Water Quality Monitor

Project Statement:
Enables real-time water quality monitoring for communities and authorities, providing open data dashboards, user reports, and alerts for contamination.

Key Features:

- Real-time monitoring via government APIs
- User-submitted pollution reports with images
- Alerts for contamination and boil advisories
- Historical trends for analysis
- NGO & authority dashboards
- Predictive alerts using analytics

Tech Stack:

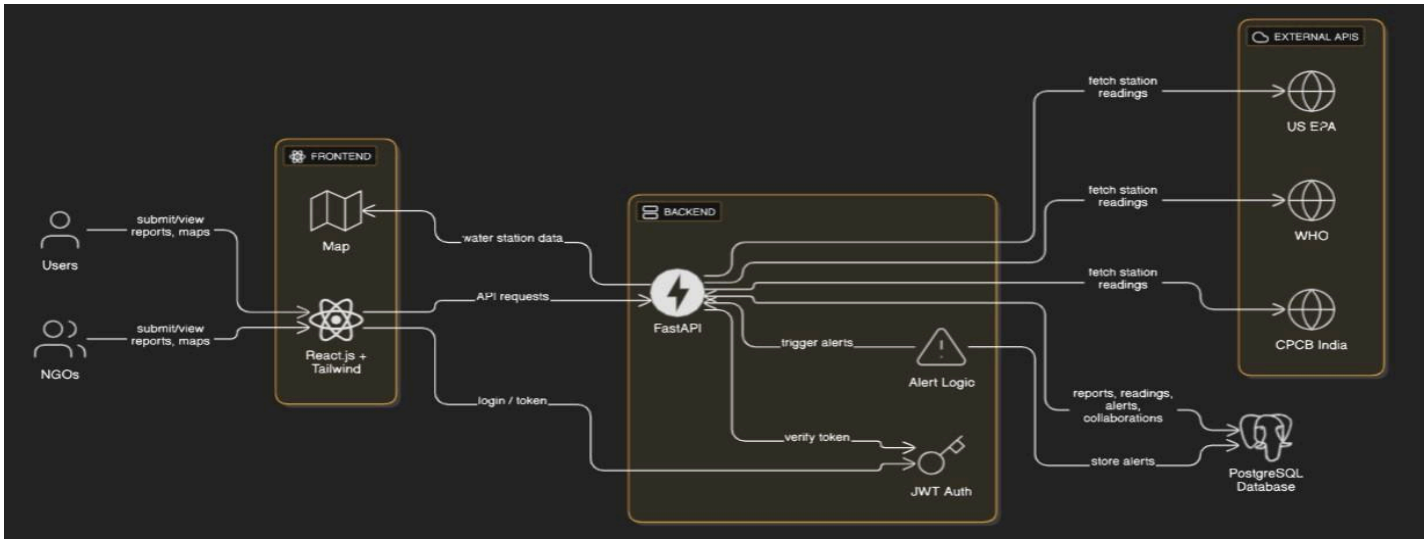
- **Frontend:** React.js + Tailwind CSS
- **Backend:** FastAPI
- **Database:** PostgreSQL
- **Auth:** JWT

Modules:

- Module A: User & Report Management
- Module B: Real-Time Water Data & Maps
- Module C: Alerts & Collaboration Tools

- Module D: Analytics & Predictive Insights

Architecture Diagram:



8-Week Milestone Plan

Milestone 1: Weeks 1–2 – Setup & Auth

- Week 1: Set up backend & database schema
- Week 2: Build login/register and map-based UI skeleton

Expected Output: Authenticated dashboard and base map view

Milestone 2: Weeks 3–4 – Reporting & Station Data

- Week 3: Implement the user reporting system
- Week 4: Display real-time station readings

Expected Output: Reports and water data feed integrated

Milestone 3: Weeks 5–6 – Alerts & History

- Week 5: Add alerts module (boil notices, contamination alerts)
- Week 6: Implement historical data graphs

Expected Output: Alerts and historical trends

Milestone 4: Weeks 7–8 – NGO Dashboard & Deployment

- Week 7: Add NGO collaboration tools
- Week 8: Predictive alerts, QA, deployment

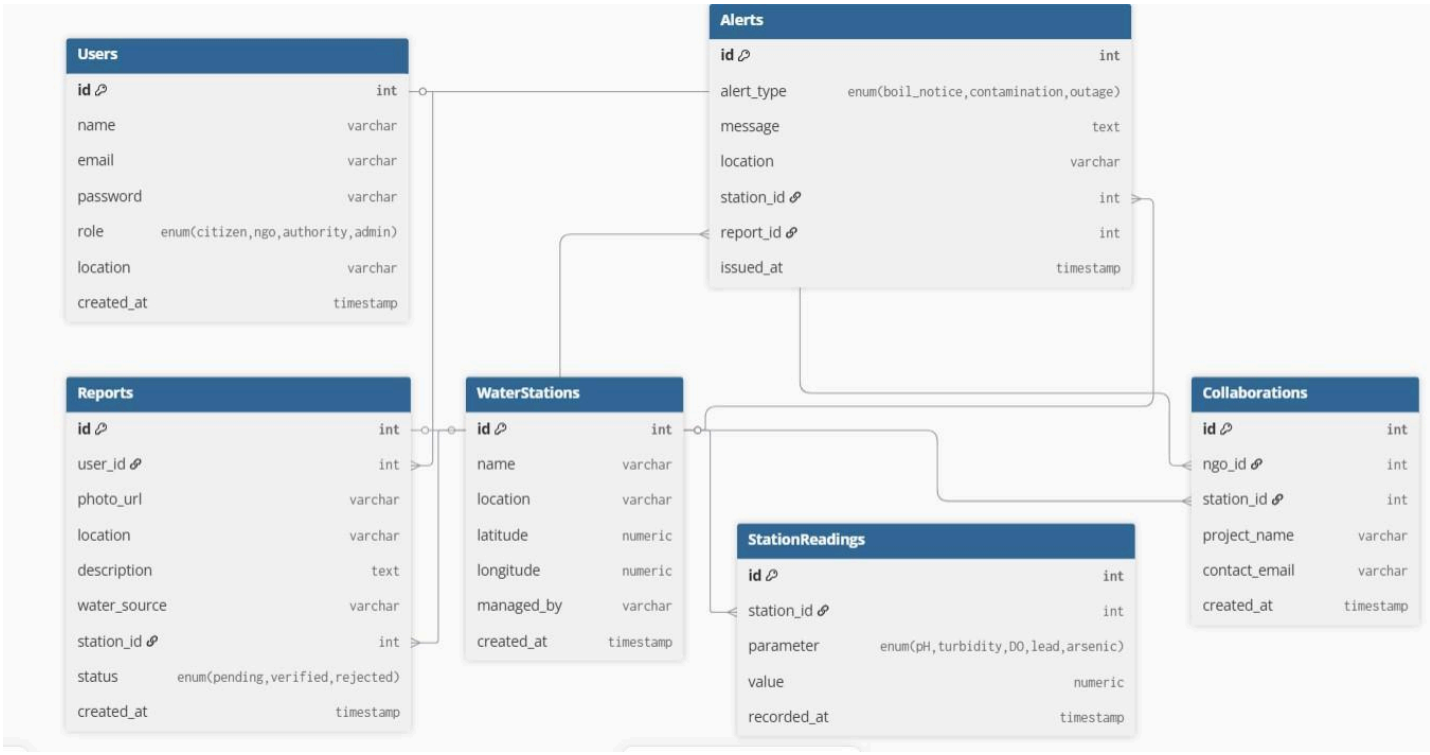
Expected Output: Complete water monitoring platform

Expected Project Outcome:

WaterWatch will provide a collaborative, real-time water safety dashboard that engages users and authorities.

Database Schema:

- **Users:** id (INT, PK), name (VARCHAR), email (VARCHAR, UNIQUE), password (VARCHAR), role (ENUM: 'citizen','ngo',' authority', 'admin'), location (VARCHAR), created_at (TIMESTAMP)
- **Reports:** id (INT, PK), user_id (FK to Users.id), photo_url (VARCHAR), location (VARCHAR), description (TEXT), water_source (VARCHAR), status (ENUM: 'pending',' verified', 'rejected'), created_at (TIMESTAMP)
- **WaterStations:** id (INT, PK), name (VARCHAR), location (VARCHAR), latitude (NUMERIC), longitude (NUMERIC), managed_by (VARCHAR), created_at (TIMESTAMP)
- **StationReadings:** id (INT, PK), station_id (FK to WaterStations.id), parameter (ENUM: 'pH', 'turbidity', 'DO', 'lead', 'arsenic'), value (NUMERIC), recorded_at (TIMESTAMP)
- **Alerts:** id (INT, PK), type (ENUM: 'boil_notice','contamination',' outage'), message (TEXT), location (VARCHAR), issued_at (TIMESTAMP)
- **Collaborations:** id (INT, PK), ngo_name (VARCHAR), project_name (VARCHAR), contact_email (VARCHAR), created_at (TIMESTAMP)


Users
id int
name varchar
email varchar
password varchar
role enum(citizen,ngo,authority,admin)
location varchar
created_at timestamp
Alerts
id int
alert_type enum(boil_notice,contamination,outage)
message text
location varchar
station_id int
report_id int
issued_at timestamp
Reports
id int
user_id int
photo_url varchar
location varchar
description text
water_source varchar
station_id int
status enum(pending,verified,rejected)
created_at timestamp
WaterStations
id int
name varchar
location varchar
latitude numeric
longitude numeric
managed_by varchar
created_at timestamp
Collaborations
id int
ngo_id int
station_id int
project_name varchar
contact_email varchar
created_at timestamp
StationReadings
id int
station_id int
parameter enum(pH,turbidity,DO,lead,arsenic)
value numeric
recorded_at timestamp