Name – Aditya Sharma

Project Name – Data collection Backend Assignment (Atlan)

Project Assigned Date – 22/11/2023.

Project Completion Date – 26/11/2023

Data Collection Platform - Design Specification

Overview

The Data Collection Platform is designed to provide a robust and flexible solution for managing forms, collecting responses, and integrating with external services. This document outlines the architectural design, key components, and features of the platform.

Architecture

The platform follows a three-tier architecture:

Presentation Tier (Frontend):

User interfaces for form creation, response submission, and integration management.

Utilizes HTML, CSS, and JavaScript for web-based interfaces.

Responsive design for optimal user experience across devices.

Application Tier (Backend):

Implemented using Node.js and Express for the server.

MongoDB as the primary database for storing forms and responses.

Modular controllers for handling form, response, and integration logic.

Integration with external services like Google Sheets and SMS gateways.

Data Tier (Database):

MongoDB for storing forms and responses.

Collections: forms, questions, responses, integrations.

Indexing for efficient querying.

Ensure eventual consistency for data integrity.

Components

1. Form Management

- Forms:

  Stored in the forms collection.

Each form has metadata (e.g., title, description) and an array of questions.

Questions stored in the questions collection.

2. Response Management

- Responses:

Stored in the responses collection.

Each response is associated with a form and contains answer data.

Validation against business rules using the validation Service.

Integration with external services upon response submission.

3. Integration

- Google Sheets Integration:

Utilizes the google Sheets Service to update a designated Google Sheets spreadsheet.

Spreadsheet ID and API key configured in the .env file.

Maps response data to rows in the sheet.

- SMS Integration:

Utilizes the sms Service to send SMS receipts.

Integration with an SMS gateway service.

Configured in the .env file with service-specific credentials.

Eventual Consistency

Ensure eventual consistency across the platform.

Leverage MongoDB's replication features for data redundancy.

Implement retry mechanisms for failed integration attempts.

Failsafe and Recovery

Implement mechanisms for recovering from power, internet, or service outages.

Use logging to record errors and failures for debugging and analysis.

Scalability

Design to handle millions of responses across hundreds of forms.

Optimize database queries and indexing.

Implement load balancing and horizontal scaling if needed.

Security

Protect sensitive information (API keys, service credentials) using environment variables.

Implement user authentication and authorization for form and response management.

Regular security audits and updates.

Monitoring and Logging

Utilize logging for system health monitoring.

Set up alerts for critical issues or failures.

Monitor database performance and resource utilization.

Testing

Implement unit tests for controllers and services.

Integration tests for end-to-end testing of form creation, response submission, and integrations.

Conclusion

This design specification provides a detailed overview of the Data Collection Platform, covering its architecture, key components, features, and considerations for scalability, security, and monitoring. Follow the outlined design principles to ensure a robust and reliable platform for data collection and integration.