**Full Stack Development with MERN**

**Database Design and Development Report**

|  |  |
| --- | --- |
| Date | 17-07-2024 |
| Team ID | SWTID1720010107 |
| Project Name | HIRE CONNECT |
| Maximum Marks | 15 |

**Project Title**: HIRE CONNECT

**Date**: 17-07-2024

**Prepared by**: KRITHIKRAJ

**Objective**

The objective of this report is to outline the database design and implementation details for the HIRECONNECT 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: [list attributes like \_id, name, email, password, createdAt, updatedAt]

**2. Posts**

- Attributes: [list attributes like \_id, title, content, author (references User), createdAt, updatedAt]

**3. Comments**

- Attributes: [list attributes like \_id, text, post (references Post), author (references User), createdAt, updatedAt]

**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,

password: String,

createdAt: Date,

updatedAt: Date

}

```

2. Collection: posts

- Schema:

```

{

\_id: ObjectId,

title: String,

content: String,

author: ObjectId (references users),

createdAt: Date,

updatedAt: Date

}

```

3. Collection: comments

- Schema:

```

{

\_id: ObjectId,

text: String,

post: ObjectId (references posts),

author: ObjectId (references users),

createdAt: Date,

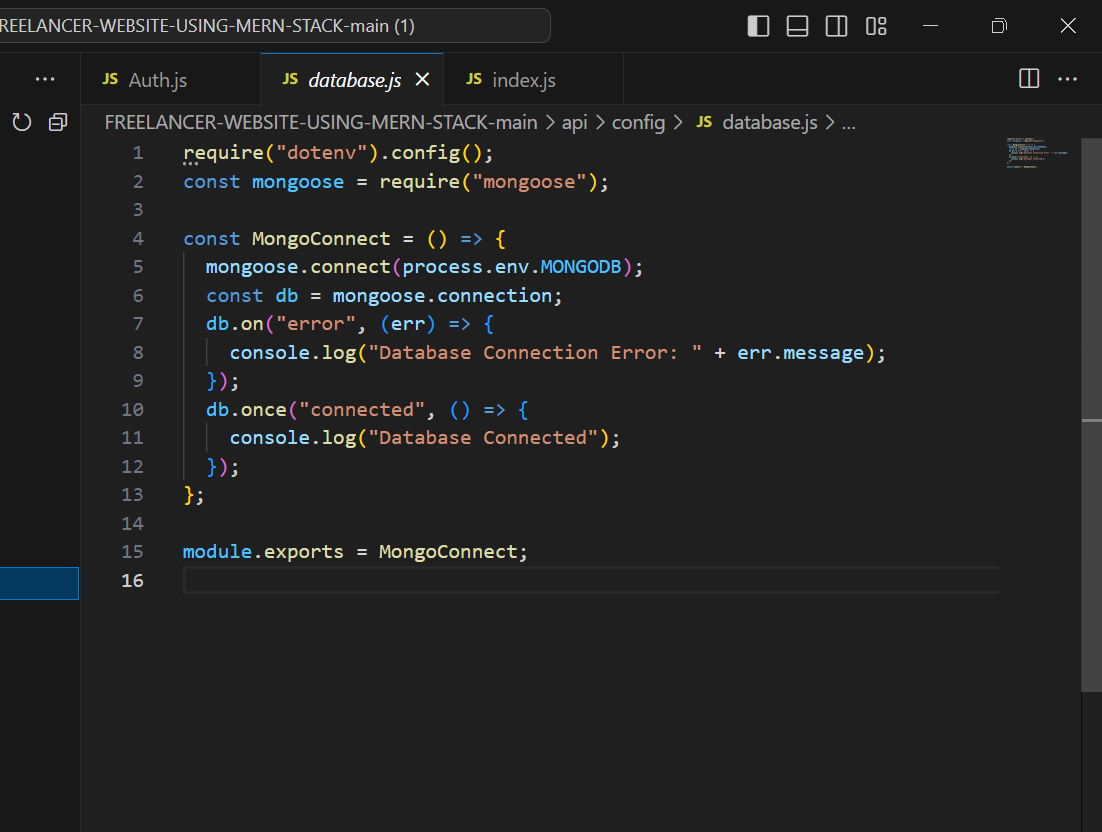
updatedAt: Date

}

```

**Integration with Backend**

* Database connection:



* The backend APIs interact with MongoDB using Mongoose ODM Key interactions include:
  + **User Management:** CRUD operations for users.
  + **Post Management**: CRUD operations for posts, with user authentication.
  + **Comment Management:** CRUD operations for comments associated with posts.
  + **Authentication and Authorization**: Handle user login and protect routes using JWT.
  + **Data Management**: Manage other data models with similar CRUD operations.
  + **Validation and Error Handling**: Use Mongoose schemas for validation and handle errors consistently.