** to orchestrate microservices, ensuring automated scaling and high availability of the backend services in response to increased demand.
- Leveraged **Google Big Query** to enable fast, SQL-like queries over large datasets, providing healthcare providers with quick access to patient analytics, including trends in health conditions and treatment outcomes.

- Implemented **Google Stack driver** for centralized monitoring of the EMR/EHR system's health, including performance metrics, logs, and error tracking to ensure smooth operations.

Environment: Google Cloud Platform (GCP), Compute Engine, Cloud SQL, Google Pub/Sub, BigQuery, Kubernetes Engine, Docker, Node.js, Express, MongoDB, RabbitMQ, JWT, Mocha, Chai, Vue.js, Bootstrap, Tailwind CSS, Axios, HIPAA Compliance.

Morgan Stanley **August 2021-July 2023**
Full Stack Developer

Project Description: I have Contributed to the development of advanced **wealth management tools** for financial products and planning services. Delivered **scalable**, **secure**, and **user-friendly** applications to optimize **investment strategies**, streamline **portfolio management**, and enhance **client engagement**. Designed robust **microservices architectures**, integrated **cloud solutions**, and built **responsive user interfaces** for seamless user experiences. Ensured data accuracy, regulatory compliance, and system reliability while collaborating with cross-functional teams to align technical solutions with **business objectives**.

Frontend Development (React):

- Developed dynamic and **responsive user interfaces** using **React**, ensuring seamless interactions across web and mobile platforms.
- Integrated **Redux** to handle complex state management, enabling consistent and predictable data flow across various components in large-scale applications.
- Utilized **Tailwind CSS** for utility-first styling and **Material-UI** for building consistent, accessible, and responsive components, ensuring a polished, high-quality user experience in wealth management tools.
- Integrated **REST APIs** to fetch and display dynamic financial data, such as market trends, transaction history, and investment performance, seamlessly linking frontend interfaces with backend systems.
- Performed thorough UI testing using **Jest** and **React Testing Library**, validating the correctness and functionality of interactive components and forms, ensuring a reliable user experience across platforms.
- Applied **Reacts lazy loading** and **code-splitting** techniques to reduce initial load times, improve page speed, and enhance overall performance for large-scale, data-heavy financial applications.
- Implemented **OAuth 2.0** and **JWT authentication** tokens in the frontend to maintain secure communication with backend services, ensuring that sensitive financial data remains protected during client interactions.

Backend Development:

- Built a scalable and maintainable **microservices-based architecture** using **Spring Boot**, ensuring each service could be developed, deployed, and scaled independently to handle the high complexity of financial data processing.
- Developed secure **REST APIs** to manage the lifecycle of financial products, transactions, and client portfolio data, ensuring a seamless exchange of data between various systems within the wealth management platform.
- Utilized **Hibernate** and **JPA** for **Object-Relational Mapping (ORM)**, optimizing database queries for performance and scalability while reducing the complexity of database interactions, critical for handling large amounts of transactional data.
- Implemented **Apache Kafka** for event-driven architecture, allowing real-time processing of transactional data and improving the scalability and reliability of microservices in the wealth management platform.
- Created comprehensive test suites using **JUnit** and **Mockito** for unit and integration testing of backend services, ensuring that APIs and database interactions functioned as expected without errors.
- Utilized **Kafka** for **event-driven architecture**, enabling real-time data streaming and processing.
- Conducted **unit testing (JUnit)** and **integration testing (Mockito)** for backend service reliability.

Cloud and Data Solutions:

- Deployed backend services on **AWS EC2**, optimized storage using **S3**, and utilized **RDS** for relational database management.
- Integrated **Elasticsearch** for fast financial data searches and predictive analysis capabilities.
- Managed application deployment and orchestration using **Kubernetes** and **Docker** for microservices scalability and fault tolerance.
- Configured **IAM roles and access policies** to ensure secure interactions with AWS resources.
- Integrated **Spring Cloud Sleuth** and **Zipkin** for distributed tracing to diagnose performance bottlenecks.
- Implemented **event-driven architecture** using **Apache Kafka** for real-time financial data streaming.

Environment: Java EE, Spring Boot, React, Angular, Hibernate, JPA, RESTful APIs, Tailwind CSS, AWS (EC2, S3, RDS), Kubernetes, JUnit, Mockito, Docker, OAuth 2.0, JWT, Agile Methodology.

Quest Diagnostics **December 2019-July 2021**
Software Developer

Project Description: I contributed to building a Diagnostic Test Management and Analytics Platform to optimize test lifecycle management, streamline appointment scheduling, and provide actionable insights for healthcare providers. The platform focused on enhancing operational efficiency, ensuring data accuracy, and enabling data-driven decision-making.

Frontend Development (Angular):

- Developed dynamic dashboards using Angular to present real-time analytics on diagnostic test performance, resource allocation, and appointment trends.
- Designed a **modular component-based architecture** with reusable Angular components to ensure maintainability and scalability of the application.
- Implemented **NgRx** for state management, ensuring consistent handling of application state across complex workflows and multi-user interactions.
- Enabled **real-time data synchronization** across the platform using WebSocket's, keeping test results, scheduling, and resource statuses updated in real time.
- Built **dynamic forms with reactive form handling**, ensuring precise validation for appointment booking, patient records, and data entry.
- Integrated **role-based access control** in the frontend to provide user-specific functionality, ensuring secure access to sensitive data.

Backend Development

- Designed a scalable microservices architecture using Spring Boot to handle test scheduling, inventory management, and analytics.
- Utilized **PostgreSQL** for structured data storage and implemented advanced indexing techniques for faster query processing.
- Configured an **API Gateway** to manage incoming requests efficiently, ensuring secure communication between frontend and backend services.
- Integrated **RabbitMQ** for asynchronous communication, enabling smooth task execution such as appointment reminders and report generation.
- Conducted rigorous backend testing with **TestNG** and **Mockito**, ensuring API reliability and data integrity.

Cloud and Data Solutions:

- Deployed the application backend on Azure App Services, ensuring availability and fault tolerance.
- Managed report documents and large datasets using **Azure Blob Storage**, enabling secure access and scalability.
- Utilized **Azure Key Vault** to securely store application credentials, certificates, and API keys.

- Managed application containerization and orchestration using **AKS**, allowing seamless scaling of microservices.
- Integrated **Azure Monitor** to track backend performance, resource utilization, and error logging.
- Implemented automated **data backup and disaster recovery** strategies to ensure uninterrupted operations.

Environment: Angular, Java, Spring Boot, PostgreSQL, RESTful APIs, RabbitMQ, WebSockets, Azure (App Services, Blob Storage, Key Vault, AKS, Monitor), TestNG, Mockito, OAuth 2.0, RBAC, Agile Methodology.

IPrism Technologies **October 2017-November 2019**
Software Developer

Project Description: I contributed to the development of an eCommerce platform aimed at providing a seamless shopping experience for customers, optimizing product management, and enhancing order processing. The platform focused on improving user engagement, boosting sales, and ensuring a secure and responsive shopping environment.

Frontend Development

- Developed the user interface of the eCommerce application using **HTML, CSS, and JavaScript**, ensuring a responsive and mobile-friendly design.
- Integrated third-party libraries for enhancing **UI/UX**, such as carousel sliders for featured products and modals for product details.
- Worked on form validation and input handling for user registration, login, and checkout processes, ensuring proper error handling and data integrity.
- Implemented **AJAX**-based asynchronous requests to fetch product data and user information without page reloads, providing a smoother browsing experience.
- Conducted cross-browser testing to ensure compatibility and accessibility across various devices and browsers.

Backend Development

- Created simple CRUD (Create, Read, Update, Delete) operations for managing product catalogs, customer profiles, and order details.
- Integrated basic logging using Java libraries (e.g., Log4j) to track application errors and system activities, improving issue troubleshooting.
- Ensured data consistency by implementing basic transaction management for order processing and payment confirmation.
- Assisted in setting up database relationships, such as one-to-many (product-category) and many-to-many (user-orders), to properly structure data.
- Created basic admin panels to manage products, view customer orders, and process returns or refunds.
- Assisted in setting up crone jobs for regular database backups and data cleanup processes.

Cloud and Data Solutions

- Deployed the eCommerce application on AWS EC2 instances with basic auto-scaling configurations, ensuring the platform could handle fluctuating traffic during peak shopping periods.
- Managed MySQL databases on AWS RDS, ensuring high availability and simplified maintenance of customer, product, and order data.

- Configured AWS CloudFront as a Content Delivery Network (CDN) to distribute static content (images, CSS, JavaScript) efficiently and reduce page load times.
- Implemented SSL certificates on the AWS load balancer to encrypt data during transmission, ensuring secure communication between the client and server.
- Set up automated daily backups of MySQL databases using AWS RDS snapshot functionality to ensure data persistence and minimize data loss risks.
- Used AWS S3 for storing product images and user-generated content, enabling efficient retrieval and reducing server load.
- Integrated AWS CloudWatch to monitor EC2 instances' health and application logs for real-time performance tracking, ensuring application stability.
- Set up AWS Cost Explorer to monitor cloud costs and optimize resources based on the application's usage patterns, ensuring cost-efficiency during development and production.

Environment: HTML, CSS, JavaScript, jQuery, Java, Spring Framework, MySQL, RESTful APIs, AWS (EC2, RDS, S3, CloudFront, CloudWatch, Elastic Beanstalk), JWT, Log4j, Agile Methodology.

EDUCATION

Master's in computer science – University of Missouri Kansas City, Kansas City, MO
Bachelor of Technology in Computer Science and Engineering – Dhanekula Institute of Engineering and Technology, Vijayawada, India