User Database Localhost Test

Here is what I did to test the user database on localhost. This test allows us to view our user database in the MongoDB shell locally.

Download and install MongoDB 2.6.10 on your computer

1. Download MongoDB 2.6.10 from https://www.mongodb.org/dl/win32/x86_64
   - Uninstall other versions if necessary and delete the MongoDB folder from Program FIles
   - Use ctrl + f function and type "2.6.10" to find version 2.6.10 faster
   - Download the 2008plus-2.6.10 signed version
2. For installation type, choose Complete
   - There should now be a folder in C:\Program Files called MongoDB 2.6 Standard
3. Rename the MongoDB 2.6 Standard folder to MongoDB

Configuring MongoDB

1. Create a folder named **data** and a folder named **log** in C:\Program Files\MongoDB
2. Create a folder named **db** in the data folder. The db folder will contain the databases we create
3. Open Command Prompt (Windows) **as administrator** and navigate to the **bin** folder in MongoDB
4. Next, configure the data & log folder with this command:
   mongod --directoryperdb --dbpath "C:\Program Files\MongoDB\data\db"
   --logpath "C:\Program Files\MongoDB\log\mongo.log" --logappend --rest --install
5. Start MongoDB as a service by entering net start MongoDB. It should be successful. This is just to check if step 4 was successful
6. Run net stop MongoDB to stop the service

Setting up and running Mongo Shell

Next, we'll set up the Mongo shell. This allows us to view data from our user database, and later, the image database. Perform the following steps from now on to view our databases.

1. Run mongod.exe with this command:

   mongod.exe --dbpath="C:\Program Files\MongoDB\data\db"

   This was where we set up our database folder during the configuration process

   - If this command was successful, the command prompt should now say "waiting for connections on port 27017"

2. Open another command prompt as administrator

3. In the new command prompt, navigate to MongoDB\bin and enter the command mongo.exe to run the mongo shell

4. Check the other command prompt running mongod.exe. It should now say that there is 1 connection

5. We can now run MongoDB commands in this shell to view our databases

Testing our user database

We should now have mongod.exe and mongo.exe running on two separate command prompts. Now we can test the user database. I wrote a test Account document in RegisteredUsersCollection/Account.js which adds a user to our registered user collection. The test account document contains a test email, username, and password.
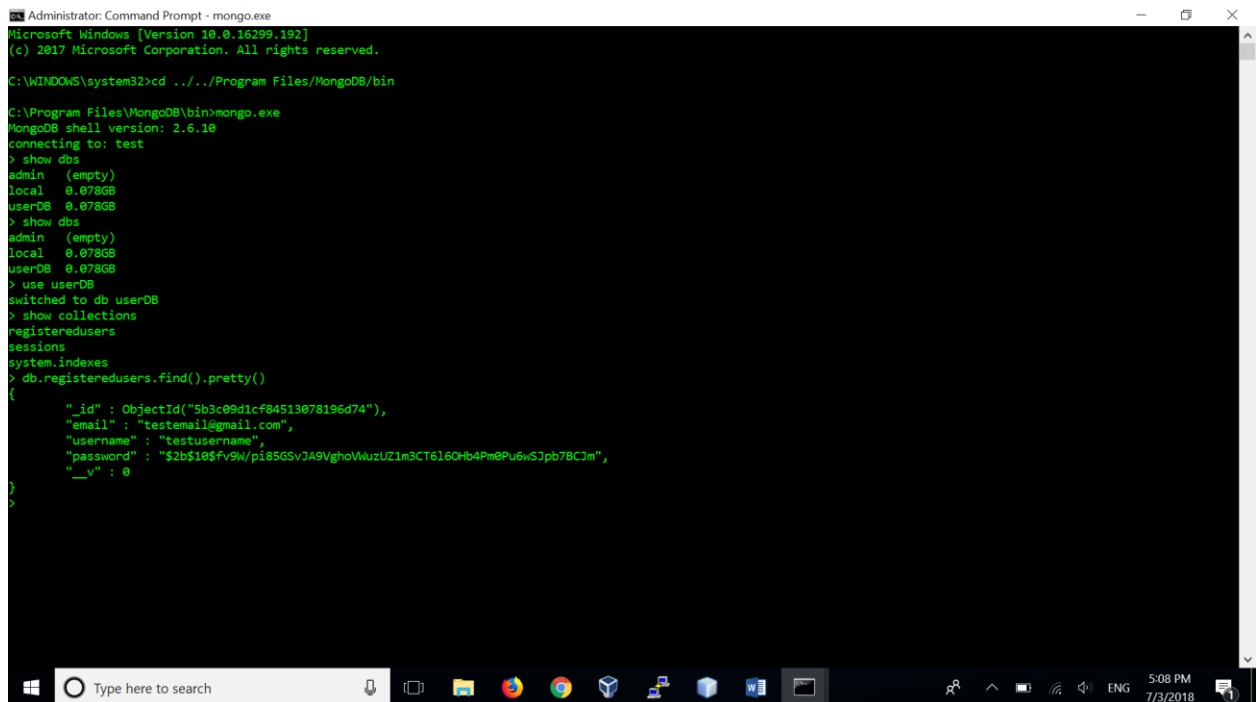
Email: testemail@gmail.com

Username: testusername

Password: testpassword

The following test allows us to view this test document in the MongoDB shell.

1. Open a third command prompt and navigate to the myapp folder in our project.

2. Mongod.exe should now say that another connection has been added

3. In the Mongo shell (mongo.exe), type **show dbs** to view all databases. The database userDB should be there.

    - This is because I set our mongoose connection to localhost/userDB in our Server.js file, with localhost being the db folder in C:\Program Files\MongoDB\data.

4. To enter userDB, type use userDB in the Mongo shell

5. To view collections, type show collections. The collections registeredusers and sessions should be there.

6. To view all account documents, type db.registeredusers.find() to show all accounts in the registered users collection

    a) To make the output more readable, add the pretty() command after find(), like this: db.registeredusers.find().pretty()

    b) This shows the information from the test account document from Account.js. Notice that the sample password has been hashed.

7. Here is a screenshot of the previous steps in mongo.exe

With this, the user database test is completed. To summarize, from now on, we'll have to open two Command Prompts as administrator. One to run mongod.exe –dbpath= path to db

folder and the other to run mongo.exe to view our databases locally. We only have to run node server.js once to create the database locally. To test on the server, I think we'd have to change our mongoose connection to MongoDB on the server