const bookingSchema = require("../schemas/bookingModel");

const propertySchema = require("../schemas/propertyModel");

const userSchema = require("../schemas/userModel");

//////////adding property by owner////////

const addPropertyController = async (req, res) => {

try {

let images = [];

if (req.files) {

images = req.files.map((file) => ({

filename: file.filename,

path: `/uploads/${file.filename}`,

}));

}

const user = await userSchema.findById({ \_id: req.body.userId });

const newPropertyData = new propertySchema({

...req.body,

propertyImage: images,

ownerId: user.\_id,

ownerName: user.name,

isAvailable: "Available",

});

await newPropertyData.save();

return res.status(200).send({

success: true,

message: "New Property has been stored",

});

} catch (error) {

console.log("Error in get All Users Controller ", error);

}

};

///////////all properties of owner/////////

const getAllOwnerPropertiesController = async (req, res) => {

const { userId } = req.body;

try {

const getAllProperties = await propertySchema.find();

const updatedProperties = getAllProperties.filter(

(property) => property.ownerId.toString() === userId

);

return res.status(200).send({

success: true,

data: updatedProperties,

});

} catch (error) {

console.error(error);

return res

.status(500)

.send({ message: "Internal server error", success: false });

}

};

//////delete the property by owner/////

const deletePropertyController = async (req, res) => {

const propertyId = req.params.propertyid;

try {

await propertySchema.findByIdAndDelete({

\_id: propertyId,

});

return res.status(200).send({

success: true,

message: "The property is deleted",

});

} catch (error) {

console.error(error);

return res

.status(500)

.send({ message: "Internal server error", success: false });

}

};

//////updating the property/////////////

const updatePropertyController = async (req, res) => {

const { propertyid } = req.params;

console.log(req.body);

try {

const property = await propertySchema.findByIdAndUpdate(

{ \_id: propertyid },

{

...req.body,

ownerId: req.body.userId,

},

{ new: true }

);

return res.status(200).send({

success: true,

message: "Property updated successfully.",

});

} catch (error) {

console.error("Error updating property:", error);

return res.status(500).json({

success: false,

message: "Failed to update property.",

});

}

};

const getAllBookingsController = async (req, res) => {

const { userId } = req.body;

try {

const getAllBookings = await bookingSchema.find();

const updatedBookings = getAllBookings.filter(

(booking) => booking.ownerID.toString() === userId

);

return res.status(200).send({

success: true,

data: updatedBookings,

});

} catch (error) {

console.error(error);

return res

.status(500)

.send({ message: "Internal server error", success: false });

}

};

//////////handle bookings status//////////////

const handleAllBookingstatusController = async (req, res) => {

const { bookingId, propertyId, status } = req.body;

try {

const booking = await bookingSchema.findByIdAndUpdate(

{ \_id: bookingId },

{

bookingStatus: status,

},

{

new: true,

}

);

const property = await propertySchema.findByIdAndUpdate(

{ \_id: propertyId },

{

isAvailable: status === 'booked' ? 'Unavailable' : 'Available',

},

{ new: true }

);

return res.status(200).send({

success: true,

message: `changed the status of property to ${status}`,

});

} catch (error) {

console.error(error);

return res

.status(500)

.send({ message: "Internal server error", success: false });

}

};

module.exports = {

addPropertyController,

getAllOwnerPropertiesController,

deletePropertyController,

updatePropertyController,

getAllBookingsController,

handleAllBookingstatusController,

};