Homestay Booking System Using MongoDB

1. Overview

A sample MongoDB schema and utility functions for a homestay booking platform connecting tourists with native hosts offering short-term stays.

---

2. Collections and Sample Data

2.1. Users Collection

```
db.users.insertOne({
  name: "John Doe",
  email: "johndoe@example.com",
  phone: "+1234567890",
  address: "123 Main St, New York, NY",
  bookings: [ObjectId("661e2abc1234567890abcdef")],
  preferredRoomType: "Deluxe",
  status: "active",
  dateJoined: new Date("2024-01-15"),
  lastLogin: new Date("2025-04-16T10:00:00Z")
});
```

2.2. Hosts Collection

```
db.hosts.insertOne({
  name: "Alice Smith",
  email: "alicesmith@example.com",
  phone: "+1987654321",
  location: "Los Angeles, CA",
  roomsAvailable: 5,
  rating: 4.7,
  pricePerNight: 120,
  roomTypes: [ObjectId("661e2def1234567890abcdef")],
  amenities: [ObjectId("661e2f001234567890abcdef")],
  description: "Cozy homestay near downtown with all modern facilities.",
  status: "active",
  joinDate: new Date("2023-11-12"),
  reviews: [ObjectId("661e2f321234567890abcdef")]
});
```

2.3. Admins Collection

```
db.admins.insertOne({
  username: "superadmin",
  email: "admin@example.com",
  role: "SuperAdmin",
  password: "hashed_encrypted_password_here",
  permissions: [ObjectId("661e2f621234567890abcdef")],
  dateJoined: new Date("2022-09-01")
});
```

2.4. Bookings Collection

```
db.bookings.insertOne({
  customerId: ObjectId("661e2aa01234567890abcdef"),
  hostId: ObjectId("661e2dd01234567890abcdef"),
  roomType: ObjectId("661e2def1234567890abcdef"),
  startDate: new Date("2025-05-01"),
  endDate: new Date("2025-05-04"),
  status: "booked",
  durationDays: 3,
  totalPrice: 360,
  specialRequests: "Late check-in",
  paymentStatus: "paid",
  createdAt: new Date()
});
```

2.5. Reviews Collection

```
db.reviews.insertOne({
  userId: ObjectId("661e2aa01234567890abcdef"),
  hostId: ObjectId("661e2dd01234567890abcdef"),
  rating: 5,
  reviewText: "Amazing host and very clean room. Highly recommend!",
  reviewDate: new Date()
});
```

2.6. Transactions Collection

```
db.transactions.insertOne({
  bookingId: ObjectId("661e30201234567890abcdef"),
  customerId: ObjectId("661e2aa01234567890abcdef"),
  amount: 360,
  paymentMethod: "Credit Card",
  paymentStatus: "successful",
  transactionDate: new Date()
});
```

2.7. Rooms Collection

```
db.rooms.insertOne({
  hostId: ObjectId("661e2dd01234567890abcdef"),
  roomType: ObjectId("661e2def1234567890abcdef"),
  pricePerNight: 120,
  amenities: [ObjectId("661e2f001234567890abcdef")],
  availability: "available",
  description: "Spacious room with sea view, air conditioning, and balcony."
});
```

3. Utility Functions

3.1. Book a Room

```
function bookRoom(customerId, hostId, roomType, startDate, endDate, totalPrice) {
  const booking = db.bookings.insertOne({
    customerId: ObjectId(customerId),
    hostId: ObjectId(hostId),
    roomType: ObjectId(roomType),
    startDate: new Date(startDate),
    endDate: new Date(endDate),
    status: "booked",
    durationDays: (new Date(endDate) - new Date(startDate)) / (1000 * 60 * 60 * 24),
    totalPrice: totalPrice,
    specialRequests: "",
    paymentStatus: "pending",
    createdAt: new Date()
  });
  db.users.updateOne({ _id: ObjectId(customerId) }, { $push: { bookings: booking.insertedId } });
  db.hosts.updateOne({ _id: ObjectId(hostId) }, { $inc: { roomsAvailable: -1 } });
}
```

3.2. Cancel a Booking

```
function cancelBooking(bookingId) {
  const booking = db.bookings.findOne({ _id: ObjectId(bookingId) });
  db.bookings.updateOne({ _id: ObjectId(bookingId) }, { $set: { status: "cancelled" } });
  db.hosts.updateOne({ _id: ObjectId(booking.hostId) }, { $inc: { roomsAvailable: 1 } });
  db.users.updateOne({ _id: ObjectId(booking.customerId) }, { $pull: { bookings: ObjectId(bookingId) } });
}
```

3.3. Complete a Booking

```
function completeBooking(bookingId) {
  db.bookings.updateOne({ _id: ObjectId(bookingId) }, { $set: { status: "completed" } });
}
```

3.4. Vacate Room

```
function vacateRoom(bookingId) {
  const booking = db.bookings.findOne({ _id: ObjectId(bookingId) });
  db.bookings.updateOne({ _id: ObjectId(bookingId) }, { $set: { status: "vacated" } });
  db.hosts.updateOne({ _id: ObjectId(booking.hostId) }, { $inc: { roomsAvailable: 1 } });
}
```

3.5. Leave a Review

```
function leaveReview(userId, hostId, rating, reviewText) {
  const review = db.reviews.insertOne({
    userId: ObjectId(userId),
    hostId: ObjectId(hostId),
    rating: rating,
    reviewText: reviewText,
    reviewDate: new Date()
  });
  db.hosts.updateOne({ _id: ObjectId(hostId) }, { $push: { reviews: review.insertedId } });
}
```