**Database Design and Development Report**

|  |  |
| --- | --- |
| Date | 11 April 2025 |
| Team ID | SWTID1743172790 |
| Project Name | Social Media Web App |
| Maximum Marks |  |

**Project Title**: Stock Trading Web App

**Date**: 11/04/25

**Prepared by**: Komal Baid, Shreeya Joshi, Shreya Saniya, Vanshika Jain

**Objective**

The objective of this report is to outline the database schema design and implementation details for the **Social Media Web App**, including the structure of data models, relationships among entities, and MongoDB integration using Mongoose.

**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:** username, email, password, profilePicture, bio, followers, following, savedPosts, createdAt

**2. Posts**

* **Attributes:** user, image, caption, likes, comments, createdAt

**3. Comments**

* **Attributes:** post, user, text, createdAt

**4. Messages**

* **Attributes:** sender, receiver, text, createdAt

**Implement the Database using MongoDB**

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

Database Name: Sphere App

const mongoose = require('mongoose');

const bcrypt = require('bcryptjs');

const UserSchema = new mongoose.Schema({

username: {

type: String,

required: true,

unique: true,

},

email: {

type: String,

required: true,

unique: true,

},

password: {

type: String,

required: true,

},

profilePic: {

type: String,

default: '',

},

posts: {

type: Array,

default: [],

},

followers: {

type: Array,

default: [],

},

following: {

type: Array,

default: [],

},

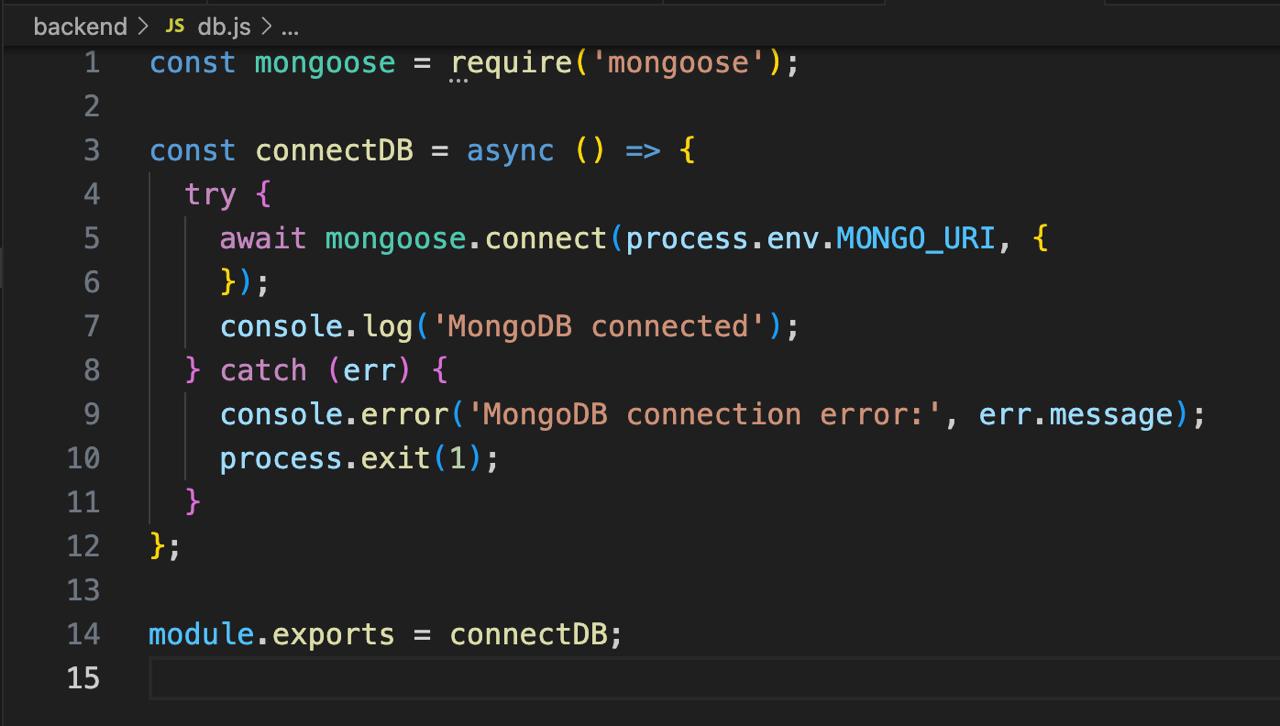
}, { timestamps: true });

const User = mongoose.model('User', UserSchema);

module.exports = User;

**Integration with Backend**

Connection established using **Mongoose** in the server.js or db.js file.



**Backend API Interactions with MongoDB**

* **User Management:**
  + Register/Login users
  + Update profile, followers, and saved posts
  + Password hashing using bcrypt
* **Post Management:**
  + Create, retrieve, delete, and like/unlike posts
  + Upload images (Firebase or AWS S3)
* **Comment Management:**
  + Add comments to posts
  + Fetch all comments related to a post
* **Messaging:**
  + Real-time communication using Socket.io
  + Store messages in MongoDB with timestamps