**Executive Project Summary**

This document outlines the project's journey and progression, detailing key milestones, feature developments, and significant fixes. The project began with a fundamental task of storing JSON documents from a Kafka topic into an Oracle database and has evolved into a robust, full-stack application with enhanced features, security, and a modern user interface.

**Project Journey and Progression**

**Phase 1: Foundational Development (August 13-16, 2025)**

The initial phase focused on building the core data ingestion and storage capabilities.

* **Initial Setup**: The project was set up as a Java application to store JSON documents in an Oracle database. It was then converted to a Spring Boot application using Maven, with a

JdbcTemplate for database operations.

* **Kafka Integration**: A Kafka consumer was integrated to listen to a specified topic and store received JSON messages in an Oracle database table named json\_docs.
* **Database Migration & JPA**: The application was migrated from JdbcTemplate to Spring Data JPA. This included the creation of new entities for

JsonDoc, TradeDetail, TradeException, and AppUser.

* **Enhanced Data Handling**: Logic was added to extract specific trade details from incoming JSON messages and store them in a new trade\_details table. The

KafkaJsonListener was also refactored to save messages to the database first, with subsequent processing handled asynchronously.

**Phase 2: UI, APIs, and Core Enhancements (August 17-21, 2025)**

This phase introduced a user-facing interface and critical backend functionalities.

* **API and UI for Data Viewing**: A full-stack solution was implemented to view stored JSON data. This included a new REST API endpoint (

GET /api/data) with date range filtering and a React UI to display the results.

* **Feature Screens**: New screens were added to the frontend:
  + **Trade Inquiry**: A screen was created to allow users to search for trade details by client reference number.
  + **Trade Exception Inquiry**: A screen and its corresponding backend validation were introduced to track trades that failed validation rules (e.g., trade date, settlement date, principal amount).
  + **Fund Master**: A new screen was created for performing CRUD (Create, Read, Update, Delete) operations on fund details.
* **Authentication & Security**: Backend infrastructure for user authentication and data entitlements was established.

spring-boot-starter-security was added, and data was filtered based on the authenticated user's fund entitlements. A

/health endpoint was also introduced for monitoring application status.

* **Database & CORS Fixes**: The database was migrated to MySQL, and the application was configured to use Spring profiles for a better development experience with an in-memory H2 database for testing. Persistent Cross-Origin Resource Sharing (CORS) issues were resolved by centralizing the configuration within Spring Security.

**Phase 3: Refinement and Automation (August 22-24, 2025)**

The final phase focused on improving the user experience, hardening the application, and adding new features.

* **UI/UX Improvements**: The frontend was completely overhauled with Material-UI for a modern look. Features like a persistent left navigation drawer and a grouped bar chart for trade statistics were added.
* **Kafka Management**: A new REST endpoint was created to allow authorized users to publish messages directly to the inbound Kafka topic for testing purposes. An API endpoint was also introduced to browse messages from a Kafka topic without affecting the consumer offset.
* **Enhanced Functionality**:
  + Excel export functionality was added to the Trade Inquiry screen.
  + The search logic was refactored to support searching by either client reference number OR date range.
  + A unique ID generator was implemented for robust tracking in a multi-server environment.
* **Bug Fixes**: Critical bugs were addressed, including:
  + Backend authentication and header validation issues.
  + A bug that caused fund creation to fail due to a

PUT request being sent instead of POST.

* + Correcting the date format mismatch for trade date filtering.