**Full Stack Development with MERN**

**Database Design and Development Report**

|  |  |
| --- | --- |
| Date | 19 July 2024 |
| Team ID | SWTID1719933836 |
| Project Name | Connectify |
| Maximum Marks |  |

**Project Title**: Connectify

**Date**: 19 July 2024

**Prepared by**: Chidambaram Suresh

**Objective**

The objective of this report is to outline the database design and implementation details for the Connectify 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. User**

- Attributes: \_id, email, password, username, \_\_v, pfp

**2. Post**

- Attributes: \_id, userId, likes, caption, username, image, comments, date, \_\_v, likedBy

**3. Friends**

- Attributes: \_id, userId, friends, \_\_v

**3. Requests**

- Attributes: \_id, userId, requests, \_\_v

**3. Messages**

- Attributes: \_id, sender, receiver, message, timestamp, \_\_v

**Implement the Database using MongoDB**

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

Database Name: Connectify

1. Collection: usermodel

- Schema:

```

{

\_id: ObjectId,

email: String,

password: String,

username: String,

\_\_v: Int32,

pfp: String

}

```

2. Collection: postsmodel

- Schema:

```

{

\_id: ObjectId,

userId: ObjectId (reference users),

likes: Int32,

caption: String,

username: String,

image: String

comments: Array,

date: Date,

\_\_v: Int32,

likedBy: Array

}

```

3. Collection: friendsmodels

- Schema:

```

{

\_id: ObjectId,

userId: ObjectId (reference users),

friends: Array

\_\_v: Int32

}

```

4. Collection: requestsmodel

- Schema:

```

{

\_id: ObjectId,

userId: ObjectId (reference users),

requests: Array

\_\_v: Int32

}

```

5. Collection: messagesmodel

- Schema:

```

{

\_id: ObjectId,

sender: ObjectId (reference users),

receiver: ObjectId (reference users),

message: String,

timestamp: Date,

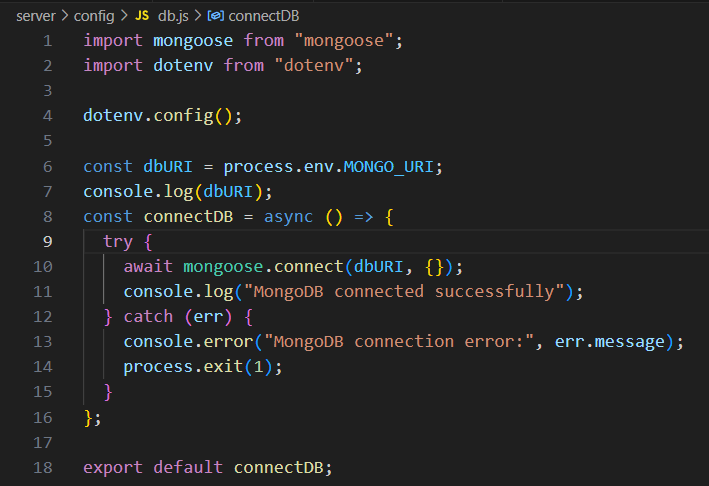
\_\_v: Int32

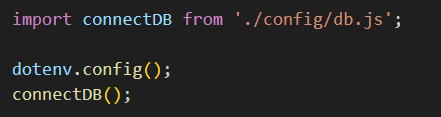
}

```

**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.
  + Input Validation: Using built-in validation rules.
  + Population: It is a feature to reference documents in other collections.