Fuel Price Tracker - Phase 1: UI + Mock Data (Beginner-Pro Guide)

1. Introduction

Phase 1 is about building the frontend interface of your Fuel Price Tracker project using mock (fake) data. This lets you design the look, feel, and flow of your app before connecting it to real data in later phases. Even with mock data, your app should feel complete and responsive, with all the features in place.

2. Goals for Phase 1

- Display a map with fuel stations plotted
- Show fuel station info (name, price, company, city)
- Enable filtering (by company, city, or price)
- Allow users to favorite stations
- Use mocked data to simulate live prices

3. Tools You'll Use

- **React**: JavaScript framework for building UI
- **Leaflet.js**: Lightweight map library
- **react-leaflet**: React bindings for Leaflet
- **Bootstrap or Tailwind** (optional): for styling
- **useState / useEffect**: React hooks for managing state and lifecycle

4. Project Setup

1. Create a new React app:

`npx create-react-app fuel-tracker`

2. Install map tools:

`npm install leaflet react-leaflet`

3. Inside `public/index.html`, add Leaflet CSS:

<link rel="stylesheet" href="https://unpkg.com/leaflet@1.9.3/dist/leaflet.css" />

5. Step-by-Step: Display a Map

1. Inside `App.js`, import components:

Fuel Price Tracker - Phase 1: UI + Mock Data (Beginner-Pro Guide)

```js
import { MapContainer, TileLayer, Marker, Popup } from 'react-leaflet';
import 'leaflet/dist/leaflet.css';
```

2. Add basic map:

```js
<MapContainer center={[32.08, 34.78]} zoom={8} style={{ height: '90vh' }}>
  <TileLayer url="https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png" />
</MapContainer>
```

6. Create Mock Fuel Data

In a new file `mockData.js`:

```js
const stations = [
  { id: 1, name: 'Yellow Haifa', company: 'Yellow', city: 'Haifa', price: 7.65, lat: 32.8, lng: 35.0 },
  { id: 2, name: 'Sonol Tel Aviv', company: 'Sonol', city: 'Tel Aviv', price: 7.75, lat: 32.07, lng: 34.78 }
];
export default stations;
```

7. Plot Stations on the Map

In `App.js`, import mock data and add Markers:

```js
import stations from './mockData';
{stations.map(station => (
  <Marker key={station.id} position={[station.lat, station.lng]}>
    <Popup>
      <b>{station.name}</b><br />{station.city}<br />?{station.price}
    </Popup>
```

~~~
  </Marker>
))}
```
~~~

8. Add Filtering UI

Create dropdowns or sliders to filter stations by company, price, or city.

Use `useState()` to store the selected filter and display only stations that match.

9. Favorite Stations

- Add a heart icon or star next to each station
- Use `useState()` or `localStorage` to track favorites
- Highlight favorites on the map (e.g., custom Marker color)

10. Final Touches

- Make sure your app works well on mobile
- Add basic styles or a theme
- Include loading/error states, even for mock data
- Structure folders: `/components`, `/data`, `/utils`, `/styles`