Hackathon Submission Report

Clinical Data Monitoring System (CDMS)

Hackathon Submission Report - Clinical Data Monitoring System (CDMS)

Project Overview

The Clinical Data Monitoring System (CDMS) is a state-of-the-art, AI-powered platform designed to revolutionize clinical trial monitoring. It provides Clinical Research Associates (CRAs) and Data Quality Teams (DQT) with real-time insights, risk-based monitoring capabilities, and automated reporting to ensure patient safety and data integrity.

Key Features

1. Unified Monitoring Dashboard

- Real-time KPIs: Instant visibility into Total Sites, Total Subjects, High-Risk Sites, and Clean Patient percentages.
- Visual Analytics: Interactive risk distribution charts (Pie/Bar) and regional performance analysis using Recharts.
- Premium UI: Implemented with a "Neon Tech" glassmorphism aesthetic for high-impact visual clarity.

2. High-Precision Risk Analysis

- Site-Level Risk: Advanced scoring algorithm that identifies sites requiring immediate attention based on missing pages, open issues, and uncoded terms.
- Patient-Level Deep Dive: Granular monitoring of individual patient data quality indices (DQI).
- Interactive Data Grid: Responsive tables with hover effects and detailed drill-downs.

3. AI Assistant & Automated Insights

- GPT-Powered Intelligence: Natural language interface for querying complex clinical datasets (OpenAI GPT-4/GPT-5 series).
- Smart Recommendations: AI-driven action plans for resolving site-level risks.
- Dynamic Reporting: One-click generation of Site Performance, CRA, and Risk Analysis reports.

4. Enterprise Collaboration & Reporting

- Cloud-Powered Email Integration: Ability to send generated reports directly to stakeholders via Amazon SES (Simple Email Service). This provides a professional, scalable, and highly deliverable email solution for trial-ready communications.
- Comprehensive Exports: PDF and CSV export capabilities for all major data views.
- Issue Tracking: Integrated alerting system for creating, assigning, and resolving clinical monitoring issues.

Technical Architecture

Frontend

- Framework: React 19 (Modern, optimized rendering)
- Styling: Vanilla CSS + Tailwind CSS (Custom "Neon Tech" design system)
- UI Components: Shadcn UI + Framer Motion (Fluid animations)
- State Management: React Hooks (Efficient and predictable)

Backend

- Framework: FastAPI (High-performance Python API)
- Cloud-Scale Data Infrastructure:

- Mor
availa

- Supabase: High-performance backend-as-a-service leveraging PostgreSQL for powering the clinical trial datasets with real-time capabilities.

- Communication: RESTful API with JWT & Firebase Hybrid Authentication.
- Cloud Notification Engine: Amazon SES integrated via Boto3 for reliable, authenticated email delivery (SPF/DKIM/DMARC compliant).

Intelligence Layer

- Large Language Model: Seamless OpenAI integration for sophisticated clinical reasoning, data synthesis, and automated report generation.

Design Philosophy

The CDMS uses a Premium Glassmorphism design language. We prioritized "Aesthetics with Purpose," ensuring that the futuristic neon-dark theme doesn't just look impressive but actively aids in data scanning and critical insight identification.

Folder Structure Summary

- `/frontend`: The core React application, including custom UI components and page logic.
- `/backend`: The FastAPI server, database adapters, and AI integration logic.
- `DEPLOYMENT.md`: Step-by-step guide for production hosting.
- `SETUP_INSTRUCTIONS.md`: Comprehensive local environment configuration.
- `WINDOWS_SETUP.md`: OS-specific setup for development.

Conclusion

This submission represents a complete, scalable, and production-ready solution for modern clinical trials. By combining cutting-edge AI with a user-centric interface and robust cloud infrastructure (AWS, Supabase, MongoDB Atlas), the Clinical Data Monitoring System sets a new standard for data quality and trial efficiency.

> For detailed data cleaning, preprocessing steps, and in-depth technical interpretations of the results, kindly refer to the accompanying `.ipynb` notebook file included in the submission package.