Database Design and Development Report

Date	27th JUNE 2025
Team ID	LTVIP2025TMID47699
Project Name	HouseHunt – House Rent Web Application

Project Title: HouseHunt – House Rent Web Application

Date: 27th JUNE 2025

Objective

The objective of this report is to outline the database design and implementation details for the HouseHunt – House Rent Web Application project, including schema design and database management system (DBMS) integration.

Technologies Used

- **Database Management System (DBMS):** MongoDB
- **Object-Document Mapper (ODM):** Mongoose

Design the Database Schema

The database schema is designed to accommodate the following entities and relationships:

1. Users

- Attributes:
  - _id: ObjectId
  - name: String
  - email: String (unique)
  - password: String
  - createdAt: Date
  - updatedAt: Date

2. Properties

- Attributes:

  - _id: ObjectId
  - title: String
  - description: String
  - location: String
  - price: Number
  - owner: ObjectId (references User)
  - createdAt: Date
  - updatedAt: Date

3. Bookings

- Attributes:
  - _id: ObjectId
  - property: ObjectId (references Property)
  - user: ObjectId (references User)
  - bookingDate: Date
  - createdAt: Date
  - updatedAt: Date

Implement the Database using MongoDB

The MongoDB database is implemented with the following collections and structures:

Database Name: [your_database_name]

1. Collection: users

- Schema:

```
{
 _id: ObjectId
 name: String
email: String (unique)
password: String
createdAt: Date
updatedAt: Date
}
```

2. Collection: properties

 - Schema:

```
{
_id: ObjectId
title: String
description: String
 location: String
 price: Number
 owner: ObjectId (references User)
 createdAt: Date
 updatedAt: Date
}
```

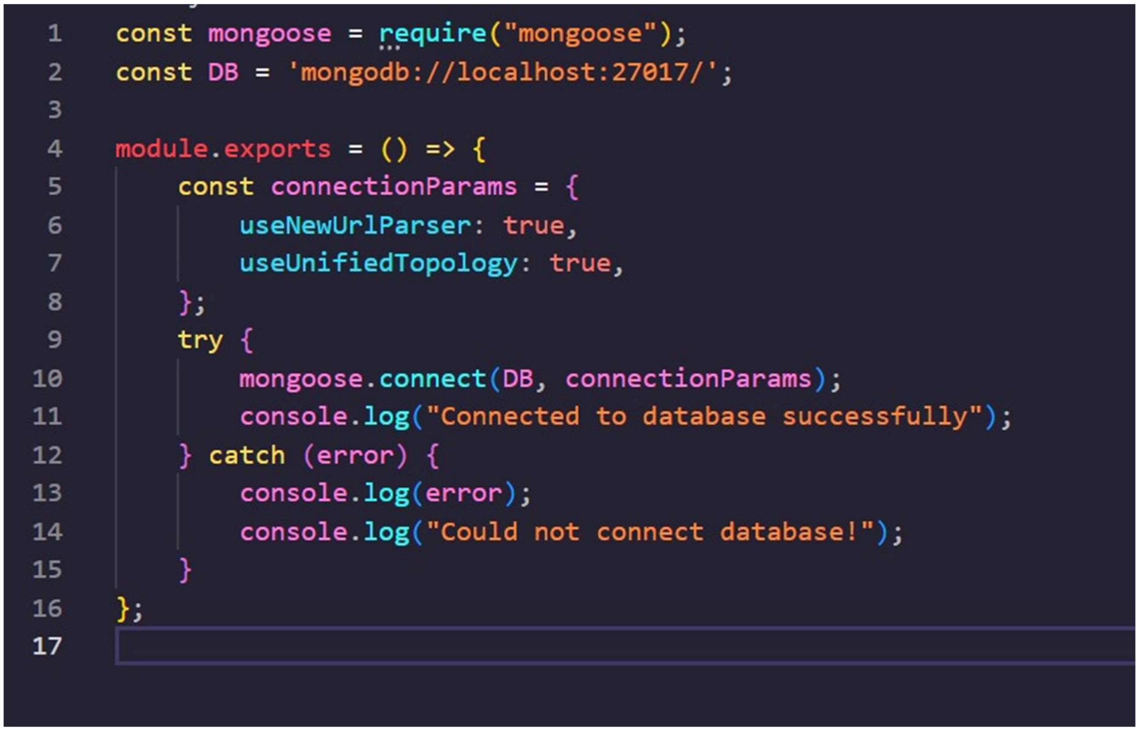
3. Collection: bookings

 - Schema:

```
{
 _id: ObjectId
property: ObjectId (references Property)
 user: ObjectId (references User)
 bookingDate: Date
 createdAt: Date
 updatedAt: Date
}
```

Integration with Backend

- Database connection:



```
const mongoose = require("mongoose");
const DB = 'mongodb://localhost:27017/';

module.exports = () => {
    const connectionParams = {
        useNewUrlParser: true,
        useUnifiedTopology: true,
    };
    try {
        mongoose.connect(DB, connectionParams);
        console.log("Connected to database successfully");
    } catch (error) {
        console.log(error);
        console.log("Could not connect database!");
    }
};

```

- The backend APIs interact with MongoDB using Mongoose ODM Key interactions include:

    - User Management: CRUD operations for users.

    - Posting Management: CRUD operations for posting properties with user authentication.

Booking Management: CRUD operations for booking properties.