Fuel Price Tracker - Final Integration & Pro Guide

1. Welcome to the Final Phase

This guide explains how to bring together everything you?ve built across Phases 1?5 and polish the project like a professional. It includes integration tips, common bugs to watch for, optimization ideas, and strategic advice to level up your development process.

2. Combining All Phases (Workflow Summary)

- Frontend loads the map and calls the backend for station data
- Backend fetches data from MongoDB (populated by your scraper)
- Logged-in users access saved preferences from backend APIs
- Scheduler updates station prices and sends alerts in background
- Frontend displays everything dynamically via API calls

3. Key Integration Points

- Connect frontend filters to query backend `/stations` API
- When scraper runs, it should **update**, not duplicate, station data
- Alert logic reads both `station.price` and `user.threshold`
- Use loading spinners or skeleton UIs to improve experience during fetches

4. Common Pitfalls (and Fixes)

- **Scraper breaks**: Site structure changed ? inspect elements again and update selectors
- **Data mismatch**: Normalize city names (e.g., 'Tel Aviv-Yafo' vs 'Tel Aviv')
- **JWT errors**: Double-check token storage and expiration handling
- **No alerts sent**: Ensure cron is running, threshold logic is correct, email account isn't blocked

5. Optimization Ideas

- Add caching (e.g., Redis) to reduce DB calls
- Store fuel price history for trends/graphs
- Show 'last updated' timestamps in UI
- Add map clustering for better UX at zoomed-out levels

Fuel Price Tracker - Final Integration & Pro Guide

6. Professional Touches

- Mobile-first design using media queries or Tailwind
- Include a ?last known good? data backup system
- Implement role-based access (admin vs. user)
- Add user feedback system: report wrong prices or suggest station

7. Preparing for Job Applications

- Clean, commented codebase on GitHub
- Live working demo (no broken links or secrets)
- Include screenshots, walkthrough video, or README badges
- Highlight tech stack + what you built in your resume

8. Final Thoughts

This project shows frontend skill, backend logic, automation, data scraping, and deployment ? it?s a full-stack, real-world solution. If you ever add AI price prediction or fleet company dashboards, it could turn into a full SaaS platform. Stay curious, break things, fix them, and ship fast!