Ibrahim Olanrewaju

Senior Software Engineer Team Lead

+2348094227050 | olanrewajui2020@gmail.com | Lagos, Nigeria

PROFESSIONAL SUMMARY

A team-lead senior software engineer with over 8 years of experience developing highly scalable, reactive, and distributed systems for high-volume businesses. With the ability to code and suggest changes to make the finished project glitch-free, as well as the confidence to make ideas heard, A demonstrated ability to maintain and improve company software in order to improve operational functionality in accordance with business requirements. with solid engineering professional abilities.

TECH STACK

FRONTEND	CSS3 8 years, HTML5 8 years, Bootstrap 5 years, JavaScript 8 years, jQuery 7 years, Angular 3 years, Vue.js 2, React 3 years, JSP 8 years, Thymeleaf 6 years
BACKEND	Scala 2 years, Java 8 years, PHP CodeIgniter 3 years, Node.js 6 years, Typescript 5 years, Go 1 year Play2 Framework 5 years, Spark Framework 6 years, Spring Framework 6 years, Kafka, RabbitMQ, BullMQ, Microservices, Docker, Kubernetes
DATA STORAGE	MySQL, PostgreSQL, MongoDB, Redis, and Cassandra
SERVERS/DEV OPS	Kubernetes, Amazon ECS, ECT, EC2, S3, Docker, GitHub Action, Jenkins, CircleCI, Git, GCP, Nginx
CLOUD SERVICES	MS Azure, AWS, DigitalOcean, Google Cloud
DEPENDENCE BUILD TOOLS	Maven, SBT, NPM, Composer
APIs PROTOCOL TYPE	SOAP, Restful, GraphQL, gRPC
METHODOLOGIES IN SOFTWARE	Object-Oriented Analysis & Design, SOLD Principle & Design Patterns, Rest Style Architecture Software Development Life Circle, Design Microservices Architecture with patterns & principles
PARADIGMS	OOP Programming, Functional Programming, Current Programming
OPERATING SYSTEMS	Ubuntu, Mac OS, Windows 7,8,10

DETAILS SUMMARY

- Led a team of 6 software engineers through the software development life cycle for 16 client projects.
- Continuously improve the deployment process.
- Develop industry and product technical expertise by following and championing development best practices.
- Troubleshoot, optimise and tune for performance.
- Work with the development team to establish coding standards and monitor guidelines that provide prompt feedback to allow for timely adjustment and revalidation.
- Establishing and executing configuration management solutions for Node.js(Javascript and Typescript), Spring Boot, Spring Cloud, Play 2 Framework, and PHP Codeigniter development environments.
- Provide expertise, direction, and ownership of configuration management policy and methodology.

- Design, develop, and implement complete configuration management policies and processes to support multiple product lines and services.
- Strives to reduce complexity and assist development, QA, and engineering by utilising Source Control and CM tool sets to decrease their time to value.
- Be responsible for creating the technical approach to projects throughout all phases of the software development life cycle.
- Technically lead and mentor engineers on best practices; mentor and inspire more engineers to collaborate. exceptional oral and written communication skills; the ability to form and maintain effective working relationships with team members and cross-functional teams.
- Strong understanding of the software and systems development lifecycle; experience with continuous delivery and integration is preferred.
- Find and help fix bugs and stability/scalability issues in a very time-critical environment.

EXPERIENCE

Team Lead & Senior Backend Software Engineer | OrnaMap & DCP | Lagos, Nigeria, Sept 2022– Present
website: https://ornamap.com

Experience Summary:

- As a team lead, I promoted a collaborative environment by encouraging team members to share ideas and knowledge. Held regular meetings, code reviews, the use of design patterns, and brainstorming sessions to ensure alignment and continuous improvement.
- Works within a small, focused team using agile methodologies (Scrum) to manage the development process, including sprint planning, daily stand-ups, and retrospectives.
- Provided technical guidance and mentorship to junior developers in the team, helping them overcome challenges and grow their skills.
- Maintained clear communication with stakeholders, offering regular updates on project progress and addressing concerns promptly. Ensured project deliverables met stakeholder expectations and quality standards.
- Successfully managed project timelines, delivering the DCP microservices architecture such as Gateway Service, E-Commance Service, Notification Message Service, User and Business Management Service, Product QR Code Verification Service, Product QR Code Generation Service, E-wallet Services, and Financial Accounting Services on time and within budget, exceeding stakeholder expectations with a stable and reliable service.
- Achieved significant improvements in system performance and scalability through optimised architecture and efficient data handling, ensuring the platform could handle increasing loads and high transaction volumes seamlessly.
- Introduced comprehensive security measures to protect sensitive data and ensure regulatory compliance, facilitating smooth audits and building user trust through robust security practices.
- Facilitated the professional growth of team members, resulting in a highly skilled and motivated team. Foster a positive and innovative work environment, encouraging continuous learning and improvement.
- Assign tasks based on individual team members's strengths and expertise, ensuring balanced workloads and timely completion of tasks.

PROJECTS

OrnaMap Service:
OrnaMap is a real-time schedule, trip fares, route suggestions, and navigation for Lagos Blue Line Metro, BRT Buses, Lagos Ferry Service, and Lagos - Ibadan Train Service.

As a Senior Backend Software Engineer at OrnaMap, I have been instrumental in designing and implementing robust backend services that drive our transit and mapping applications. My role involved leveraging a variety of technologies and tools to enhance the functionality, security, and user experience of OrnaMap services.

- Developed and maintained high-performance backend services using Spring Boot, ensuring high performance and maintainability.
- Employed MySQL with Java Persistence API (JPA) for efficient and reliable data management.
- Implemented mechanisms to blacklist and store IPs exceeding rate limits, enhance security, and protect against abuse.
- Designed and developed a comprehensive transit ticket vending service. Integrated with third-party ticket issuers through the Cowrry API, ensuring seamless ticket purchases and validations.
- Integrated payment gateway services with Flutterwave and Paystack to support diverse payment options, improving user convenience and expanding market reach.
- Implemented OAuth2 for instant user registration and authentication, significantly enhancing security and the user experience.
- Introduced and implemented the Multi-criteria Range Raptor algorithm to improve transit route suggestion accuracy. This enhancement has significantly optimised travel times and route selections for users.
- Integrated GraphQL to provide flexible querying capabilities. This allows for detailed and dynamic report generation, enabling clients to request specific data and reducing the overall payload size for increased efficiency.
- Integrated with the Google Maps API to pull location data, enhancing the accuracy of transit route suggestions and user navigation.
- Implemented request caching to minimise the frequency of calls to the Google Maps API. This optimisation saved the organisation 20% in API usage costs.
- Utilised OpenStreetMap (OSM) and General Transit Feed Specification (GTFS) files to store and manage transit records efficiently.
- Developed a solution to convert GPX files to GTFS shapes, enabling the integration and use of diverse geographic data sources.
- Exposed a variety of endpoints for the OrnaMap mobile app, including. Endpoints for signup, login, OTP verification, token generation, and refreshing. Endpoints for managing user profiles and retrieving transaction reports. Endpoints for recent trips, nearest terminals, fastest routes, and available routes for various transit options such as train, ferry, and BRT bus.
- Developed and implemented custom exception-handling mechanisms to enhance error management and improve the overall user experience. This approach ensures that the system gracefully handles errors and provides meaningful feedback to users.
- Through these initiatives, I have contributed to making OrnaMap a reliable and user-friendly platform for managing transit services while also ensuring security, scalability, and efficiency.
- Published transit data to Amazon S3, ensuring scalable and reliable data storage and retrieval.
- Deployed services in Docker containers to ensure consistency across development, testing, and production environments.
- Established a continuous integration and continuous deployment (CI/CD) pipeline to streamline deployments and reduce downtime, using Jenkins for automation.

- **Technologies**: Java, Spring Boot, Eureka Client, GitHub Actions, Maven, JUnit, Mockito, Memcached, Docker, GraphQL, Kafka, MongoDB, Amazon S3, Amazon EC2, Docker.

DCP Microservices architecture:

DCP services offer seamless communication, secure payments, verified products, and convenient e-commerce services, all in one integrated platform.

Experienced and results-oriented team leader of software engineers with a proven track record of leading high-performing teams to deliver robust, scalable, and secure software solutions. Expertise in developing and managing complex both synchronous and asynchronous microservices architectures, including gateway services, financial accounting systems, e-wallet services, product code generation and verification services, and user and business management services. Adept at leveraging modern technologies and best practices to drive project success and foster team growth.

DCP Gateway Service

As a team lead, we were responsible for the development of a robust gateway service using Spring Cloud Gateway. This service acted as a central hub for routing requests to various microservices, ensuring security, load balancing, and efficient communication. The technology stack included JWT for token-based authentication, Redis for token storage, Kafka for event-driven messaging, and Spring

Data Elasticsearch for efficient search capabilities. The service also utilised Eureka for service discovery, Spring Cloud LoadBalancer for load balancing, Docker for containerisation, and GitHub Actions for CI/CD.

- We designed and implemented a scalable asynchronous and synchronous microservices architecture using Spring Cloud and leveraged Eureka for service discovery to route incoming requests to appropriate microservices based on service names in the URL path.
- We used FeignClient to consume REST API endpoints exposed by third-party services or other microservices, facilitating smooth integration and data exchange.
- Implement a declarative HTTP client integrated with Ribbon for load balancing and Hystrix for circuit breakers to handle failures gracefully and provide fallback mechanisms.
- Set up the Eureka Server as a service registry where all microservices register themselves.
- We implemented JWT for secure token-based authentication, storing tokens in Redis for fast access and validation.
- We implemented OAuth2 for instant user registration and authentication, enhancing security and the user experience.
- We used rate-limiting strategies to protect the gateway from DDOS attacks, ensuring stability and reliability.
- We implemented Spring Cloud Config, which offers server and client-side support for externalised configuration in a distributed system.
- We integrated Elasticsearch Kafka to enable efficient search operations across various services.
- Utilised Kafka for event-driven messaging, listening to incoming messages (e.g., generate batch product, activate the batch, batch compromise) and routing them to the notification service and other relevant services. Ensured secure communication with Kafka using client identity and token authentication, maintaining the integrity and security of the messages.
- We configured the Eureka server for service registration and discovery, enabling seamless interaction between the gateway and other microservices, and used Spring Cloud LoadBalancer to distribute requests evenly across instances, enhancing performance and reliability.
- We exposed a multitude of API endpoints connected to various microservices, including financial accounting services, e-wallet banking services, product protection unique code and QR code generation services, and more.
- Set up a CI/CD pipeline using GitHub Actions, automating the build, test, and deployment processes to ensure rapid and reliable deployments.
- We ensured role-based access controls to restrict access to sensitive endpoints, maintaining data security and compliance with industry standards.
- We containerised the gateway service using Docker, ensuring consistency across different environments (development, testing, and production).
- We developed comprehensive unit and end-to-end tests using Spring Test libraries, ensuring the reliability and robustness of the service.
- **Technologies**: Java, Spring Cloud, Eureka Client, Hystrix, Ribbon, GitHub Actions, Maven, JUnit, Mockito, Apache JMeter, Redis, Docker, GraphQL, Kafka, Elasticsearch, Amazon S3, Amazon EC2, Docker.

DCP User and Business Management Services

As a team lead of software engineers, I had the opportunity to lead the development of the User and Business Management Services project. This service aimed to provide a comprehensive solution for managing individual users, stores, businesses, government agencies, and administrators, along with implementing robust security measures, efficient data handling, and seamless integration with external systems. The project was built using a modern tech stack that included TypeScript, Node.js, Express, MongoDB with Mongoose, Kafka, Joi for request validation, Jest and Mocha for testing, JWT for authentication, Eureka client for service discovery, GraphQL for efficient querying, Docker for containerisation, and a CI/CD pipeline.

- We designed a scalable and maintainable user and business service. Utilised the Eureka client for efficient service discovery and communication between microservices.
- We Implemented MongoDB as the primary database, utilising Mongoose for robust data modelling and validation.
- Integrated GraphQL to provide flexible and efficient querying capabilities, maximising the number of endpoint calls and optimising the response payloads.
- Integrated Kafka to handle asynchronous tasks such as sending OTPs and email notifications, ensuring efficient processing and scalability.
- We Implemented JWT for secure authentication and authorization, ensuring that only authenticated users could access sensitive endpoints.

- We implemented mechanisms for rate limits for API requests and blacklist IPs that exceeded the rate limits, protecting the system from abusive traffic and potential security threats.
- We designed a flexible role-based access control (RBAC) system for managing user roles and privileges, allowing dynamic assignment and revocation of roles.
- We developed a custom exception-handling mechanism to improve error management and enhance the user experience.
- Exposed RESTful APIs for various endpoints such as user signup, login, OTP generation and verification, generate token, refresh token password management, given privileges, revoking privileges, and user, business, and administrator management.
- Utilised Joi for request validation, ensuring data integrity and correctness.
- Implemented comprehensive unit and integration tests using Jest and Mocha, ensuring the application was thoroughly tested and reliable.
- Established a robust CI/CD pipeline using Jenkins, automating the build, test, and deployment processes. This ensured that new features and updates could be deployed rapidly and reliably.
- Containerised the application using Docker, ensuring consistency across development, testing, and production environments. Managed multiple containers using Docker Compose.

- **Technologies**: Node.js, Express.js, Typescript, Jest, Docker, GraphQL, MongoDB, Amazon S3, Amazon EC2, Amazon, Docker, Kafka, Swagger.

DCP Notification Message Service

As a team lead, I spearheaded the development of a comprehensive notification message service using Node.js with TypeScript and Express.js. This service was designed to handle various types of notifications, such as OTPs, password changes, account activations, and more, integrating seamlessly with backend services. The project leveraged a modern technology stack, including JWT for authentication, Joi for request validation, Jest and Mocha for testing, Eureka client for service discovery, GraphQL for reporting, Docker for containerisation, the CI/CD pipeline, and MongoDB with Mongoose for storage. Kafka was used for messaging event-listeners from other services, and the service was deployed on AWS EC2 instances.

- We designed a scalable microservices architecture with Eureka for service discovery, enabling efficient communication and integration between services.
- Implemented MongoDB as the primary database, using Mongoose for schema definitions and validations to ensure data integrity and consistency.
- Integrated GraphQL to provide flexible and efficient querying capabilities, allowing users to generate detailed message report analyses.
- Implemented JWT for secure token-based authentication, ensuring that only authorised users could access the endpoints.
- Employed express rate- limits to protect the service from DDOS attacks and blacklisted IPs that exceeded the rate limits.
- Secured Kafka with client token identity authentication to ensure only authorised services could produce and consume messages.
- Exposed RESTful APIs for various endpoints such as receiving notification messages, managing templates, and scheduling notifications.
- Developed APIs for scheduling notifications to be sent immediately or at specified intervals, providing flexibility in message delivery.
- Created endpoints for managing ready-to-use templates for various notification types, such as OTP, SMS, email, and chat messages.
- Developed a custom exception-handling mechanism to improve error management and enhance the user experience.
- Integrated Kafka for handling asynchronous notification messages, ensuring efficient processing and delivery.
- Utilised BullMQ for managing job queues, allowing for prioritised processing of messages based on business rules.
- I used Joi for request validation, ensuring incoming data adhered to expected formats and business rules.
- Implemented validation mechanisms to ensure messages met business rules, with bulk messages requiring approval from authorised administrators.
- Integrated Sentry for error and performance monitoring, ensuring issues were tracked and addressed promptly.
- Implemented logging of all notification activities, allowing for detailed tracking and analysis of sent messages.
- Implemented comprehensive unit and integration tests using Jest and Mocha, ensuring the application was reliable and bug-free.
- Set up a robust CI/CD pipeline using Jenkins, automating the build, test, and deployment processes. This ensured rapid and reliable deployments to AWS EC2 instances.

- Containerised the application using Docker, ensuring consistency across development, testing, and production environments.
- **Technologies**: Node.js, Express.js, Typescript, Jest, Docker, GraphQL, MongoDB, BullMQ, Amazon EC2, Amazon, Docker, Kafka.

DCP Product QR Code Verification Service

As the team lead of a group of software engineers, I spearheaded the development of the Product Unique QR Code Verification Service. This service was designed to ensure the authenticity of products, track counterfeit products and user interactions, handle large data batches, and generate detailed reports. The project utilised a combination of cutting-edge technologies, including Java Spring Boot, Cassandra, Kafka, Validator, JWT, Eureka client, GraphQL, Docker, the CI/CD pipeline, and comprehensive security measures such as IP blacklisting and rate limiting.

- We designed a product QR verification service to enhance the system's scalability, maintainability, and fault tolerance. Utilised Eureka for service discovery, enabling efficient inter-service communication.
- Implemented Cassandra as the primary database for storing product QR code information, leveraging its high performance for read and write operations. Ensured fast access to large volumes of data, essential for the verification service.
- Developed RESTful APIs using Spring Boot to handle QR code verification, user activity tracking, and other service functionalities. We ensured these APIs were secure, efficient, and easy to use.
- Integrated Kafka to handle asynchronous processing and real-time data streaming. This enabled efficient processing of large data batches and communication between microservices.
- Implemented background processing for large requests using the @Async annotation, improving responsiveness and user experience.
- Implemented JWT for secure user authentication and authorization, ensuring that only authenticated users could access the service.
- Developed mechanisms to limit the rate of API requests and blacklist IPs that exceeded the limits, protecting the service from abusive traffic and potential security threats.
- Designed a flexible role and privilege management system, allowing dynamic assignment and revocation of roles. This ensured that users had the appropriate level of access based on their roles.
- Offered endpoints for loyalty points redeemable for rewards.
- Developed a custom exception-handling mechanism to improve error management and enhance the user experience.
- Utilised Spring Boot's Validator to ensure data integrity and correctness, preventing invalid data from entering the system.
- Implemented comprehensive unit and integration tests using JUnit and Mockito. Conducted performance testing to identify and resolve system bottlenecks, ensuring the application could handle high transaction volumes.
- Integrated GraphQL to provide flexible and efficient querying capabilities. This enabled the generation of instant reports on product and user activity data, offering valuable insights to clients.
- Established a robust CI/CD pipeline using Jenkins, automating the build, test, and deployment processes. This ensured that new features and updates could be deployed rapidly and reliably.
- Containerised the application using Docker, ensuring consistency across development, testing, and production environments. Managed multiple containers, including Cassandra, Kafka, and Zookeeper, using Docker Compose.
- **Technologies**: Java, Spring Boot, Eureka Client, GitHub Actions, Maven, JUnit, Mockito, Apache JMeter, Redis, Docker, GraphQL, Kafka, MongoDB, Amazon S3, Amazon EC2, Docker.

DCP Product QR Code Generation Service

As the team lead of software engineers working on the Product QR Code Generation Service, I was responsible for overseeing the entire development lifecycle of a robust and scalable service. This service was designed to manage product information, generate unique codes and QR codes, and ensure security through various advanced technologies. My role involved project planning, team coordination, technical guidance, and ensuring the successful deployment and maintenance of the service.

- We designed a scalable microservices architecture using Spring Boot, with Eureka for service discovery and Kafka for asynchronous communication.
- Utilised MongoDB with JPA for dynamic schema handling and efficient data storage and retrieval.

- Implemented JWT for secure authentication and authorization, along with role-based access control (RBAC) for flexible user privilege management.
- Conducted regular code reviews to ensure adherence to coding standards and best practices, promoting high-quality code and maintainability.
- Develop core features such as product management, unique code generation, QR code generation, and more.
- Developed RESTful APIs to expose functionalities for creating, updating, disabling products, and generating unique codes in batches.
- Optimised database queries and implemented caching mechanisms Spring Memcached, improving system performance and response times.
- Implemented comprehensive security measures, including JWT authentication, IP blacklisting, and data validation, ensuring the application adhered to industry standards and protected sensitive data.
- Implemented functionality to generate unique codes in batches and configured endpoints to handle the process.
- Integrated Kafka to send product batches to a separate service for generating unique QR codes for each product.
- Set up Kafka listeners to consume messages from the financial accounting service, flagging products for QR code generation once payment is confirmed.
- Developed custom rate-limiting logic to monitor and control API request rates, protecting the system from abuse.
- Implemented mechanisms to identify and block IP addresses that exceed rate limits, enhancing overall security.
- Designed and implemented a dynamic role and privilege management system, allowing easy assignment and revocation of roles.
- Developed a custom exception-handling mechanism to improve error management and enhance the user experience.
- We ensured robust access control using RBAC to manage permissions and access levels across different parts of the application.
- We conducted performance testing to identify and resolve bottlenecks, ensuring the application could handle high volumes of transactions efficiently.
- Implemented unit and integration tests using JUnit and Mockito to ensure robust test coverage and early detection of issues.
- Containerised the application using Docker, ensuring consistent development and deployment environments.
- Established a continuous integration and continuous deployment (CI/CD) pipeline using Jenkins to automate testing, building, and deployment processes.

- **Technologies**: Java, Spring Boot, Eureka Client, GitHub Actions, Maven, JUnit, Mockito, Apache JMeter, Redis, Docker, GraphQL, Kafka, MongoDB, Amazon S3, Amazon EC2, Docker.

DCP E-commerce Service

As a team lead of software engineers, I led the development of an e-commerce management service using Node.js with TypeScript and Express.js. The service was designed to manage a comprehensive e-commerce system with multiple endpoints, focusing on product search, cart management, checkout processes, user accounts, customer support, and advanced administrative features. The tech stack included JWT for authorization, Joi for request validation, Jest and Mocha for testing, Eureka client for service discovery, GraphQL for reporting, Docker for containerisation, a CI/CD pipeline, and MongoDB with Mongoose for storage. Kafka was utilised for event-driven messaging, and the service was deployed on AWS EC2 instances.

- We designed and implemented a scalable e-commerce service with Spring Boot using Eureka for service discovery, enabling efficient communication between services.
- We utilised MongoDB as the primary database, leveraging Mongoose for schema definitions, data validation, and managing complex queries.
- Integrated GraphQL to provide flexible querying capabilities, allowing detailed and dynamic generation of reports and allowing clients to request exactly the data they need, improving efficiency and reducing the payload size
- We implemented JWT for secure token-based authentication, ensuring that only authorised users could access the endpoints.
- Used express-rate-limit to prevent DDOS attacks and implemented IP blacklisting to secure the service from abusive traffic.
- We ensured secure Kafka communication with client identity authentication, maintaining the integrity and security of event-driven messages.
- We developed RESTful APIs for various functionalities, including product search, cart management, order processing, and user account management.
- We implemented Joi for request validation, ensuring data integrity and compliance with business rules.
- Set up comprehensive unit and integration tests using Jest and Mocha, ensuring the reliability and stability of the application.

- Developed endpoints for searching and filtering products by various criteria like categories, price, brand, and ratings. Implemented an advanced search option for more specific queries.
- We created endpoints for displaying product thumbnails and detailed information pages, including images, reviews, and specifications.
- We implemented functionality for adding products to the cart, managing cart items, and viewing cart summaries.
- Developed a seamless checkout process with guest checkout options, account creation, secure billing and shipping information handling, multiple payment options, and order confirmation.
- Managed user accounts with features for order history tracking, saving addresses, storing payment information, and accessing loyalty programmes.
- Developed features for tracking stock levels, bulk uploading products, and categorising products into various categories and tags.
- We implemented order processing, invoice generation, and shipping management, integrating with shipping providers for tracking deliveries.
- We built tools for accessing user profiles, segmenting customers for targeted marketing, and sending newsletters and promotional emails via the Kafka message-to-notification service.
- We created detailed sales reports, customer insights, and inventory reports to monitor performance and support decision-making.
- Enabled the creation and management of discount codes, moderation of product reviews, and use of SEO tools for optimising product listings.
- Ensured compliance with PCI-DSS standards for secure payments, implemented SSL encryption for data protection, and established role-based access controls.
- Supported integration with third-party apps, provided APIs for custom integrations, and synced inventory with major marketplaces like Aliexpress and Jumia.
- Established a robust CI/CD pipeline using Jenkins, automating the build, test, and deployment processes. This ensured rapid and reliable deployments to AWS EC2 instances.
- We containerised the application using Docker, ensuring consistency across development, testing, and production environments.

- **Technologies**: Node.js, Express.js, Typescript, Jest, Memcached, Docker, GraphQL, MongoDB, Amazon S3, Amazon EC2, Docker, Kafka, Swagger.

DCP E-wallet Services

As a team lead of software engineers working on an E-wallet service, my role involved guiding a team through the design, development, and deployment of a comprehensive and secure E-wallet service. This service integrates multiple advanced technologies such as Java Spring Boot, MySQL with JPA, Kafka, Validator, JWT, RSA, Eureka client, Docker, CI/CD pipelines, IP blacklisting, rate limiting, and third-party payment gateways (Flutterwave and Paystack) along with Polaris Bank API integration. Below is a detailed account of my responsibilities and achievements in this role.

- Spearheaded the development of our backend services using Java Spring Boot. We leveraged MySQL with JPA for efficient and reliable data management.
- Ensured ACID transactional integrity for critical operations like funding wallets, withdrawals, and transfers, maintaining consistency and reliability, and placing a lens on the wallet account while doing the financial transactions process.
- Developed a custom exception-handling mechanism to improve error management and enhance the user experience.
- Integrated Kafka to handle event-driven messaging, allowing real-time processing of transactions and checking balances, withdraws, and fund wallets.
- Enabled horizontal scaling of services to manage high transaction volumes seamlessly.
- Implemented secure authentication using JWT and RSA, ensuring robust protection of user data and transactions. Enhanced security with options for SMS/app-based 2FA and biometric authentication for user logins.
- Implemented IP blacklisting to protect against malicious attacks. Established rate limits to prevent abuse and ensure service availability.
- We ensure all sensitive data is encrypted in transit and at rest.
- Utilised Eureka for service discovery, facilitating efficient communication between microservices.
- Deployed fraud detection algorithms to monitor transactions for suspicious activity and trigger alerts, ensuring a secure environment for both users and merchants.
- Simplified account creation with options to link existing bank accounts or credit cards.

- Allowed users to manage multiple e-wallets or bank accounts from a single user profile.
- Facilitated peer-to-peer transfers and merchant payments both online and at physical locations using QR codes.
- Integrated KYC and AML processes to meet regulatory requirements and ensure all payment transactions adhered to PCI-DSS standards.
- Enabled adding funds from various sources, withdrawing to bank accounts or ATMs, and setting up scheduled transfers.
- We successfully integrated with payment gateways like Flutterwave and Paystack for seamless transactions and integrated with the Polaris Bank API to expand our banking services.
- Offered real-time transaction monitoring and detailed reporting to help merchants track their sales and customer activity.
- We integrate POS payments at physical stores and enable easy invoicing and billing for merchants.
- We delivered insights into sales trends, customer behaviour, and KPIs through real-time dashboards and forecasting tools.
- We implement cashback on purchases and loyalty points redeemable for rewards.
- Established a CI/CD pipeline using Jenkins for automated testing, integration, and deployment, reducing deployment time and minimising errors.
- Deployed applications in Docker containers, ensuring consistent environments across development, testing, and production.
- Developed RESTful APIs using Spring Boot for seamless integration with Gateway and third-party services.
- Implemented unit and integration tests using JUnit and Mockito to ensure robust test coverage and early detection of issues.
- Conducted performance testing to identify and resolve bottlenecks, ensuring the application could handle high volumes of transactions efficiently.

- **Technologies**: Java, Spring Boot, Eureka Client, GitHub Actions, Maven, JUnit, Mockito, Apache JMeter, Redis, Docker, GraphQL, Kafka, MongoDB, Amazon S3, Amazon EC2, Docker.

DCP Financial Accounting Services

As a team lead of software engineers working on financial accounting services, my role encompasses overseeing the design, development, and deployment of a robust and secure financial accounting platform. Leveraging a diverse technology stack, including Java Spring Boot, MySQL with JPA, Kafka, Spring Security with JWT, Docker, and CI/CD pipelines, I ensure our solutions are scalable, efficient, and secure. Below is a detailed account of my responsibilities and achievements in this role.

- Led the development of backend services using Java Spring Boot, ensuring high performance and scalability.
- Implemented ACID transactions to ensure consistency and reliability for all financial operations, such as recording transactions and managing the general ledger.
- Developed custom exception handling to enhance error management and the user experience.
- Integrated Kafka for event-driven messaging, enabling real-time processing of financial transactions and notifications. Facilitated the horizontal scaling of services to handle high transaction volumes seamlessly.
- Utilised JWT and RSA for secure authentication and authorization, protecting user data and financial transactions. Implemented IP blacklisting to enhance security and prevent malicious attacks. Set up rate limits to ensure service availability and prevent abuse.
- Developed systems to assign and revoke user privileges and roles dynamically.
- Implemented Eureka for service discovery to ensure efficient communication between microservices.
- Implemented sensitive financial data is encrypted both in transit and at rest to protect against breaches.
- Implemented fraud detection systems to monitor transactions and identify suspicious activities.
- Implemented financial reports that adhere to GAAP and IFRS standards. Generated reports required by regulatory bodies to ensure compliance.
- Managed a comprehensive list of all accounts used in the company's general ledger.
- Recorded and tracked all financial transactions, ensuring each transaction affected at least two accounts to maintain balance.
- Scheduled and processed payments to vendors via checks, ACH, and wire transfers.
- Created and sent invoices to customers, tracking incoming payments, and managing overdue accounts via Kafka.
- Tracked and managed company assets, including acquisition, depreciation, and disposal.
- Automatically calculated depreciation using methods such as straight-line and declining balance.
- Exposed endpoints for tracked revenue, expenses, and profits over specific periods, reported cash inflows and outflows over periods and generated tailored reports for management decision-making.
- Implemented automatically calculated taxes for various transactions, and prepared and filed tax returns, ensuring compliance with local regulations.
- Developed and monitored budgets, generated tailored management reports, and created visual KPIs.
- Automated invoicing and reconciliation, integrated with ERP systems, and utilised Spring AI cc prediction.

- Integrated payroll functionalities to manage employee compensation, ensuring compliance with tax regulations and labour laws.
- Calculated the cost associated with the production of goods and services sold.
- Used Docker to create consistent development, testing, and production environments, streamlining deployments.
- Established a CI/CD pipeline using Jenkins for continuous integration and deployment, reducing deployment times and minimising errors.
- Implemented unit and integration tests using JUnit and Mockito, ensuring robust test coverage and early detection of issues. Conducted performance testing to identify and resolve bottlenecks, ensuring the application can handle high volumes of transactions efficiently.
- **Technologies**: Java, Spring Boot, Eureka Client, GitHub Actions, Maven, JUnit, Mockito, Apache JMeter, Redis, Docker, GraphQL, Kafka, MongoDB, Amazon S3, Amazon EC2, Docker.

Senior Software Engineer | Turing Enterprises Inc | Los Altos, California, United States, June 2021– Oct 2022
website: https://www.turing.com

Turing.com - Hire Remote Developers | Silicon Valley calibre software engineers | Remote U.S. developer jobs | Full-stack, mobile, frontend, backend DevOps.

PROJECTS

Enhanced Cart Handling Optimizations (ECHO) Services
website: https://www.techstylefashiongroup.com

Role: Backend Developer
Is an eCommerce critical process to optimise the shopping cart for design and usability, in order to prevent cart abandonment and increase shopping cart conversion rates.

- Work in an agile environment and participate in conceptualising and implementing platform features for our global web properties.
- Works within a small focused team using Agile/Scrum principles to develop eCommerce services and platform features.
- Act in the capacity of an eCommerce application software engineer to actively participate in the definition, development, testing, test automation support and implementation of software functionalities.
- Implement a library of utilities to validate data integrity and sanitise data requests.
- Integrating with Site24x7 to enable monitoring of metrics, traces, and logs for different layers of cloud architecture from a single console Keep track of the availability and performance of services and servers.
- Integrate with Swagger to create, auto-document, test, and consume RESTful web services.
- Implemented features such as adding products by adding a product by session, adding bundle customer products by ID, adding products by cart ID, updating Cart Line By CartId, removing products by CartId, etc.
- Validates the provided cart membershipID token list against tokenQty.
- Transform from an MSSQL model to an ECHO DTO. It loops through (recursively) object keys to make them camelCaseString format.
- Set up process listeners for an abrupt app stop to close connections.
- Managed deployment CI/CD pipeline Github Jenkin AWS with Docker.
- **Technologies**: Node.js, Javascript, Jira, Jest, Docker, GraphQL, Restful, MSSQL, Amazon S3, Amazon EC2, Amazon-SNS, Amazon Dynamo DB, Docker, Redis, Swagger.

Team Lead | TravelBeta.com | Lagos, Nigeria | Contract

Website: https://travelbeta.com

Role: Team Lead

TravelBeta.com is one of the top 3 travel companies in Nigeria. It leverages its profit from its vendor competition by extracting the best price from all its suppliers. Likewise easing the booking and increases the overall customer after-sales experience.

Created and system that performs API calls for flights, hotels and other travel services service from two or more vendors; We aggregated the results and provided the best results based on the business's pre-defined conditions.

- Lead a team of 3 software engineers through the software development life cycle.
- Built a reservation website for a travel company that allows for flight and hotel searches, reservations, and payment.
- The backend service was redesigned to accommodate the integration of multiple GDS providers (Sabre and Amadeus) for flight information and TBOHolidays for hotel bookings and reservations.
- Rebuilt the CRM dashboard with ReactJS to accommodate the new API change.
- Containerised the web services with Docker, wrapped them around the Kubernetes cluster and deployed them to the Google Cloud Platform (GCP).
- Migrated the application from a monolithic application to a multi-service application to ensure application reliability.
- Created several HTTP header authentications with JWT.
- Used Kafka to stream results from different sources and consumers by different activities.
- **End Result:** Increase customer experience and trust in the business. Which increases overall sales.
- **Technologies**: Java, Spring Framework, Typescript(Node.js, Express), Kafka, Redis, MySQL, GCP, Docker, MongoDB, Kubernetes, ReactJS(Javascript).

Senior Backend Engineer | Ajala. ng | Lagos, Nigeria | Contract

Website: https://ajala.ng

Role: Senior Backend Engineer

Ajala.com is an online travel management company. The company provide travel services to its clients through the online channel. Through B2B and B2C business models, the company sells a wide range of services by providing individual platforms for use.

Created and system that performs API calls for flights, hotels and other travel services. Thus, we apply the business profit and incentive calculation to the prices obtained from the supplier via API and sell the item to the customer.

- Built a reservation website for a travel company that supports flight search and reservation for local flights.
- Built microservices APIs for flight and fraud-prevention services.
- Integrated Paystact payment gateway for payment and confirmation of flight reservations.
- Created several HTTP header authentications with JWT.
- Create a service to manage fraudulent activities and prevent them.
- Used Kafka to stream results from different sources and consumers by different activities.
- **End Result:** Effectively created a booking portal for customers to book flights and other travel services.
- **Technologies**: Java, Spring Framework, Kafka, Redis, MySQL, AWS, Docker, Kubernetes, Angular

Senior Software Engineer Team Lead | Fisshbone&Lestr Technology | Lagos, Nigeria, Feb 2018 – May 2021
website: https://www.fisshboneandlestr.com

Fisshbone&Lestr *is an ICT company that provides bespoke software for her parent company CELD.*

Create several solutions among them is an electronic reward commodity CashToken and guaranty cashback Ipoints, ESPI E.t.c. The system enables every patronage or gift to become a life-changing cash opportunity.

- Be a strong contributor working with product management, engineering, and QM partners to identify solutions, contributing to the overall design, proof-of-concept development, and technical strategy, and coordinating with external teams and vendors.
- Led a team of software developers in the creation of applications for clients such as reward systems, fanatic-as-a-service, message-as-a-service, financial accounting-as-a-service, eWallet-as-a-service, Chatbot-as-a-Service, and so on.
- Actively participate in all phases of the Agile Development process including definition, design, development, test, deployment, and support.
- My primary assignment at Fisshbone and Lestr is to work with a team of Senior Software Developers to improve the bulk CashToken gifting and chatbot (Fela) using Google Dialogflow and Kommunicate platform to interact with customers.
- Review, Design and Implement System Requirements.
- Managing and Integrating Third Party Systems/Services Integrations.
- Developed and implemented CI/CD pipelines to automate the deployment of applications.
- Helped scale the system to handle concurrent connections ranging from 5,000 to 15,000 active connections.
- Improved deployment process within AWS (ex. cross-region automated deployment).
- AWS services administration: IAM, VPC, EC2, S3, CodeDeploy, RDS, CloudWatch, CloudFormation, Elastic BeanStalk; VCS: GitHub, Docker, Scripting: Shell, Slack, Improve automated test and simulation frameworks.
- Develop and automate standard operating procedures around common failure scenarios.
- **Technologies**: Codeigniter, ReactPHP, Node.js, Typescript, Vue.js, Angular, React.js, Jira, Jest, Azure DevOps, GitHub, Bitbucket, Trello, Docker, Slack, GraphQL, Restful, MySQL, MongoDB, Amazon S3, Amazon EC2, Amazon-Route 53, Amazon-SNS, Azure VM, DigitalOcean, Docker, Redis, RabbitMQ, Strapi.

PROJECTS

CashToken Version One

website: https://www.cashtoken.ng

Role: Fullstack Developer

An electronic reward commodity system.

Maintains and Integrates the customer loyalty reward microservice with third-party clients.

- CashToken buck gifting was implemented using RabbitMQ as a queue system.
- We achieve an unbiased draw engine sorting system by generating a random permutation of a finite sequence and implementing a CashToken draw engine using the Fisher-Yates shuffle algorithm.
- The payment gateway was Integrated with the Futterwave platform.
- Written a custom ORM database persistence and database schema.
- Administrator backend for system report, analysis and activities.
- UniTest PHPUnit and Jest.
- Implement users' accessibility sessions, page menus, and privileges using Redis. This is to avoid having to make a request to fetch client information per request in the database. This increases the application's given privileges and revoking privileges performance by 25%.
- Implemented and refactored existing UI, which improved UI rendering speed significantly and drove a better user experience.
- Trained, improved and implemented Google Dialogflow intents to improve customer interaction with the Fela chatbot by 56%.
- AWS S3. We archive old CashToken customers' gifting data in the database that has a size of 20 million records or more and store it as an object in S3 buckets. We encrypt the records before upload, specify the version control, and also name the records with the client ID and date-time, in order to avoid overriding the data. Using S3 to back up outdated data improves the database's speed and efficiency in terms of updating and reading. When a customer requests records for an analysis report, we extract data from buckets.
- Ensuring optimal performance of the central database and responsiveness to front-end requests.
- Designed responsive UI using Bootstrap 4, and React.
- Understanding of React fundamentals was used to promote better component lifecycle practices, resulting in a 23% increase in turnaround time with 100% deadline adherence.
- Introduced more widespread use of isomorphic React and Node.js for web applications, which reduced load times by approximately 35%.
- Followed documentation to always remain up-to-speed on what needs to be updated in response to new release versions.
- Designed and implemented the backend with Node.js, Typescript and Codeigniter.
- Managed deployment CI/CD pipeline GitHub Action for cloud-host on GCP with Docker.

Centra Messaging Service

Role: Team Lead

Designed and maintained Central Message-as-a-Service.

- Header authentication client's request with a public and private keys pair, Bearer with signature Hash using SHA512 and IP whitelisting.
- The Template Service endpoint manages all ready-to-use templates for OTP, SMS, email, chat, and other push notification messages. It also provides REST APIs to create, update, disable, and manage templates. It also provides a UI dashboard page to check and manage message templates from the web.
- The Simple Notification endpoint exposes APIs to integrate clients with backend services. It's the main service, which will handle simple notification requests.
- The Bulk Notification endpoint exposes APIs to integrate clients with backend services. It's the main service, which will handle bulk notification requests.
- The Simple Notification endpoint exposes APIs to integrate clients with backend services. It's the main service, which will handle simple notification requests.

- The scheduling notifications endpoint provides APIs to schedule notifications immediately or at any given time. It could be any of the following: second, minute, hourly, daily, etc.
- Validation and prioritisation are solely responsible for validating notification messages against business rules and the expected format. Bulk messages should be approved by the authorised system administrator only.
- Event Priority Queues (Kue): It provides an event hub service, which will consume messages from notification services in high, medium, and low-priority topics. It sends processed and validated messages to the notification handler service, which internally uses the notification preferences service to check the users' personal preferences. It has three topics, which are used to consume or send messages based on business priority: high, medium, and low.
- Notification Messages: It manages notification messages. It persists in sending messages to databases and maintains an activity log. The same message can be resent using the APIs of these services. It will provide APIs to add, update, and read both old and new messages. It also provides a web dashboard with a filter option to filter messages based on different criteria, like date range, priority, module user, user group, etc.
- Exposing message report analysis endpoint using GraphQL.
- UniTest Mocha and Jest.
- Database persistence using Sequelize ORM and MySql Database schema.
- Designed and implemented the backend with Node.js.
- Managed deployment CI/CD pipeline GitHub Action for cloud-host on AWS (EC2) with Docker.

iPoints

Role: Team Lead

Designed and maintained Financial-as-a-Service and managed continuous development and Integration (DevOps).

- Ensured the programming team developed high-quality working applications for clients.
- Monitored all aspects of applications being developed to ensure they met quality standards.
- Held team meetings and prioritised work tasks.
- Ipionts' cash-back and buck-gifting were first designed using the divide and conquer approach, with a cron job to fire the processes. It was later improved by 36% by using RabbitMq to distribute the tasks among the workers.
- Integrated Futterwave Payment Gateway for clients, to buy Ipoints products and also subscript for insurance services.
- Exposing API endpoint for Ipoints wallet to wallet transfer, bulk Ipoints upload in CSV and over restful API, checking wallet balance, redeeming voucher etc. Those features are provided in web UI, API and USSD using AfricasTalking Services.
- Using client CSRF tokens to prevent CSRF attacks by making it impossible for an attacker to construct a fully valid HTTP request suitable for feeding a victim user.
- API herder authentication with JWT using RSA Public/Private key pairs.
- Ipoints transfer, checking wallet balance, redeeming vouchers etc. Those features are provided in the web UI, API and USSD using AfricasTalking.
- Implement auto insurance subscriptions by triggering a cron job on the start date of the insurance subscription, then pushing it to a RabbitMq to distribute the tasks among the workers.
- Implemented a comprehensive single insurance subscription. It checked a user's Ipoints accumulation to see if they were eligible for the insurance, then pushed the request to a RabbitMQ to distribute the tasks. It processed the task after setting its priority to "high" in the queue.
- Ipoints cash-back rest web-services API integrations with third-party clients ESPI.
- UniTest PHPUnit, Jest.
- Written a custom ORM database persistence and MySql Database schema.
- Designed and implemented the backend with Node.js and Codeigniter.
- Managed deployment CI/CD pipeline for cloud-host on AWS GitHub Action with Docker.

- Automated Testing, we make our code do some unit and integration testing. We set up our CI pipeline with the test, build, and dev-deploy jobs, which are meant to run unit and integration tests, if passed, then build to generate a deployable project, and then deploy to the dev environment so as to enable QA or developers to test before pushing to master/main or release branch. We use the CI/CD platform GitHub Action.

Espi

Role: Backend Developer

A financial accounting and payment processor as a service.

Maintains and maintains Financial Accounting-as-a-Service microservice. Working alongside a team of developers in security aspects and transactions.

- To ensure the secure exchange of data between the payment provider's platform and the eSPI platform payment providers would be required to implement security measures, especially where the payment webhook is called.
- Implement hashing of the request and response parameters using the SHA512 algorithm, with keys exchanged during implementation.
- IP Lockdown between the eSPI platform and the payment providers such that every request would be required and validated that it's coming from the provided Payment Providers IP address(es). The exchange of IP addresses occurs during implementation.
- UniTest Jest.
- Provisioning of a username and password exchanged during implementation, with both passed as part of the request parameters.
- Implement credit and debit wallet transactions.
- Implement wallet-to-wallet transfer.
- Designed and implemented the backend with Node.js and Typescript.

MyBankUSSD

website: https://mybankussd.com

Role: Team Lead.

MyBankUSSD is Africa's premier 'All African Banks' USSD Service aggregator.

Designed an eWallet & Banks USSD System for payment services, Payment-as-a-services. Working alongside a team of developers.

- Estimate engineering efforts, plan implementations and roll out system changes.
- Ensures developed software is handed off to QA for further testing.
- Implemented the client payment payload with Sha256 hash, which is used for passing secure values to the payment gateway.
- API herder authentication with JWT using RSA Public/Private key pairs and also whitelist IPs is a security feature often used for limiting and controlling access only to trusted users.
- Loading the payment UI on Iframe and listening on click to closing tag, URL navigation buttons in other give warning or terminate the payment process.
- Implement Register number for USSD (All Nigerian Banks), Buy Airtime, CashToken & Utilities, Local Transfers, Account Opening, Fund MyUSSD Wallet, Payments @ Merchant Locations (Online & Offline), Access to Credit & Loans and Check Balances & BVN.
- Implement Payment APIs and Dedicated USSD Strings for Direct Dials and Rewards Wallet (Unified Reward System: CashToken).
- Implemented direct USSD integration with telecommunications resulting in a 23% reduction in service costs.
- UniTest Mocha and Jest.
- Designed and implemented a front-end UI with Vue.js.
- Designed and implemented the backend with node.js

AirtelThanks USSD

Role: Team Lead.

Airtel telecommunication in Nigeria partners with CashToken as a reward commodity.

- Integrated USSD endpoint Airtel using the site-to-site VPN infrastructure to create a tunnel between two networks so that traffic is not exposed to the public internet.
- Storing and tracking users' activities were implemented by Redis hash.
- Reduced USSD data exchange size by 34%, by designing a proprietary message compression system.
- Integrated directly with the Tech which by implication reduced service cost by 15%.
- Improved payment UI rendering speed significantly and drove a better user experience.
- Configuring VPN infrastructure also improves upon things like bandwidth and efficiency by 38%. Better performance is something no Internet user would ever argue with.
- Implemented users' activities with Redis hash to improve the speed of instant accessing user information by 48%.
- Designed and implemented a front-end UI with React.
- Designed and implemented the backend with Node.js Typescript with Express.js and MongoDB.
- Implement integration and unit test with Jest, supertest and MongoDB-memory-server.

Fela MarketPlace

Role: Team Lead.

Fela is the world's 1st Multi-Tenant Chatbot-as-a-Service Marketplace on Social Media. Fela Market Place is a platform hosted on WhatsApp and online where buyers can purchase goods and services and will be rewarded with CashToken.

- Improving customer relations by integrating Natural Language Processing systems into the existing chatbot. I did a week's research on the best engine then we moved to execution from there. I ended up adding the Dialogflow Artificial Intelligence (AI) support, making the chatbot available on Telegram, Facebook, Web using Kommunicate (web), and WhatsApp using the Pickyassist platform.
- Providing features such as buying airtime & data, Electricity payment, cable TV payment, Buying CashToken, Top-up Fela eWallet, checking wallet balance and redeeming with third-party API.
- Managed deployment CI/CD pipeline on Firebase Google Cloud.
- Trained and improved Google Dialogflow intents to improve user interaction by 56%.
- CI/CD pipeline improves the application development process, particularly in the integration and testing phases, as well as during delivery and deployment.
- A week of research gave me complete flexibility in their approach to a problem. Novelty and creativity are encouraged, it gives an insight into the tools and languages that are best suited to the job.
- Using Entities to identify and extract useful data from natural-language inputs.
- Using the Joi library to validate user requests improved the data integrity by 47%.

CashToken Version Two

website: https://ng.cashtoken.africa

Role: Team Lead.

An electronic reward commodity system.

Design and Integrate the customer loyalty reward microservice with third-party clients with regional platforms (Nigeria, Ghana, Zambia and Ivory Coast). Working with a team of developers to work as frontend and backend.

- Establishing effective working relationships with team members and cross-functional teams.
- Established using the Agile SCRUM methodology. Involved in daily stand-up meetings, sprint showcases, and sprint retrospectives.
- Interact with other developers' teams and end users to design and implement new framework features.

- Worked in a 5 layering structured team environment to develop, enhance, and support web-based products.
- Create complex JIRA workflows including project workflows, screen schemes, permission schemes and notification schemes in JIRA.
- Involved in designing the user experience interface (UI/UX strategy) and converting findings into UI designs.
- Leveraged an MVC design pattern to organize Angular controllers, Custom directives, factories and views.
- Worked on the REST/Web API to create the services and tested them on Postman and then used in Angular, HTTP service calls and bound the data in the table using ng-repeat attributes.
- Used Angular as a framework to create a Single Page Application (SPA) which can bind data to specific views and synchronize data with the server using Sass, LESS, Bootstrap, and Angular.
- Version control systems, such as Subversion (SVN), CVS, Perforce, and Visual Source Safe, were extensively used.
- Experience in CSS pixel-level layout, consistency with browsers, version and platform independence.
- Created animation effects during the data load while waiting for the Ajax response using JQuery.
- Debugging using Firebug and web developer tools on Chrome, Firefox, and Opera.
- Involved in the creation of complex features such as a landing page, a client dashboard and an admin dashboard, report analysis, purchasing CashToken, withdrawing money, and so on.
- Working knowledge of search engines such as Elastic Search is a plus.
- Integrated payment service with myBankUSSD.
- One-way sign-up and login were implemented with OAuth 2.0 and OpenId connect.
- Using GraphQL to fetch and update data in a declarative manner. Put differently, we climb up one step higher on the API abstraction ladder and don't have to deal with low-level networking tasks ourselves anymore.
- Provided environment configuration based on a country and also customised the build in angular.json to build with a designated country name on CLI.
- Implement unit and integration tests using UniTest Jasmine, Arma, and Jest.
- Managed deployment CI/CD pipeline DevAzure and cloud-hosted Google Firebase and GCP.
- Designed and implemented a front-end UI with Angular and React.
- Designed and implemented the backend with Node.js and Typescript with Express.js.

Senior Software Engineer (Fullstack) | Travelden | Lagos, Nigeria, Dec 2017 – Jan 2018

Website: https://finchglowtravels.com

Travelden.com is a travel wing of Finchglow Group. It provides flight hotels and other travel services through the online sales channel

- Developed a Travel and Tourism web app with a team of developers. My responsibilities are implementing the database persistence, database design, GDS Integration (the main core of the software) and Interactive UI.
- Developed a flight ticket booking, hotel and tourism web application by integrating Amadeus GDS API.
- Process automation for the Aviation Management Web App for the Lagos Aviation Academy My responsibility is to build software from end to end. It includes presentation layout UI, business logic, database design, and unit testing.
- **Technologies**: Java, Scala, Play 2 framework, Spring boot, Angular.js, CodeIgniter, Restful, Bitbucket, MySQL, MongoDB, Amazon EC2.

PROJECTS

Travelden

website: https://www.travelden.com/

Role: Frontend And Backend Developer.

Created and system that performs API calls for flights, hotels and other travel services. Thus, we apply the business profit and incentive calculation to the prices obtained from the supplier via API and sell the item to the customer.

Developed B2B and B2C Online Travelling Agency (OTA) Booking Portal.

- Developed Distributed In-Memory Caching for the Booking Engine.
- Developed Audit and Sales Report Portal for Staff.
- Consuming and Provision of JAX-WS and JAX-RS API.
- Built microservice APIs for flight and fraud-prevention services
- Created several HTTP authentications authentication with JWT
- Create a service to manage fraudulent activities and prevent them
- Used Kafka to stream results from different sources and consumers by different activities
- **End Result:** Effectively created a booking portal for customers to book flights and other travel services.

Lagos Aviation Academy

website: https://www.lagosaviationacademy.com/

Role: Team Lead.

Developed the System from End-to-End. Aviation training solution. Ensuring all functionality meets business demands.

- Develop student enquiry and registration process.
- Develop SSO (Single Page Sign-In) Across Domains.
- Develop Real-Time Payment Analytics & Student Reporting System.
- Support Business Analysts in Develop Software Requirements.

Nogeltravels

website: https://nogletravels.com

Role: Frontend And Backend Developer

Team members to develop and improve booking flights and hotels.

- Integrated Rest and SOAP Web-Services APIs with Amadeus (GDS).
- Improved search speed by 43% by using Memcached to store flight results for session reusability.
- Integrated SOAP Web-Services APIs Rooms_xml distribution system.
- Database persistence using Ebean Java and Mysql.
- Designed responsive front-end UI with Bootstrap and Anguar.js

Junior Software Engineer (Fullstack) | Travelfix | Abuja, Nigeria, Oct 2016 – Dec 2017

website: http://travelfix.com

- Developed a travel and tourism web app with a team of developers as a Java developer. My responsibilities are to implement database persistence, database design, business logic and unit testing.
- **Technologies**: Java, Play 2 Framework, Spark Framework, MySQL, Bitbucket, Amazon EC2

PROJECTS

Travelfix

Role: Fullstack Developer

Team Member to Developing Travels and Tourism System Back-End.

- Team Member in Developing Travels and Tourism System Back-End.
- Built microservices APIs for flight and fraud-prevention services.
- Implement several HTTP header authorisations with JWT.
- Implement a service to manage fraudulent activities and prevent them.
- Database Persistence using Ebean Java and Mysql.
- Rest and SOAP Web-Services APIs Integrations with Sabar.
- Designed responsive front-end UI with Bootstrap and JQuery.

Intern Developer | Co-Creation Hub | Lagos, Nigeria, Nov 2015 – Oct 2016

Website: https://cchubnigeria.com

CcHUB is Nigeria's first open living lab and pre-incubation space designed to be a multi-functional, multi-purpose space where work to catalyze creative social tech ventures takes place. The HUB is a place for technologists, social entrepreneurs, government, tech companies, impact investors and hackers in and around Lagos to co-create new solutions to the many social problems in Nigeria.

- I worked with the Efiko team as a backend developer. My responsibilities include testing the functionalities of the application, logging bugs and issues discovered, and working with the team to make the appropriate corrections.
- I worked intimately with the EduChamber team as a backend developer. My responsibilities included database design implementation, testing applications for correctness, and business logic implementation.
- **Technologies**: PHP, Node.js, Jquery, HTML, CSS, MySql, Bitbucket

PROJECTS

Educhamber

Role: Backend Developer

Is an e-learning platform designed for examination preparation? It's a platform that targets students preparing for SSCE, NECO or JAMB examinations. It provides practice questions for all eligible subjects in these examinations.

Networking and Computer Engineer | Gallant System | Lagos, Nigeria, Oct 2014 – Nov 2015

- Assist staff and users in resolving computer-related issues such as malfunctions and program errors.
- Test maintain and monitor computer programs, systems, and networking, including the installation of computer programs and system maintenance.
- Coordination and linking of computer systems within an organization to improve compatibility and information sharing.
- Maintaining operations by monitoring errors and stoppage messages, observing peripheral equipment entering a command to correct errors.
- Installation and configuration of network devices, printers, computer programs, and operating systems.

EDUCATION

BSc, Computer Science | ESGT Benin University | Cotonou, Benin **2018 – 2022**

Honor Diploma, Software Engineering | National Institute of Information Technology | Lagos, Nigeria **2011 – 2014**

CERTIFICATE

Software Architecture

PROFILES

Personal Website: https://semslam.github.io/my-portfolio/

Linked In: https://www.linkedin.com/in/ibrahim-olanrewaju-2b8473103/

GitHub: https://github.com/semslam