**Database Design and Development Report**

|  |  |
| --- | --- |
| Date | 18th July 2024 |
| Team ID | SWTID1720019378 |
| Project Name | HouseHunt – House Rent Web Application |
| Maximum Marks |  |

**Project Title**: HouseHunt – House Rent Web Application

**Date**: 18th July 2024

**Prepared by**: MERN Stack Nexus

**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

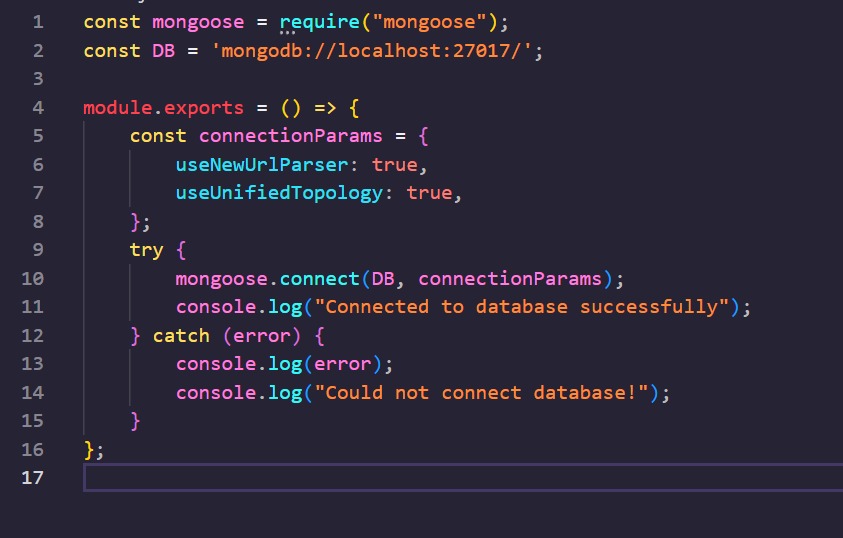
updatedAt: Date

}

```

**Integration with Backend**

* Database connection:



* 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.