Take Home Task Fullstack

Develop a Fullstack application for viewing and scheduling trucks. This should include both a backend and a frontend.

Tools and Technologies:

- **Frontend**: React (for SPA) or Next.js/Remix (for server-rendered)
- **Backend Language**: TypeScript with free choice of Node, Bun or Deno
- **Database**: PostgreSQL

Tasks:

- **Display Construction Sites and Trucks**:
  - Frontend should list all trucks and construction sites, including their names and coordinates.
  - [Optional] Display both the trucks and construction sites on a map.
- **Scheduling Interface**:
  - Users can select a construction site from a list or map view.
  - Users can specify a starting time within the next 4 hours for scheduling trucks for picking up material from the site.
  - Users can enter the number of trucks needed and at what cadence they should arrive at the construction site.
    - Example: User selects 8 trucks and a 10 minute cadence.
  - The backend calculates which trucks could get there within that time and offers an option, trying to minimize distances driven.
  - Users see feedback about which trucks would get there at what time given the schedule.
  - Submitting the proposed schedule, saves the scheduling information in the backend

Database Schema and Seed Data:

Schema:

```
CREATE TABLE trucks (
  id SERIAL PRIMARY KEY,
  latitude DECIMAL(10, 8) NOT NULL,
  longitude DECIMAL(11, 8) NOT NULL,
  model VARCHAR(100),
  make VARCHAR(100),
  year INTEGER,
  capacity INTEGER,
  status VARCHAR(50)
);
CREATE TABLE construction_sites (
  id SERIAL PRIMARY KEY,
  latitude DECIMAL(10, 8) NOT NULL,
  longitude DECIMAL(11, 8) NOT NULL,
  name VARCHAR(100),
);
CREATE TABLE scheduling (
  -- To be determined by the implementing developer
);
```

Seed data:

```
INSERT INTO trucks (latitude, longitude, model, make, year, capacity, status)
SELECT
  53.187178 + (RANDOM() * 0.36 - 0.18) AS latitude, -- Adjusting for a variance of about 20km
  9.753796 + (RANDOM() * 0.56 - 0.28) AS longitude, -- Adjusting for a variance of about 20km
```

```
  'Model ' || (RANDOM() * 100)::INT::VARCHAR AS model,
  'Make ' || (RANDOM() * 100)::INT::VARCHAR AS make,
  2000 + (RANDOM() * 24)::INT AS year, -- Random year between 2000 and 2024
  (RANDOM() * 20000 + 6000)::INT AS capacity, -- Random capacity between 6 and 24tons
  CASE WHEN RANDOM() < 0.5 THEN 'Available' ELSE 'Unavailable' END AS status -- Randomly assigning status
FROM
  generate_series(1, 250) -- Generates 250 rows;

INSERT INTO construction_sites (latitude, longitude, name)
SELECT
  53.187178 + (RANDOM() * 0.36 - 0.18) AS latitude, -- Adjusting for a variance of about 20km
  9.753796 + (RANDOM() * 0.56 - 0.28) AS longitude, -- Adjusting for a variance of about 20km
  'Site ' || (RANDOM() * 1000000)::INT::VARCHAR AS name -- Generating a unique name for each site
FROM
  generate_series(1, 15) -- Generates 15 rows;
```

Documentation and Submission:

- **Documentation**:
  - Include instructions for setting up and running the application, both backend and frontend.
- **Submission**:
  - Please push the code and readme into the GitHub repository that this task was shared in with you.