Rooming List Management App - Use Cases

1. Authentication (JWT)

- Log in to obtain a JWT token.

- Protect endpoints using JWT authentication (middleware).

- Verify token validity for each secure API request.

2. Rooming List Management

- Retrieve all Rooming Lists from the database.

- Manually insert a Rooming List (admin only).

- Delete all Rooming Lists (used before JSON insertion).

- Filter Rooming Lists by:

- Event ID (eventId)

- RFP Name (text)

- Agreement Type

- Status (Active, Closed, Cancelled)

- Sort Rooming Lists by cutOffDate (ascending or descending).

3. Booking Management

- Get all bookings for a specific Rooming List.

- Insert individual bookings (optional).

- Delete all bookings (before new JSON insert).

4. JSON Data Import

- Frontend button triggers the following:

- Delete all Rooming Lists, Bookings, and Relations.

- Parse 3 JSON files:

- - rooming-lists.json

- - bookings.json

- - rooming-list-bookings.json

- Insert parsed data into related tables with proper associations.

5. Testing

- Write at least 2 unit tests for the backend (e.g., Rooming List or Bookings endpoints).

- Use Jest and Supertest to simulate API calls.

6. Frontend: Rooming List Visualization

- Display RoomingListCard for each Rooming List.

- Show: rfpName, agreement_type (bold), cutOffDate.

- Calculate checkInDate (earliest) and checkOutDate (latest) from bookings.

- Group Rooming Lists by eventId with horizontal scroll if more than 3.

- Apply consistent styling (colors, spacing, scroll, buttons).

7. Filtering & Search

- Search bar:

- Search by event name, rfpName, or agreement_type.

- Use debounce to avoid excessive API calls.

- Filter panel:

- Toggle visibility of Rooming Lists by status (Active, Closed, Cancelled).

8. Rooming List Card Interaction

- View Bookings button:

- On click, make an API call.

- Display associated bookings in browser console.

- View Agreement button with tooltip 'Show Agreement as PDF' (placeholder).

9. Responsive Design

- Ensure proper rendering on:

- Desktop

- Tablet

- Mobile Devices

10. Optimization & Performance

- Use Zustand / Context / Redux Toolkit for global state.

- Avoid unnecessary re-renders using memo or useMemo.

11. Dockerization

- Create Dockerfile for backend.

- Create docker-compose.yml to launch:

- PostgreSQL database

- Backend and Frontend containers

- Configure environment variables using .env file.

Bonus Features

- Pagination or infinite scroll for Rooming Lists.

- Add icons and micro-interactions (as shown in the mockup).

- Show visual confirmation message after importing JSON data.

Project Modules Structure

- Organize code in modules/epics like:

- auth/

- rooming-lists/

- bookings/

- import/

- filters/

- ui/