Project Part 1

Austin Lee, Jacob Mulroy, Conner Hundt, D’Angelo Abell

Main Process

The virtual box program with MongoDB was already installed, along with a copy of the final project from the previous semester. We use this project as a base to build off, as it already had much of the setup and functionality required for this assignment.

To create the hospital database and its collections, we opened a terminal and used the following commands:

- mongo *Init the mongoDB service*
- use Students *Create and access the Students database*
- db.dropDatabase() *This was to remove the existing data from the project*
- use Hospital *Created a new database for the project*
- db.createCollection(“info1”)
- db.createCollection(“info2”)

Before launching the program, we did have manually create a **public** folder in the meteor directory and put the dateSet.csv file there for access. Using the **meteor** command in a new terminal, we then launched the website and made our changes to the .html file, along with the client and server-side javascript files. This involved adding numerous controls and forms to accept data, along with a button to process the individual upload and a file-upload button to handle the bulk upload:

After inserting some data into the colllections, we were able to add an extra column to the collections by using the following mongoDB commands in the terminal:

- mongo
- use Students
- db.info2.update({“age”:{$eq: 21-30}},{$set{“Patient Type”: “Young”}},{multi: true})

Errors

While the individual upload worked perfectly, we encountered our main problem when attempting to upload the .csv file in bulk. Using the command:

- **mongoimport --type csv -d Hospital -c info2--headerline --drop products.csv**

in a regular terminal, we were able to successfully insert the bulk file into the info2 collection by hand. However, trying to upload it through the html interface proved much more difficult. We attempted several different ways, including accessing the terminal through mongoDB commands, and attempting to upload and parse the file manually through the client-side javascript. Unfortunately, none of these methods worked sucessfully. The internet console displayed an error where we were unable to gain access to the data file, as it could not resolve the filename from the upload button. As of submission, we were unable to fix the issue.