

# Room Booking SaaS – Admin Portal & Booking System

This is a Firebase-based SaaS platform with:

- A **React frontend** for room booking
- A **Node.js backend API** (Firebase Admin SDK) for managing users and roles

## Firebase Rules & API Guide

This explains how the Firebase Security Rules work and how to use the backend API to manage users and roles.

### Firestore Rules Explanation

These rules are already applied to your Firebase project. Here's how they work:

```
rules_version = '2';
service cloud.firestore {
  match /databases/{database}/documents {

    // Superadmin can read/write everything
    match /{companyId}/{docId} {
      allow read, write: if request.auth.token.role == 'superadmin';
    }

    // Companyadmin and users can only access their company's data
    match /{companyId}/{docId} {
      allow read, write: if request.auth.token.company == companyId &&
        request.auth.token.role in ['companyadmin', 'user'];
    }
  }
}
```

### Rule Logic Summary

Role	Access Description
superadmin	Full access to all collections and documents
companyadmin	Can access/edit only their company's data (e.g., Dell, Lux, etc.)
user	Can access their company's data, mainly for room booking

- **Collections:** Each document is organized under a collection named after the company.  
Example: `/Dell/Rooms`, `/Dell/Users`

### API Endpoints (Node.js – Firebase Admin SDK)

All APIs are protected via Firebase Admin and can be called by your React frontend.

### **POST /make-companyadmin**

Access: Superadmin only Purpose: Create a company admin and assign a company

Request:

```
{
  "requesterUid": "<SUPERADMIN_UID>",
  "email": "admin@company.com",
  "password": "yourpass",
  "company": "Dell"
}
```

▯ Assigns role companyadmin and company name as custom claims.

### **POST /make-user**

Access: Companyadmin only Purpose: Create a regular user under a company

Request:

```
{
  "requesterUid": "<COMPANYADMIN_UID>",
  "email": "user@company.com",
  "password": "yourpass",
  "company": "Dell"
}
```

▯ Assigns role user with the correct company claim.

### **POST /getlistcollection**

Access: Superadmin only Purpose: View all root-level collections and documents

Request:

```
{
  "uid": "<SUPERADMIN_UID>"
}
```

▯ Useful to verify all companies and data.

## **Deployed link**

Frontend: <https://roomly-frontend.vercel.app/>

**Please do not share backend url with anyone however it already have strict security on it**

Backend: <https://roomly-nodejs.vercel.app/>

## **Helpful Links**

- [Firebase Console – Authentication](#)

---

## Ownership Note

You now own the Firebase project. My access has been removed.