**Index.js**

**require("dotenv").config();**

**const express = require("express");**

**const cors = require("cors");**

**const bodyParser = require("body-parser");**

**const uniqid = require('uniqid');**

**const app = express();**

**app.use(cors());**

**app.use(bodyParser.json());**

**const port = process.env.PORT || 3000;**

**app.listen(port, () => console.log(`Your app is running with ${port}`));**

**let rooms = [];**

**let roomNo = 100;**

**let bookings =[];**

**let date\_regex = /^(0[1-9]|1[0-2])\/(0[1-9]|1\d|2\d|3[01])\/(19|20)\d{2}$/;**

**// let time\_regex = /^(0[1-9]|1\d|2[0-3])\:(0[1-9]|1\d|2\d|3\d|4\d|5\d)/;**

**let time\_regex = /^(0[0-9]|1\d|2[0-3])\:(00)/;**

**app.get("/", function (req, res) {**

**res.json({**

**output: "Homepage"**

**});**

**});**

**app.get("/getAllRooms", function (req, res) {**

**res.json({**

**output: rooms**

**});**

**});**

**app.get("/getAllBookings", function (req, res) {**

**res.json({**

**output: bookings**

**});**

**});**

**app.post("/createRoom", function (req, res) {**

**let room = {};**

**room.id = uniqid();**

**room.roomNo = roomNo;**

**room.bookings = [];**

**if(req.body.noSeats){room.noSeats = req.body.noSeats} else{res.status(400).json({ output: 'Please specify No of seats for Room'})};**

**if(req.body.amenities){room.amenities = req.body.amenities} else{res.status(400).json({ output: 'Please specify all Amenities for Room in Array format'})};**

**if(req.body.price){room.price = req.body.price} else{res.status(400).json({ output: 'Please specify price per hour for Room'})};**

**rooms.push(room);**

**roomNo++;**

**res.status(200).json({ output: 'Room Created Successfully'})**

**});**

**app.post("/createBooking", function (req, res) {**

**let booking = {};**

**booking.id = uniqid();**

**if(req.body.custName){booking.custName = req.body.custName} else{res.status(400).json({ output: 'Please specify customer Name for booking.'})};**

**if(req.body.date){**

**if (date\_regex.test(req.body.date)) {**

**booking.date = req.body.date**

**} else{**

**res.status(400).json({ output: 'Please specify date in MM/DD/YYYY'})**

**}**

**} else{**

**res.status(400).json({ output: 'Please specify date for booking.'})**

**}**

**if(req.body.startTime){**

**if (time\_regex.test(req.body.startTime)) {**

**booking.startTime = req.body.startTime**

**} else{**

**res.status(400).json({ output: 'Please specify time in hh:min(24-hr format) where minutes should be 00 only'})**

**}**

**} else{**

**res.status(400).json({ output: 'Please specify Starting time for booking.'})**

**}**

**if(req.body.endTime){**

**if (time\_regex.test(req.body.endTime)) {**

**booking.endTime = req.body.endTime**

**} else{**

**res.status(400).json({ output: 'Please specify time in hh:min(24-hr format) where minutes should be 00 only'})**

**}**

**} else{**

**res.status(400).json({ output: 'Please specify Ending time for booking.'})**

**}**

**const availableRooms = rooms.filter(room => {**

**if(room.bookings.length == 0){**

**return true;**

**} else{**

**room.bookings.filter(book =>{**

**if((book.date == req.body.date) ){**

**if((parseInt((book.startTime).substring(0, 1)) > parseInt((req.body.startTime).substring(0, 1)) ) &&**

**(parseInt((book.startTime).substring(0, 1)) > parseInt((req.body.endTime).substring(0, 1)) ) ){**

**if((parseInt((book.startTime).substring(0, 1)) < parseInt((req.body.startTime).substring(0, 1)) ) &&**

**(parseInt((book.startTime).substring(0, 1)) < parseInt((req.body.endTime).substring(0, 1)) ) ){**

**return true;**

**}**

**}**

**}**

**else{**

**return true;**

**}**

**})**

**}**

**});**

**if(availableRooms.length == 0){res.status(400).json({ output: 'No Available Rooms on Selected Date and Time'})}**

**else{**

**roomRec = availableRooms[0];**

**let count =0;**

**rooms.forEach(element => {**

**if(element.roomNo == roomRec.roomNo){**

**rooms[count].bookings.push({**

**custName: req.body.custName,**

**startTime: req.body.startTime,**

**endTime: req.body.endTime,**

**date: req.body.date**

**})**

**}**

**count++;**

**});**

**let bookingRec = req.body;**

**bookingRec.roomNo = roomRec.roomNo;**

**bookingRec.cost = parseInt(roomRec.price) \* (parseInt((bookingRec.endTime).substring(0, 1)) - parseInt((bookingRec.startTime).substring(0, 1)));**

**bookings.push(bookingRec);**

**res.status(200).json({ output: 'Room Booking Successfully'})**

**}**

**});**