This project involved the development of a full-stack enterprise-level web application using Java, Spring Boot, React, and microservices architecture. The system was designed to handle complex business workflows, data management, and API integrations across multiple platforms. Key functionalities included secure user authentication, scalable microservices, real-time event processing with Kafka, and cloud deployment using AWS and Azure. The project ensured high performance, scalability, and security through modern DevOps, CI/CD pipelines, and containerization with Docker and Kubernetes.

Key Features & Responsibilities:

1. Full-Stack Application Development
    - Developed a Single Page Application (SPA) using React 16 with Redux for state management.
    - Designed and implemented backend microservices using Spring Boot and Flask.
    - Developed RESTful APIs and GraphQL APIs for seamless data exchange.
    - Designed custom administrator dashboards with React.js and GraphQL for profile management and data analytics.
2. Microservices & API Development
    - Developed RESTful Microservices using Spring Boot and Flask, deployed on AWS using EBS and EC2.
    - Implemented Kafka producers and consumers in Java for event-driven architecture.
    - Developed GraphQL API server using Express-GraphQL library for efficient data fetching.
    - Used OAuth2, AWS Cognito, and JWT authentication for secure API access.
3. Cloud & DevOps Integration
    - Deployed Spring Boot microservices using Docker & Kubernetes on AWS & Azure.
    - Implemented CI/CD pipelines using Jenkins, Git, and Maven for automated deployments.
    - Built Kubernetes environments via Terraform, ensuring scalability and infrastructure automation.
    - Configured CloudWatch, CloudFormation, S3, RDS, SNS, and SQS for monitoring and messaging.
4. Database & Data Processing
    - Designed PostgreSQL, MySQL, and MongoDB databases for transactional and NoSQL data storage.
    - Used Cassandra database for user grid management and custom user functionality.
    - Developed PL/SQL stored procedures and triggers for automated data processing.
    - Implemented database connectivity using JDBC with Oracle 9i for legacy integrations.
5. Security & Performance Optimization
    - Integrated Azure security mechanisms for web applications.
    - Implemented Kafka security best practices such as encryption, authentication, and authorization.
    - Conducted incident investigations, vulnerability assessments, and malware analysis.
    - Developed GraphQL introspection queries to visualize schema relationships and optimize performance.
6. UI/Frontend Development
    - Developed dynamic web pages using React.js, AngularJS, Node.js, Bootstrap, and Material.io.
    - Migrated legacy UI components to React styled components.
    - Created an adaptive web design using Media Queries, HTML5, CSS3, and AJAX.
    - Developed unit tests for React components using Mocha, Gulp, Chai, Protractor, Karma, and Node.js.
7. Messaging & Event Processing
    - Developed Kafka producers and consumers to handle real-time streaming data.
    - Implemented RabbitMQ for message queuing and asynchronous processing.
    - Utilized AWS SNS & SQS for messaging services.
8. API Monitoring & Logging
    - Monitored REST API performance using Postman, CloudWatch, and ELK stack.
    - Ensured API backward compatibility by managing API versioning with Swagger/OpenAPI.
9. Testing & Deployment
    - Developed unit tests using JUnit, Mockito, and Rational Clear Case for version control.
    - Automated application processes using Quartz scheduler for batch operations.
    - Implemented SOAP-based XML web services for external system integrations.

How It Works:

1. User Authentication & Dashboard Access:
    - Users register and log in via OAuth2, AWS Cognito, or JWT authentication.
    - Role-based access ensures secure data retrieval.
2. Backend Microservices Execution:
    - Spring Boot microservices handle different business functionalities (transactions, user management, notifications).
    - API endpoints are exposed via REST and GraphQL.
3. Real-Time Data Processing:
    - Kafka consumers and producers handle live data streams.
    - Transactions are logged and monitored for security & performance optimization.
4. Database Operations & Management:
    - PostgreSQL & MySQL store structured transactional data.
    - MongoDB & Cassandra are used for NoSQL-based user & session storage.
5. Deployment & DevOps Management:
    - CI/CD pipelines (Jenkins, Git, Maven) ensure automated deployments.
    - Docker & Kubernetes provide containerized scalability.
    - AWS services like EC2, S3, CloudWatch ensure smooth monitoring.

Technologies & Tools Used:

Backend Development:

- Java, J2EE, Spring Boot, Spring MVC, Spring Integration
- Hibernate ORM, DAO, TDD, JUnit, Mockito, Log4j
- Microservices with Flask and Django
- RESTful APIs, GraphQL, SOAP Web Services

Frontend Development:

- React.js, AngularJS, Redux, Material.io, Bootstrap
- HTML5, CSS3, JavaScript, TypeScript, AJAX, JSON
- JSP with JavaBeans, JSTL, and Custom Tag Libraries

Cloud & DevOps:

- AWS: EC2, S3, RDS, Elastic Beanstalk, CloudFormation, SNS, SQS
- Azure: Web App Services, Security Integration
- Docker & Kubernetes for container orchestration
- CI/CD Pipelines: Jenkins, Git, Maven, VSTS

Database & Storage:

- PostgreSQL, MySQL, MongoDB, Cassandra, Oracle 9i
- PL/SQL Stored Procedures, Triggers, JDBC Connectivity

Messaging & Streaming:

- Apache Kafka, RabbitMQ, AWS SNS & SQS

Testing & Monitoring:

- JUnit, Mockito, Protractor, Mocha, Chai, Karma, Rational Clear Case
- CloudWatch, ELK Stack, Postman for API testing