Logged Life

Database Design

Patrick Mesey

12/11/2022

OVERVIEW

The web application being put together will use MongoDB for the data management. This is a document based data storage and is best use for this type of application. The application will primarily use “cards” in which each posting will be fully self-sustaining data objects.

Data Specifications

The web application will utilize Mongoose as the object data modeling library. This will help with the schema and data manipulation in conjunction with MongoDB.

Cards

The cards that will be posted are post messages that use a Schema as follows:

```
const postSchema = mongoose.Schema({
    title: String,
    message: String,
    name: String,
    creator: String,
    tags: [String],
    selectedFile: String,
    likes: { type: [String], default: [] },
    createdAt: {
        type: Date,
        default: new Date(),
    },
})
```

User – All is required data

```
const userSchema = mongoose.Schema({
  name: { type: String, required:  true },
  email: { type: String, required: true },
  passwordHash: { type: String, required: true },
  id: { type: String },
});
```

Challenge

```
{
    name: String,
    id: Number,
    active: Boolean,
    points: Number,
    achieved: Boolean
}
```

Purpose, Implementation and Interaction

Cards

Purpose:

This is the main portion of the application and reason for building.

Implementation:

These cards will be displayed on a grid with a title, image, description, and the ability to like or delete the card. After a user has logged in, then the ability to create a card is allowed. User can then also delete cards of their own and like others cards.

Interaction:

Users can search for cards, like, or delete a card.

User

Purpose:

This is not a necessary feature, but in order to reduce free posting by anyone, it will help develop simple requirements for posting a card, deleting, or liking a card.

Implementation:

Logins will utilize an email and password, which is validated with the record from the database.

Interaction:

After clicking login, the user can then login to their account or create an account at first use.

STRETCH FEATURE

Challenge

Purpose:

This is a stretch feature for possible implementation if time or later. The intention is to draw the user into the application by completing challenges in which will reward status “titles” that they can display next to their name when creating posts.

Implementation:

Challenges will check for the requirement(s) of the active challenge. i.e: “Like a card”.

Interaction:

For the above example, when a user “likes” a card, the system will check for the active challenge and if that is an option to be considered. If it is, it will set the achieved to TRUE for the challenge. **Optional feature would be to allow the user to “Accept” challenges from a list. New challenges appear as the user completes a certain number of previous tier challenges.

Commenting

Purpose:

This is a stretch feature for possible implementation if time or later. The intention is to allow logged in users to comment on cards.

Implementation:

Once a user is logged in, they can comment on a card in which everyone can see.

Interaction:

When logged in the user can click a comment button and then add a comment to the card. Others that interact with that comment button will be able to see the list of comments.

Google OAuth

Purpose:

Allows user to choose to login with their google account instead of the standard email/password.

Implementation:

Add google authentication to the server and the client side.

Interaction:

The user can click google sign in, it will pop up their google sign in options and they can choose from one of their logins.