Report Submission App - Proposal & Sprint Plan

1. Overview

The Report Submission App is a reporting and submission platform designed for field assessments, built with an offline-first architecture to ensure uninterrupted data collection in remote environments. It enables field users to record inspection data and sync it seamlessly once connectivity is available. The system includes a Super Admin Panel, Role-Based Access Control (RBAC), response dashboards, and offline data synchronization capabilities.

This document outlines the scope of work, sprint-wise development plan, technology stack, timelines & estimated hours.

2. Project Scope

Flutter Mobile App (Android)

- Authentication & Access Control ○ Login using RBAC (Super Admin, Admin, Inspector) ○ Secure JWT-based authentication ○ Role-based dashboards
- Profile Management ○ View and update profile details ○ Role-specific profile access (Inspector/Admin)
- Form Management ○ View assigned forms ○ Fill and submit forms (160+ dynamic data points) ○ Auto-save and sync when online ○ View previously filled and submitted forms ○ Image input with compression ○ Local DB storage (SQLite/Hive)
- Offline Functionality ○ Outbox pattern for data sync ○ Retry logic for failed uploads ○ Background sync service *Super Admin Web Panel (ReactJS)*
- Authentication & Roles ○ Login & session management via JWT ○ Role management: Super Admin, Admin, Inspector
    - Permissions configuration per role
- User Management
    - Create, edit, or deactivate users ○ Assign roles and manage credentials
- Form Builder & Template Management ○ Create dynamic templates with multiple field types (text, dropdown, checkbox, images)

    - Save, reuse, and edit templates ○ Assign forms to inspectors or teams
- Data Review & Analytics ○ View all submitted forms ○ Filter by user, department, or form type ○ Compare submissions over time ○ Export responses (CSV / JSON)
    - Simple analytics dashboard with completion rates *Backend (NodeJS + Express + TypeScript)*
- Authentication & Authorization ○ Secure JWT-based login ○ RBAC middleware for permission-based routing
- Data Management APIs ○ Form creation, submission, update, and retrieval endpoints ○ Role-based user CRUD APIs
- Offline Sync Logic ○ Outbox pattern implementation ○ Queue-based retry logic for uploads ○ Background sync status APIs
- Logging & Monitoring ○ API event logs
    - Error tracking & request monitoring *Database (PostgreSQL)*
- Centralized relational schema for users, templates, and submissions
- Indexed tables for faster queries
- Optimized joins for dashboard analytics

Azure Infrastructure Setup

- Azure App Service – Hosting for Node.js backend
- Azure Blob Storage (S3 equivalent) – For storing form attachments, images
- Azure PostgreSQL Server – Managed relational database
- Azure Application Insights – Real-time monitoring and performance tracking