Tutorial 5

Server-side validations

There are many ways to performn server-side validations. In this tutorial, we will focus on using `mongodb` for validations. MongoDB has syntax for passing a validator schema to the collection on creation, which is then enforced. Any violation of the schema returns an error to the server, which is then propagated to the client.

It is entirely possible to manually perform these validations server-side, however it helps to have the database quickly and easily validate data for the purposes of this project.

For this tutorial, you are expected to validate that the `name` and `tweet` attributes both contain the same validation rules on the server as they do on the frontend. They must both be valid strings, but `tweet` has the additional restriction of being 1 <= length <= 280.

For more information, refer to mongodb schema validation documentation: https://www.mongodb.com/docs/manual/core/schema-validation/

After adding the validation schema to the collection, make sure to wrap the `insertOne` function calls, etc. with `try/catch` statements. `try/catch` statements catch any error that is called by code within the `try` statement, and is passed as a parameter to the `catch` statement. For example:

```
try {
    // code that throws an error
} catch (err) {
    console.error(err);
}
```

Note that with promises, you have to `await` them for errors to be thrown properly, otherwise they will silently error asynchronously in the background.

```
// Do this
try {
    await collection.find()
}

// Don't do this
try {
    collection.find();
}
```