

# Semester Project Submission 2: Airbnb-inspired - Backend Setup, Static Listings, and Page Creation

## Objective

Enhance the project by integrating a Node.js and Express.js backend, creating detailed pages for listings and booking, and ensuring data is served through routes. This will provide a richer user experience by allowing users to view property details and interact with booking pages.

## Requirements

### 1. Backend Setup

- Set up an Express.js server with Node.js.
- Create static JSON data for property listings.
- Implement basic API endpoints to serve data for listings and bookings.

### 2. API Endpoints

- **Get all listings:** GET `/api/listings`
- **Get listing details by ID:** GET `/api/listings/:id`
- **Search functionality** (basic filter using static data): GET `/api/listings/search?query=<location>`
- **Create a booking (mock):** POST `/api/bookings`

### 3. Frontend Components and Pages

- **Listing Details Page:**
  - Display property information such as images, title, type, amenities, number of guests, bedrooms, bathrooms, and price per night.
  - Include a **Book Now** button that redirects to the booking page.
- **Booking Page:**
  - Display a form for the user to input booking details (e.g., check-in and check-out dates).
  - Implement basic form validation.
  - Show a summary of the selected property and total price calculation.

### 4. React Router Integration

- Install and set up `react-router-dom` for navigation between pages.
- Define routes for:
  - **Homepage:** `/`
  - **Listing Details:** `/listings/:id`
  - **Booking Page:** `/book/:id`
- Ensure navigation between the homepage, listing details, and booking pages works seamlessly.

## 5. Backend Routes in Node.js

- Add the following routes in the Express.js server:
  - **GET /api/listings**: Fetch all listings.
  - **GET /api/listings/**  
: Fetch a specific listing by ID.
  - **POST /api/bookings**: Mock route for booking data submission, responding with a success message.

## 6. Frontend Integration

- Update **ListingCard** components to link to their respective **Listing Details Page**.
- Connect the **Listing Details Page** to fetch data using the ID from the URL.
- Implement the **Booking Page** to handle form submission and display user-entered data (mock booking confirmation).

## 7. State Management Enhancements

- Use `useState` and `useEffect` hooks to manage data fetching and form states.

## 8. Responsive Design

- Ensure all pages are responsive and maintain a consistent design.

## 9. Styling

- Continue using your chosen styling method for cohesive UI/UX.

## Deliverables

1. **GitHub repository link** with full source code for the React app and Node.js backend.
2. **README.md file** with:
  - Project description.
  - Instructions for running the frontend and backend servers.
  - Details about routing and navigation.

## Evaluation Criteria

- Proper implementation of the **Listing Details Page** and **Booking Page**.
- Functional routes in the backend.
- Correct data fetching and rendering on the frontend.
- Clean code structure and responsiveness.