HTL LEONDING

Exercise: Blog

Introduction

The goal of this exercise is the creation of a blog-like system based on MongoDB. The system should allow the user or creator to write new entries.
But instead of just plain text and some predefined fields like 'title' or 'date' the system should also support links, images and some other data fitting to your topic of choice. (Plain text should be an option also).
That way it is possible to save the data in some fixed format/datatype and to search and analyse accordingly.
Additionally any reader should be able to comment on existing entries.

Database tasks

Realize following Tasks via MongoDB:

1. Creation of a MongoDB Database
2. Create following elements and think about the fields needed (tasks 3 & 4 give some hints).
   The choice between multiple collections and embedded collections is yours:
   - BlogUsers
   - BlogEntries
   - BlogCategories
   - Comments

3. You may choose your structure and the relations but have to follow your given/chosen design pattern.
4. Create at least 5 blog-users with following properties: username, name (firstname und lastname), email, password.
5. Create 5 or more blog-entries for at least 3 categories containting minimum these attributes each document:
   - title
   - author – Autor(s) of the blog entry, embedded or relation
   - description
   - creationDate
   - editDates – Array of date entries containing the edit timestamps
   - impressionCount – Long: Number of times a user visited the blog-entry
   - content – Your text, link, images and other content.
     Images should be stored base64-encoded, NOT as links.
   - commentsAllowed – May this entry be commented upon.
6. Create a schema vor the blog-entry (see task 5). It should cover the mandatory fields.
7. Add 4 comments to 3 different blog-entries (only on those that allow comments).

Queries

Queries may be integrated in the final implementation, or just executed on the mongosh.

- all blog-users where username & password match given values.
- all blog-entries being written by a certain blog-user (username).
- all blog-entries that does not contain ANY information in one of your additional fields (one of the additional fields you added to your content).
- all blog-entries where the entry has more than 1 image.
- all blog-entries where the entry contains image(s) and
- all blog-entries whose author either has a given lastname or has the value 'admin' but not 'Guest'.

- all blog-entries where the title is mentioned in the content as well.
- alle BlogUser sorted ascending by username.
- the newest (creationDate) 2 blog-entries.
- the second oldest blog-entry.
- all blog-entries created within the last week containing a link.
- the 2 newest blog-comments added to a given username's entries.

Changes

- Add a new author to an existing blog-entry or change the author, if a second author is not possible.
- Extend the newest blog-entry by an additional field 'hashtag' with content.
- Change the name of a blog-category.
- Add a blog-entry, if the entry already exists it should only be changed (upsert).
- Delete a blog-entry and its comments.

Index

Create an index to assure:

- that the username is unique within the blog-users.
- that the title and username combination of a blog-entry is unique.

Implementation

Create an application (programming language or type (web or otherwise) of your choice). You may reuse existing projects (or part of those) or get help from ChatGPT.
Focus of this exercise is the integration of MongoDB-Queries into the application.

- 1 'page' to display a blog-entry with all different data
- 1 'page' to add a new blog-entry.
- additional functions/pages/etc. if wanted/needed. (e.g. if you call your queries from the applicaton)

Realization, Hand-In

For groups of 2 or 3.

Create following artefacts and hand them in with moodle as a ZIP-file:

- Report of your doings (matching the template, PDF)
- Query Code samples and results of the queries
- Screenshots of your application

After handin is complete there will be a short review by your teachers with you showing following elements:

- Data model of your blog-entries and it's document(s)
- How will you distinguish and process the different blog-entries
- Show changes to the data (new entry/changes to an entry)
- Show and explain 2 queries chosen by the teachers

Grading

- M (25) Data model of the different kinds of blog-entries
- M (25) Queries
- (20) Updates, Index
- (30) Existence of an "UI" (functionality counts not quality)

Anything marked with a 'M' is needed to pass this assignment. Empty Hand-Ins, no visible work or no documentation will fail this assignment.