Notes

Task 1: Database Selection and Justification

- **Scenario:** You are tasked with choosing a database for a new application. The application is a social media platform with the following characteristics:
    - Users will have profiles with varying amounts of information.
    - Users will create posts that can contain text, images, and videos.
    - Posts can be liked and commented on.
    - The application requires a high degree of scalability to handle a potentially large user base.
    - Real-time updates on user activities and posts are preferred.
- **Task:**
    1. **Select a database:** Choose between a relational database (RDBMS) and a non-relational database (NoSQL).
    2. **Justify your choice:** Explain why you chose the specific type of database, referencing its key features, strengths, and weaknesses, in the context of the application's requirements.
    3. **Recommend a database:** If you choose NoSQL, recommend a specific database (e.g. MongoDB) and justify your choice.
    4. **Discuss potential drawbacks:** For your chosen database type/specific database, discuss potential drawbacks you might face and how you'd mitigate them.

Task 2: MongoDB Document Modeling with Mongoose

- **Scenario:** You are building a simple e-commerce platform using Node.js and MongoDB. You need to create a data model for products. Each product should have the following attributes:
  - `name`: String (required)
  - `description`: String
  - `price`: Number (required, must be greater than 0)
  - `category`: String (should belong to 'Electronics', 'Clothing', or 'Books' , using enums)
  - `createdAt`: Date (automatically generated with a default value of now)
  - `updatedAt`: Date (automatically updated when the document is modified)
- **Task:**
  1. **Define a Mongoose schema:** Write the Mongoose schema definition for the product using all the necessary validation, enums and default values.
  2. **Create a model:** Create a Mongoose model using the schema.

Task 3: MongoDB Compass and Querying

- **Scenario:** You have a MongoDB database with a collection named "users". Each user document has the following fields: `name`, `email`, `age`, and `location`
- **Task:**
  1. **Describe using compass:**
     - How you could use MongoDB Compass to view the list of all documents in the "users" collection.
     - How you could use MongoDB Compass to filter the data and get users whose age is greater than 25.
     - How you could use MongoDB Compass to export all the filtered user data into a JSON file.
  2. **Describe using a query:**

- Write the MongoDB query that would retrieve all the users with age greater than 25.