

# Property Rental Management System – Backend Flow & REST API

## 1. System Flow Overview

The platform follows a real-world rental lifecycle:

- 1) User browses properties.
- 2) Tenant creates a booking request.
- 3) Property manager approves or rejects the booking.
- 4) If approved, a rental (active lease or stay) is created.
- 5) Payments are made against the rental over time.
- 6) Rental eventually completes or is terminated.

### Core relationship:

PROPERTY → BOOKING → RENTAL → PAYMENTS

## 2. Role-Based Access Summary

Admin:

- Manage all users
- Approve property manager applications
- Manage categories
- View all bookings, rentals, and payments
- Delete any property

Property Manager:

- Manage only their properties
- Upload images
- Approve/reject bookings for their properties
- View rentals and payments for their properties

Tenant:

- Register and login
- Browse properties
- Create bookings
- Make payments
- View their own rentals and payments

## 3. Booking → Rental → Payment Explanation

Booking:

A reservation request for a date range. No money is finalized yet.

Rental:

Created only after booking approval. Represents real occupancy or lease.

Payments:

Each row is a financial transaction tied to a rental.

Supports:

- Monthly payments
- Partial payments
- Deposits
- Full payments

#### 4. Full REST API Endpoints

**AUTH**

POST /auth/register

POST /auth/login

GET /auth/me

**ADMIN**

GET /users

PATCH /users/:id/role

PATCH /users/:id

POST /categories

PATCH /categories/:id

DELETE /categories/:id

GET /admin/manager-applications

GET /admin/manager-applications/:id

PATCH /admin/manager-applications/:id/approve

PATCH /admin/manager-applications/:id/reject

PATCH /admin/property/:id/approve

PATCH /admin/property/:id/reject

PATCH /admin/property/:id/suspend

GET /reports/payments

GET /admin/payments

GET /admin/payments/summary

**PROPERTY MANAGER**

POST /properties

PATCH /properties/:id

DELETE /properties/:id

GET /properties/mine

POST /properties/:id/images

PATCH /bookings/:id/approve

PATCH /bookings/:id/reject

GET /rentals/property/:propertyId  
GET /payments/property/:propertyId  
GET /payments/summary/mine

## **USER**

GET /properties  
GET /properties/:id  
POST /bookings  
GET /bookings/me  
POST /rentals/:id/payments  
GET /payments/me  
POST /manager-applications  
GET /manager-applications/me  
GET /payments/me  
GET /payments/:id