**Project Name: ToDo list**

**Database Schema Design and Code:**

**Db Collections**:

**Users:**

* \_id (Primary Key)
* username
* password

**Tasks:**

* \_id (Primary Key)
* title
* description
* priority

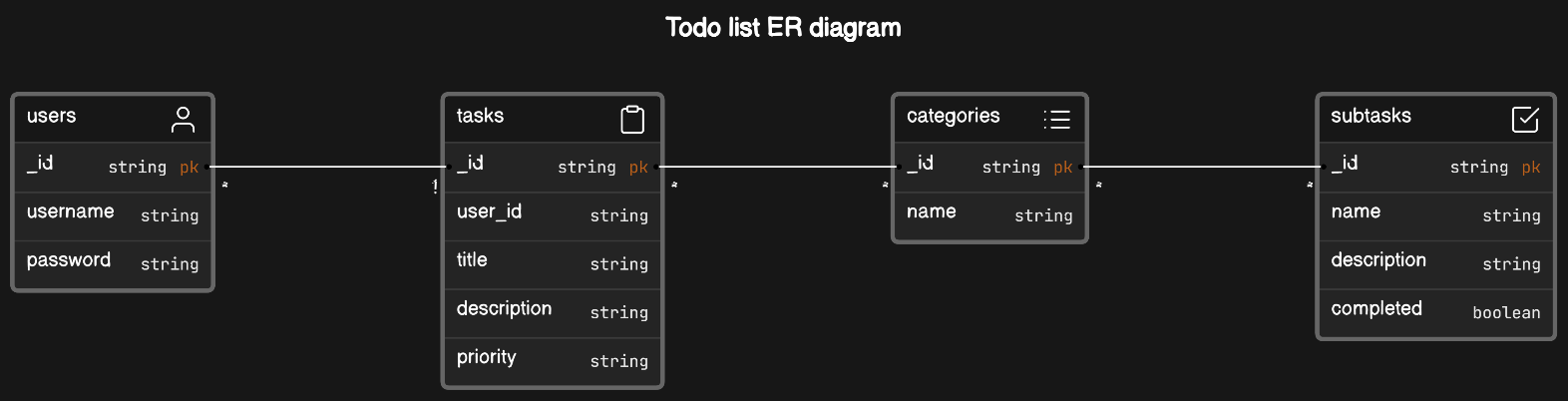
**Categories:**

* \_id (Primary Key)
* category\_name

**Subtasks:**

* \_id (Primary Key)
* description
* completed

**Entity-Relationship (ER) diagram:**



**Example of db schema in JSON format:**

**Users collection:**

{

"\_id": ObjectId("609e8d1899d4f3044c3c426b"),

"username": "user123",

"password": "hashed\_password"

}

**Task collection:**

{

"\_id": ObjectId("609e8d1899d4f3044c3c426c"),

"title": "Complete Project Proposal",

"description": "Write a detailed proposal for the upcoming project.",

"priority": "High"

}

**Categories collection:**

{

"\_id": ObjectId("609e8d1899d4f3044c3c426d"),

"category\_name": "Work"

}

**Subtask collection:**

{

"\_id": ObjectId("609e8d1899d4f3044c3c426e"),

"description": "Research project requirements",

"completed": false

}

**DB Schema Code:**

const mongoose = require('mongoose');

mongoose.connect('mongodb://localhost/todo\_list', { useNewUrlParser: true, useUnifiedTopology: true });

const db = mongoose.connection;

const userSchema = new mongoose.Schema({

username: String,

password: String

});

const taskSchema = new mongoose.Schema({

user\_id: { type: mongoose.Schema.Types.ObjectId, ref: 'User' },

title: String,

description: String,

priority: String

});

const categorySchema = new mongoose.Schema({

category\_name: String

});

const subtaskSchema = new mongoose.Schema({

task\_id: { type: mongoose.Schema.Types.ObjectId, ref: 'Task' },

description: String,

completed: Boolean

});

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

const Task = mongoose.model('Task', task);

const Category = mongoose.model('Category', category);

const Subtask = mongoose.model('Subtask', subtask);

db.once('open', async () => {

await User.create([

{ username: 'user1', password: 'password1' },

{ username: 'user2', password: 'password2' }

]);

await Task.create([

{ user\_id: user1.\_id, title: 'Task 1', description: 'Description of Task 1', priority: 'High' },

{ user\_id: user1.\_id, title: 'Task 2', description: 'Description of Task 2', priority: 'Medium' },

{ user\_id: user2.\_id, title: 'Task 3', description: 'Description of Task 3', priority: 'Low' }

]);

await Category.create([

{ category\_name: 'Work' },

{ category\_name: 'Personal' },

{ category\_name: 'Shopping' }

]);

await Subtask.create([

{ task\_id: task1.\_id, description: 'Subtask 1 for Task 1', completed: false },

{ task\_id: task1.\_id, description: 'Subtask 2 for Task 1', completed: true },

{ task\_id: task2.\_id, description: 'Subtask 1 for Task 2', completed: false }

]);

console.log('Sample data inserted successfully.');

});

db.on('error', console.error.bind(console, 'MongoDB connection error:'));