Task: Advanced Event Management System

Overview

Build a web application for managing events. The application should allow users to:

- View a list of events.
- Create, edit, and delete events.
- Filter events by date, category, or location.
- Manage user roles (e.g., admin and regular users).

Custom Database Schema

Schema Design:

```
-- Users Table
CREATE TABLE users (
    id SERIAL PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    email VARCHAR(100) UNIQUE NOT NULL,
    password_hash VARCHAR(255) NOT NULL,
    role ENUM('admin', 'user') DEFAULT 'user' NOT NULL,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

-- Locations Table
CREATE TABLE locations (
    id SERIAL PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    address TEXT NOT NULL,
    city VARCHAR(50) NOT NULL,
    state VARCHAR(50) NOT NULL,
    country VARCHAR(50) NOT NULL,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

-- Events Table
CREATE TABLE events (
    id SERIAL PRIMARY KEY,
    title VARCHAR(100) NOT NULL,
    description TEXT NOT NULL,
    date DATE NOT NULL,
    category VARCHAR(50) NOT NULL,
    location_id INT NOT NULL,
    created_by INT NOT NULL,
    FOREIGN KEY (location_id) REFERENCES locations(id),
    FOREIGN KEY (created_by) REFERENCES users(id),
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

-- Event Registrations Table
CREATE TABLE event_registrations (
    id SERIAL PRIMARY KEY,
    user_id INT NOT NULL,
    event_id INT NOT NULL,
    registration_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    status ENUM('registered', 'cancelled') DEFAULT 'registered' NOT NULL,
    FOREIGN KEY (user_id) REFERENCES users(id),
    FOREIGN KEY (event_id) REFERENCES events(id)
);
```

Sample Data

Users:

```
INSERT INTO users (name, email, password_hash, role) VALUES
('Alice Admin', 'alice.admin@example.com', 'hashed_password_1', 'admin'),
('Bob User', 'bob.user@example.com', 'hashed_password_2', 'user');
```

Locations:

```
INSERT INTO locations (name, address, city, state, country) VALUES
('Tech Park', '123 Main St', 'San Francisco', 'CA', 'USA'),
('Convention Center', '456 Elm St', 'New York', 'NY', 'USA');
```

Events:

```
INSERT INTO events (title, description, date, category, location_id,
created_by) VALUES
('React Workshop', 'Learn React basics', '2025-02-01', 'Workshop', 1, 1),
('SQL Mastery', 'Advanced SQL techniques', '2025-02-15', 'Seminar', 2, 1);
```

Event Registrations:

```
INSERT INTO event_registrations (user_id, event_id) VALUES
(2, 1),
(2, 2);
```

Requirements

1. **Backend**
   - Build an Express.js server with the following endpoints:
     - **Authentication:**
       - `POST /auth/register`: Register a new user.
       - `POST /auth/login`: Authenticate and return a JWT.
     - **Events Management:**
       - `GET /events`: Fetch all events with optional filters (`date`, `category`, `location`).
       - `POST /events`: Create a new event (admin-only).
       - `PUT /events/:id`: Update an event by ID (admin-only).
       - `DELETE /events/:id`: Delete an event by ID (admin-only).
     - **Event Registration:**
       - `POST /events/:id/register`: Register for an event.
       - `GET /events/registrations`: List all registrations for the logged-in user.
   - Use any popular ORM / Query Builder for database operations.

2. **Frontend**
    - Use React to build the UI.
    - Pages:
        - **Home**: Display a list of events with filters (date, category, location).
        - **Event Details**: Show event details and allow users to register.
        - **Admin Dashboard**: Allow admins to manage events and view registrations.
        - **Login/Register**: Authentication pages.
    - Include client-side form validation.
3. **Custom Requirements**
    - Use Context API for state management (do not use Redux).
    - Add role-based access control (admin vs. regular user).
    - Implement pagination on the events list (5 events per page).
    - Use a library like `date-fns` for date formatting.
    - Include a modal for confirming event deletions.
4. **Code Submission**
    - Push the code to a GitHub repository with a clear README file.
    - Include setup instructions for running the project locally.