Groundwater ML Platform - Project Report

1. Executive Summary

The Groundwater ML Platform is a comprehensive web application designed to monitor, analyze, and forecast groundwater quality across districts in India. It combines historical data visualization with machine learning-driven insights to help policymakers and researchers identify pollution hotspots and trends.

2. Key Features Implemented

2.1 Pollution Density Heatmap

Visualizes the geographic density of water pollution using a heatmap overlay.

Implementation Details:

- Library: Integrated 'leaflet.heat' plugin with React-Leaflet.
- Logic: Data points are weighted by their WQI (Water Quality Index). Higher WQI (>100) results in 'hot' (red) intensity areas.
- UI: Added a toggle switch in 'MapPage' to dynamically switch between Marker Cluster view and Heatmap layer.

2.2 Time Lapse Animation

An animated playback feature that visualizes how groundwater quality has evolved over the years (2019-2023).

Implementation Details:

- State Management: Uses 'isPlaying' React state and 'setInterval' to update the 'selectedYear' automatically.
- Controls: Play/Pause button added to the map interface.
- Data: Cycles through the available years dataset, triggering a map data refresh for each year step.

2.3 Interactive Risk Filters

Allows users to filter map data by clicking on summary cards (e.g., 'Excellent', 'Poor', 'Unsuitable').

Implementation Details:

- State: 'filterType' state tracks the currently selected risk category.
- Filtering: The map data array is filtered in real-time on the client side based on the selected category.
- UX: Active cards are highlighted with a border matching the category color.

3. Technical Architecture & Deployment

3.1 Monorepo Structure (Next.js + FastAPI)

- Frontend: Next.js 15 (App Router), deployed as Vercel Frontend.

- Backend: FastAPI (Python), deployed as Vercel Serverless Functions in '/api'.
- Configuration: 'vercel.json' manages the routing and build commands for both stacks.

3.2 Vercel Optimization

- Size Limit Resolution: Removed heavy ML libraries (scikit-learn, joblib) from the backend to fit within the 250MB Serverless Function limit.
- File Handling: Implemented absolute path resolution ('os.path.abspath') in Python to correctly locate CSV data files in the serverless environment.
- Upload Optimization: Configured '.vercelignore' to exclude local build artifacts ('.next', 'node_modules'), reducing upload size from 1.1GB to ~150MB.

4. Future Roadmap

- Re-integrate full ML models using ONNX Runtime for client-side or lightweight server-side inference.
- Migrate from CSV data storage to a cloud database (PostgreSQL/Supabase) for real-time data updates.
- Add more granular filtering for chemical parameters (Arsenic, Nitrate, etc.).