Fuel Price Tracker + Alert System: Full Project Layout

1. Project Overview

Fuel Price Tracker + Alert System is a web-based application designed to help users monitor live fuel prices across Israel. Users can view prices on a map, filter stations, mark favorites, and receive alerts when prices drop below a specified threshold.

2. Technology Stack

- Frontend: React.js
- Mapping: Leaflet.js or Google Maps API
- Backend: Node.js with Express
- Scraper/API: Puppeteer or Cheerio for live data collection (or mock data)
- Database: MongoDB (or PostgreSQL)
- Notifications: Email (via Nodemailer), Push Notifications (via OneSignal)

3. Phase 1: UI + Mock Data

- Design the frontend layout using React
- Create a map component with Leaflet.js
- Display mocked fuel station data (name, price, company, address)
- Implement station filtering by price/company
- Add user favorites functionality (saved in local storage)

4. Phase 2: Live Fuel Data Integration

- Write a scraper using Puppeteer to extract prices from fuel company websites
- Alternatively, simulate APIs if scraping is not feasible
- Normalize data: station name, location (lat/lng), price, timestamp
- Store fetched data in MongoDB with update timestamps

5. Phase 3: User Accounts + Database

- Create signup/login system (JWT auth)

- Store user data: email, preferences, favorite stations, price thresholds
- Connect users to their saved data using MongoDB collections

6. Phase 4: Alerts System

- Allow users to set price thresholds per station or area
- Run a background job (cron) to check updated prices
- If a station meets criteria, send alert via email or push
- Use Nodemailer for email, OneSignal API for push notifications

7. Phase 5: Deployment

- Frontend: Deploy to Vercel
- Backend: Deploy to Render or Railway
- Database: Host MongoDB on MongoDB Atlas
- Set up background jobs with services like cron-job.org or built-in Render scheduler

8. Bonus Features

- Add fuel price history (charts)
- Add dark/light mode
- Localization (Hebrew, Arabic, English)
- Admin dashboard for data monitoring
- Mobile-first responsive design

9. Final Notes

- Prioritize clean UI/UX to make the app feel professional
- Include GitHub repo link and live demo in your resume
- Focus on stability and usability for real-world relevance