**database schema for a microblogging app using MongoDB. In MongoDB, you organize data into collections instead of tables. Each collection contains documents in BSON (Binary JSON) format.**

* **Users Collection**:

db.createCollection("users", {

validator: {

$jsonSchema: {

bsonType: "object",

required: ["username", "email", "password"],

properties: {

username: {

bsonType: "string",

description: "Username of the user"

},

email: {

bsonType: "string",

description: "Email address of the user"

},

password: {

bsonType: "string",

description: "Hashed password of the user"

},

bio: {

bsonType: "string",

description: "User's biography"

},

profile\_picture: {

bsonType: "string",

description: "URL of user's profile picture"

}

}

}

}

})

* **Posts Collection**:

db.createCollection("posts", {

validator: {

$jsonSchema: {

bsonType: "object",

required: ["user\_id", "content"],

properties: {

user\_id: {

bsonType: "objectId",

description: "User ID of the post creator"

},

content: {

bsonType: "string",

description: "Content of the post"

},

created\_at: {

bsonType: "date",

description: "Date when the post was created"

},

updated\_at: {

bsonType: "date",

description: "Date when the post was last updated"

}

}

}

}

})

* **Comments Collection**:

db.createCollection("comments", {

validator: {

$jsonSchema: {

bsonType: "object",

required: ["user\_id", "post\_id", "content"],

properties: {

user\_id: {

bsonType: "objectId",

description: "User ID of the commenter"

},

post\_id: {

bsonType: "objectId",

description: "Post ID of the commented post"

},

content: {

bsonType: "string",

description: "Content of the comment"

},

created\_at: {

bsonType: "date",

description: "Date when the comment was created"

}

}

}

}

})

* **Likes Collection**:

db.createCollection("likes", {

validator: {

$jsonSchema: {

bsonType: "object",

required: ["user\_id", "post\_id"],

properties: {

user\_id: {

bsonType: "objectId",

description: "User ID of the user who liked the post"

},

post\_id: {

bsonType: "objectId",

description: "Post ID of the liked post"

},

created\_at: {

bsonType: "date",

description: "Date when the like was created"

}

}

}

}

})

* **Followers Collection**:

db.createCollection("followers", {

validator: {

$jsonSchema: {

bsonType: "object",

required: ["follower\_id", "following\_id"],

properties: {

follower\_id: {

bsonType: "objectId",

description: "User ID of the follower"

},

following\_id: {

bsonType: "objectId",

description: "User ID of the user being followed"

},

created\_at: {

bsonType: "date",

description: "Date when the follow relationship was established"

}

}

}

}

})