User.js

const UserSchema = new Schema(

{

name: { type: String, required: true },

email: { type: String, required: true, unique: true },

password: { type: String, required: true },

createOn: { type: Date, default: new Date().getTime() },

cmnd: { type: String, require: true },

updatedAt: { type: Date, default: new Date().getTime() },

role: { type: String, enum: ["CUSTOMER", "ADMIN", "OWNER"], default: "CUSTOMER" },

reservations: [

{ type: Schema.Types.ObjectId, ref: "Reservation" },

],

ownedHotels: [

{

type: Schema.Types.ObjectId,

ref: "Hotel",

},

], // Mảng các khách sạn người dùng sở hữu (tham chiếu đến Hotel),

favorites: [

{

type: Schema.Types.ObjectId,

ref: "Hotel",

},

], // Mảng các khách sạn yêu thích (tham chiếu đến Hotel)

},

{ versionKey: false }

);

// Schema cho Hotel

const hotelSchema = new Schema(

{

hotelName: {

type: String,

required: true,

},

owner: {

type: Schema.Types.ObjectId,

ref: "User",

},

description: {

type: String,

required: true,

},

address: {

type: String,

required: true,

},

services: [

{

type: Schema.Types.ObjectId,

ref: "HotelService",

},

],

facility: [

{

type: Schema.Types.ObjectId,

ref: "Facility",

},

],

rating: {

type: Number,

required: true,

},

pricePerNight: {

type: Number,

required: true,

},

images: [

{

type: String,

required: true

},

],

},

{ versionKey: false }

);

module.exports = mongoose.model("Hotel", hotelSchema);

const RoomSchema = new Schema({

type: { type: String, required: true },

price: { type: Number, required: true },

capacity: {type: Number, required: true},

description: {type: String, required: true},

images: [{ type: String, required: true}],

quantity: {type: Number, required: true}

//References

hotel: { type: Schema.Types.ObjectId, ref: 'Hotel', required: true },

bed: [{

bed: { type: Schema.Types.ObjectId, ref: 'Bed', required: true },

quantity: { type: Number, required: true }

}],

facilities: [{ type: Schema.Types.ObjectId, ref: 'RoomFacility'}],

}, {versionKey: false});

module.exports = mongoose.model('Room', RoomSchema);

const RoomFacilitySchema = new Schema({

room: { type: mongoose.Schema.Types.ObjectId, ref: 'Room', required: true },

name: { type: String, required: true },

description: { type: String },

url: {type: Buffer, required: true},

}, {versionKey: false});

module.exports = mongoose.model('RoomFacility', RoomFacilitySchema);

const reservationSchema = new Schema(

{

user: {

type: mongoose.Schema.Types.ObjectId,

ref: "User",

required: true,

},

hotel: {

type: mongoose.Schema.Types.ObjectId,

ref: "Hotel",

required: true,

},

rooms: [

{

type: mongoose.Schema.Types.ObjectId,

ref: "Room",

required: true,

},

], // Danh sách phòng được đặt

checkInDate: {

type: Date,

required: true,

},

checkOutDate: {

type: Date,

required: true,

},

status: {

type: String,

enum: [

"BOOKED", // Đã đặt, trả tiền nhưng chưa check-in

"CHECKED IN", // Đang ở, đã check-in

"CHECKED OUT", // Đã check-out, có thể để lại phản hồi

"COMPLETED", // Hoàn thành, đã phản hồi

"PENDING", // Chờ xử lý hoặc xác nhận

"CANCELLED", // Đã hủy

"NOT PAID" // Chưa trả tiền

],

default: "PENDING", // Mặc định là chờ xử lý

},

totalPrice: {

type: Number,

required: true,

default: 0

},

},

{ timestamps: true },

{versionKey: false}

);

module.exports = mongoose.model("Reservation", reservationSchema);

const HotelServiceSchema = new Schema({

hotel: {

type: mongoose.Schema.Types.ObjectId,

ref: 'Hotel',

required: true

}, // Tham chiếu đến mô hình Hotel

name: {

type: String,

required: true,

},

price: {

type: Number,

required: true

} // Giá của dịch vụ tại khách sạn này

}, {versionKey: false});

module.exports = mongoose.model('HotelService', HotelServiceSchema);

const FacilitySchema = new Schema({

name: { type: String, required: true },

description: { type: String },

url: {type: Buffer, required: true},

hotel: [

{

type: Schema.Types.ObjectId,

ref: "Hotel",

},

], // Mảng các khách sạn người dùng sở hữu (tham chiếu đến Hotel),

}, {versionKey: false});

module.exports = mongoose.model('Facility', FacilitySchema);

const bedSchema = new mongoose.Schema({

name: {

type: String,

required: true,

},

description: {

type: String,

required: true,

},

}, {versionKey: false});

module.exports = mongoose.model("Bed", bedSchema);

const FeedbackSchema = new Schema({

user: { type: mongoose.Schema.Types.ObjectId, ref: 'User', required: true }, //User ID

reservation: {type: Schema.Types.ObjectId, ref: 'Reservation', required: true}, //Reservation ID

hotel: { type: mongoose.Schema.Types.ObjectId, ref: 'Hotel', required: true }, //Hotel ID

content: { type: String, required: true },

rating: { type: Number, min: 1, max: 5, required: true },

createdAt: { type: Date, default: Date.now }

}, {versionKey: false});

module.exports = mongoose.model('Feedback', FeedbackSchema);

**SCRIPT INSERT DATABASE**

const feedbackContents = [

"Dịch vụ tuyệt vời! Nhân viên thân thiện và hỗ trợ nhanh chóng.",

"Khách sạn sạch sẽ, tiện nghi đầy đủ và rất thoải mái.",

"Bữa sáng rất ngon, phòng có view đẹp và thoáng đãng.",

"Trải nghiệm đáng nhớ, tôi sẽ quay lại lần sau!",

"Giá cả hợp lý, vị trí trung tâm thuận tiện di chuyển.",

"Hồ bơi sạch sẽ, nước ấm và có view rất đẹp.",

"Phòng ốc rộng rãi, yên tĩnh và giường rất êm.",

"Nhà hàng trong khách sạn có món ăn rất ngon.",

"Khu vực lễ tân làm việc rất chuyên nghiệp và nhanh nhẹn.",

"Khách sạn gần các điểm tham quan nổi tiếng, rất tiện lợi."

];

const roomNames = [

"Phòng Deluxe Hướng Biển",

"Phòng Superior View Thành Phố",

"Phòng Standard Giường Đôi",

"Phòng Gia Đình Cao Cấp",

"Suite Tổng Thống Sang Trọng",

"Phòng Executive Hạng Nhất",

"Phòng Tiết Kiệm Cho Cặp Đôi",

"Phòng Premium Có Ban Công",

"Phòng Cổ Điển với Nội Thất Gỗ",

"Phòng Nghỉ Dưỡng Bên Hồ"

];

const hotelDescriptions = [

"Nestled in the heart of the city, our hotel offers a luxurious stay with world-class amenities, fine dining, and personalized services. Whether you are here for business or leisure, expect an exceptional experience with breathtaking city views and top-notch hospitality.",

"Experience the grandeur of imperial elegance with our top-tier services. From opulent suites to exquisite dining options, every detail is crafted to ensure a regal and memorable stay. Indulge in our spa and wellness facilities for ultimate relaxation.",

"A peaceful retreat designed to offer tranquility and comfort in the midst of the bustling city. Our hotel features elegantly furnished rooms, a serene rooftop garden, and a variety of recreational activities to make your stay rejuvenating and enjoyable.",

"A seaside paradise where guests can wake up to the sound of waves and stunning sunrise views. Our oceanfront resort boasts an infinity pool, luxurious spa treatments, and a range of water sports for a truly unforgettable beachside escape.",

"A high-rise haven that offers spectacular panoramic views of the city skyline. Whether you are enjoying a drink at the rooftop bar or relaxing in your elegantly appointed suite, our hotel provides the perfect blend of style, comfort, and convenience.",

"An elite experience tailored for travelers seeking unparalleled luxury and sophistication. Our hotel provides exclusive access to premium lounges, fine dining restaurants, and private concierge services to cater to your every need.",

"A harmonious blend of comfort and style, our hotel offers a warm and inviting atmosphere for both business and leisure travelers. Enjoy our state-of-the-art fitness center, relaxing spa treatments, and gourmet cuisine prepared by top chefs.",

"Setting the golden standard for premium hospitality, our hotel combines timeless elegance with modern luxury. Each room is meticulously designed with plush furnishings, and our signature services ensure an extraordinary stay.",

"An urban sanctuary designed for modern travelers who seek both relaxation and adventure. With a contemporary design, cutting-edge facilities, and easy access to city attractions, our hotel is the ideal choice for an unforgettable stay.",

"A royal experience awaits you at our lush garden retreat, where you can immerse yourself in tranquility and luxury. With beautifully landscaped grounds, an award-winning spa, and world-class dining, our hotel is the perfect getaway for discerning guests."

];

const hotelNames = [

"Luxe Haven Hotel",

"Grand Imperial Hotel",

"Serene Stay Inn",

"Ocean Breeze Resort",

"Skyline Suites",

"Elite Retreat",

"Harmony Hotel",

"Golden Crest Hotel",

"Urban Oasis Hotel",

"Royal Garden Hotel"

];

const hotelAddresses = [

"12 Đường Nguyễn Huệ, Quận 1, Hồ Chí Minh",

"45 Lê Lợi, Quận 3, Hồ Chí Minh",

"78 Hoàng Diệu, Quận 4, Hồ Chí Minh",

"23 Trần Hưng Đạo, Quận 5, Hồ Chí Minh",

"90 Phan Xích Long, Quận Phú Nhuận, Hồ Chí Minh",

"56 Võ Văn Kiệt, Quận 2, Hồ Chí Minh",

"34 Tôn Đức Thắng, Quận 7, Hồ Chí Minh",

"67 Bùi Viện, Quận 1, Hồ Chí Minh",

"11 Đinh Tiên Hoàng, Quận Bình Thạnh, Hồ Chí Minh",

"102 Pasteur, Quận 3, Hồ Chí Minh",

"15 Phố Tràng Tiền, Hoàn Kiếm, Hà Nội",

"28 Hàng Bông, Hoàn Kiếm, Hà Nội",

"55 Đường Kim Mã, Ba Đình, Hà Nội",

"77 Nguyễn Chí Thanh, Đống Đa, Hà Nội",

"31 Trần Duy Hưng, Cầu Giấy, Hà Nội",

"89 Lý Thường Kiệt, Hoàn Kiếm, Hà Nội",

"102 Võ Nguyên Giáp, Sơn Trà, Đà Nẵng",

"200 Bạch Đằng, Hải Châu, Đà Nẵng",

"17 Nguyễn Văn Linh, Hải Châu, Đà Nẵng",

"65 Phan Châu Trinh, Hải Châu, Đà Nẵng",

"10 Nguyễn Văn Cừ, Ninh Kiều, Cần Thơ",

"88 Lý Thái Tổ, Thành phố Buôn Ma Thuột, Đắk Lắk",

"99 Trần Phú, Thành phố Nha Trang, Khánh Hòa",

"120 Hùng Vương, Thành phố Huế, Thừa Thiên Huế",

"35 Hoàng Diệu, Thành phố Vũng Tàu, Bà Rịa - Vũng Tàu",

"50 Nguyễn Trãi, Thành phố Hạ Long, Quảng Ninh",

"27 Lê Lợi, Thành phố Nam Định, Nam Định",

"19 Lê Hồng Phong, Thành phố Hải Phòng, Hải Phòng",

"65 Nguyễn Huệ, Thành phố Quy Nhơn, Bình Định",

"44 Trần Hưng Đạo, Thành phố Phan Thiết, Bình Thuận",

"78 Lê Duẩn, Thành phố Pleiku, Gia Lai",

"101 Hoàng Quốc Việt, Thành phố Rạch Giá, Kiên Giang",

"55 Võ Nguyên Giáp, Thành phố Đông Hà, Quảng Trị",

"92 Trường Chinh, Thành phố Tuy Hòa, Phú Yên",

"67 Nguyễn Thái Học, Thành phố Việt Trì, Phú Thọ",

"81 Phạm Ngũ Lão, Thành phố Bắc Ninh, Bắc Ninh",

"33 Hoàng Hoa Thám, Thành phố Lạng Sơn, Lạng Sơn",

"48 Lý Tự Trọng, Thành phố Cao Lãnh, Đồng Tháp",

"77 Nguyễn Văn Linh, Thành phố Bạc Liêu, Bạc Liêu"

];

const roomTypes = [

"Deluxe Room",

"Superior Room",

"Standard Room",

"Family Suite",

"Presidential Suite",

"Executive Suite",

"Budget Room",

"Premium Room",

"Classic Room",

"Resort Room"

];

const roomDescriptions = [

"Một căn phòng rộng rãi với tầm nhìn ra biển tuyệt đẹp.",

"Phòng có tầm nhìn đẹp hướng thành phố và nội thất sang trọng.",

"Căn phòng tiện nghi với giường đôi thoải mái.",

"Lý tưởng cho gia đình với không gian rộng rãi và ấm cúng.",

"Suite cao cấp với dịch vụ đặc quyền và không gian sang trọng.",

"Trải nghiệm đẳng cấp với tiện nghi hạng nhất.",

"Một lựa chọn tiết kiệm nhưng vẫn đảm bảo sự tiện nghi.",

"Căn phòng cao cấp với ban công riêng.",

"Không gian cổ điển với nội thất gỗ sang trọng.",

"Nơi lý tưởng để nghỉ dưỡng với hồ bơi gần kề."

];

const servicesName = [

"Free Wi-Fi",

"Airport Shuttle",

"24/7 Room Service",

"Spa and Wellness Center",

"Swimming Pool",

"Fitness Center",

"Conference Rooms",

"Laundry Service",

"Car Rental",

"Restaurant and Bar"

];

//Tự tạo USER nếu chạy local

const userIds = [

ObjectId("67ac892aba1cb210b487e1e1"),

ObjectId("67ac894aba1cb210b487e1e7"),

ObjectId("67ac8964ba1cb210b487e1ec"),

];

const hotelIds = [];

const serviceIds = [];

const roomIds = [];

const reservationIds = [];

const facilityIds = [];

const roomFacilityIds = []

// Danh sách loại giường cố định

const bedTypes = [

{ name: "Giường đơn", description: "Giường đơn thoải mái, thích hợp cho một người." },

{ name: "Giường đôi", description: "Giường đôi rộng rãi, phù hợp cho 2 người." },

{ name: "Giường 4 người", description: "Giường lớn có thể chứa đến 4 người." },

{ name: "2 giường đơn", description: "Phòng có 2 giường đơn, phù hợp cho nhóm 2 người." }

];

// Insert Beds

const bedDocs = db.beds.insertMany(bedTypes);

const bedIds = Object.values(bedDocs.insertedIds);

// Danh sách facility có sẵn

const facilitiesName = [

"Wi-Fi miễn phí",

"Hồ bơi",

"Bãi đỗ xe",

"Dịch vụ phòng 24/7",

"Nhà hàng",

"Trung tâm thể hình"

];

const facilitiesDescriptions = [

"Internet tốc độ cao miễn phí cho khách hàng.",

"Hồ bơi rộng rãi, sạch sẽ và hiện đại.",

"Bãi đỗ xe miễn phí dành cho khách hàng lưu trú.",

"Hỗ trợ dịch vụ phòng mọi lúc.",

"Nhà hàng phục vụ đa dạng món ăn ngon.",

"Phòng tập gym đầy đủ trang thiết bị hiện đại."

];

const roomFacilitiesName = [

"Air Conditioning",

"Flat-screen TV",

"Mini Bar",

"Private Bathroom",

"Coffee Maker",

"High-speed Wi-Fi",

"In-room Safe",

"Work Desk",

"Soundproofing",

"Balcony"

];

const roomFacilitiesDescriptions = [

"Provides cool and comfortable air during hot days.",

"Enjoy your favorite shows on a high-definition screen.",

"Stocked with a selection of snacks and beverages.",

"Includes a shower, bathtub, and complimentary toiletries.",

"Brew fresh coffee in the comfort of your room.",

"Stay connected with fast and reliable internet.",

"Securely store your valuables and important documents.",

"A comfortable workspace for business travelers.",

"Ensures a quiet and peaceful stay.",

"Enjoy a private outdoor space with scenic views."

];

//Insert Room Facility

for (let i = 1; i <= 40; i++) {

let randomIndex = i % roomFacilitiesName.length;

let randomRoomId = roomIds[i % roomIds.length]

let roomFacility = db.roomfacilities.insertOne({

room: randomRoomId,

name: roomFacilitiesName[randomIndex],

description: roomFacilitiesDescriptions[randomIndex],

url:`https://picsum.photos/600/400?random=${i \* 5}`

});

roomFacilityIds.push(roomFacility.insertedId);

}

//Insert facility

for (let i = 1; i <= 40; i++) {

let randomIndex = i % facilitiesName.length;

let randomHotelId = hotelIds[i % hotelIds.length]

let facilities = db.facilities.insertOne({

name: hotelNames[randomIndex],

description: hotelDescriptions[randomIndex],

hotel: randomHotelId,

});

facilityIds.push(facilities.insertedId);

}

// Insert 100 hotels (Mỗi khách sạn có 5 ảnh và 3-5 facility)

for (let i = 1; i <= 100; i++) {

let randomIndex = i % hotelNames.length;

let hotelFacilityIds = facilityIds.sort(() => 0.5 - Math.random()).slice(0, Math.floor(Math.random() \* 3) + 3);

let hotelServiceIds = serviceIds.sort(() => 0.5 - Math.random()).slice(0, Math.floor(Math.random() \* 3) + 3);

let hotel = db.hotels.insertOne({

hotelName: hotelNames[randomIndex],

owner: userIds[i % userIds.length],

description: hotelDescriptions[randomIndex],

address: hotelAddresses[randomIndex % hotelAddresses.length],

services: hotelServiceIds,

facility: hotelFacilityIds,

rating: Math.floor(Math.random() \* 5) + 1,

pricePerNight: Math.floor(Math.random() \* 200) + 50,

images: [

`https://picsum.photos/600/400?random=${i \* 5}`,

`https://picsum.photos/600/400?random=${i \* 5 + 1}`,

`https://picsum.photos/600/400?random=${i \* 5 + 2}`,

`https://picsum.photos/600/400?random=${i \* 5 + 3}`,

`https://picsum.photos/600/400?random=${i \* 5 + 4}`

]

});

hotelIds.push(hotel.insertedId);

}

// Insert 30 random hotel services

for (let i = 0; i < 30; i++) {

let randomIndex = i % servicesName.length

let randomHotelId = hotelIds[i % hotelIds.length]

let randomPrice = Math.floor(Math.random() \* 100) + 10;

let hotelService = db.hotelservices.insertOne({

hotel: randomHotelId,

name: servicesName[randomIndex],

price: randomPrice

});

// Optionally store the service ID for later reference

serviceIds.push(hotelService.insertedId);

}

// Insert 60 rooms (Mỗi phòng có 3 ảnh)

for (let i = 0; i < 50; i++) {

for (let j = 0; j < 3; j++) {

let selectedBeds = [

{ bed: bedIds[j % 4], quantity: Math.floor(Math.random() \* 3) + 1 }

];

let room = db.rooms.insertOne({

name: roomNames[Math.floor(Math.random() \* roomNames.length)],

type: roomTypes[Math.floor(Math.random() \* roomTypes.length)],

price: Math.floor(Math.random() \* 200) + 50,

capacity: Math.floor(Math.random() \* 4) + 1,

description: roomDescriptions[Math.floor(Math.random() \* roomDescriptions.length)],

images: [

`https://picsum.photos/300/200?random=${i \* 3 + j}`,

`https://picsum.photos/300/200?random=${i \* 3 + j + 1}`,

`https://picsum.photos/300/200?random=${i \* 3 + j + 2}`

],

quantity: Math.floor(Math.random() \* 5) + 1,

hotel: hotelIds[i],

bed: selectedBeds

});

roomIds.push(room.insertedId);

}

}

// Trạng thái của Reservation

const reservationStatuses = [

"BOOKED", // Đã đặt, trả tiền nhưng chưa check-in

"CHECKED IN", // Đang ở, đã check-in

"CHECKED OUT", // Đã check-out, có thể để lại phản hồi

"COMPLETED", // Hoàn thành, đã phản hồi

"PENDING", // Chờ xử lý hoặc xác nhận

"CANCELLED", // Đã hủy

"NOT PAID" // Chưa trả tiền

];

// Insert 30 Reservations - Trạng thái ngẫu nhiên

for (let i = 0; i < 100; i++) {

let randomStatus = reservationStatuses[Math.floor(Math.random() \* reservationStatuses.length)];

let reservation = db.reservations.insertOne({

user: userIds[i % 3],

hotel: hotelIds[i % 20],

rooms: [roomIds[(i \* 3) % 60], roomIds[(i \* 3 + 1) % 60]],

checkInDate: new Date("2024-06-01"),

checkOutDate: new Date("2024-06-05"),

status: randomStatus,

totalPrice: Math.floor(Math.random() \* 500) + 100

});

reservationIds.push(reservation.insertedId);

}

// Insert 10 Feedbacks - chỉ áp dụng với reservation có trạng thái hợp lệ

const feedbackSet = new Set();

for (let i = 0; i < 40; i++) {

let randomIndex = i % feedbackContents.length;

let userId = userIds[i % 3];

let hotelId = hotelIds[i % 20];

let reservation = db.reservations.findOne({ \_id: reservationIds[i] });

// Chỉ cho phép feedback với reservation có trạng thái CHECKED OUT hoặc COMPLETED

if (["COMPLETED"].includes(reservation.status)) {

let feedbackKey = `${userId}-${hotelId}`;

if (!feedbackSet.has(feedbackKey)) {

db.feedbacks.insertOne({

user: userId,

reservation: reservationIds[i],

hotel: hotelId,

content: feedbackContents[randomIndex],

rating: Math.floor(Math.random() \* 5) + 1,

createdAt: new Date()

});

feedbackSet.add(feedbackKey);

}

}

}