Name: Cameron L'Ecuyer  
Class ID: 17  
Email: [cjlth5@mail.umkc.edu](mailto:cjlth5@mail.umkc.edu)  
Professor: Yugyung Lee  
[MyGitHub](https://github.com/camlecuyer/CS5551_11-1_LabAssignments)

Technical Partner: Name: Sneha Mishra  
Class ID: 21  
Email: [smccr@mail.umkc.edu](mailto:smccr@mail.umkc.edu)  
[GitHub](https://github.com/SnehaMishra28/CS5551_SnehaMishra_labassignments/wiki/CS5551_(11_1)_LabAssignment_lab%235)

**Objective**

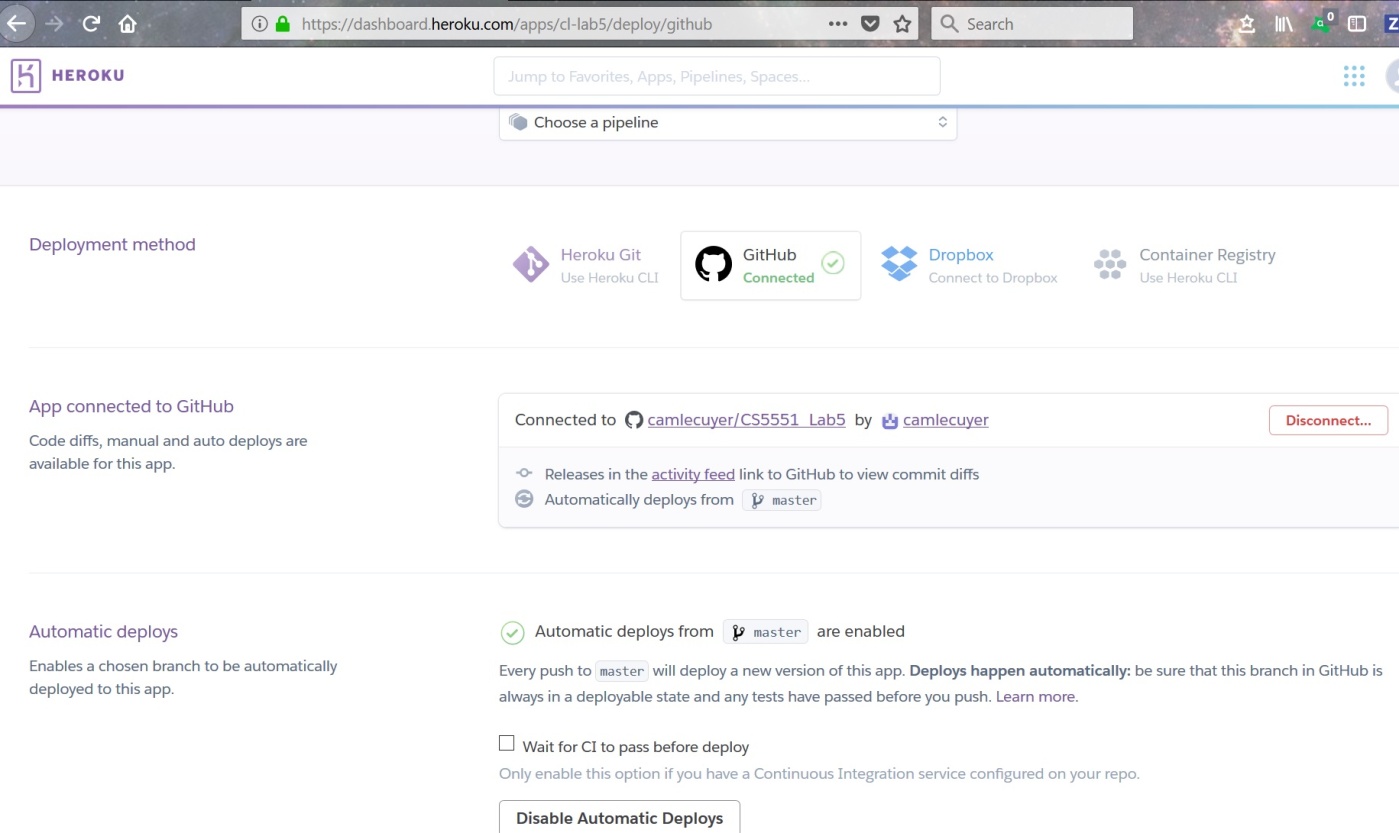
Create a MEAN Stack application using NodeJS, MongoDB, and Heroku, with a login and register page, a home page, and a NPM module that is relevant to our project.

**Features**

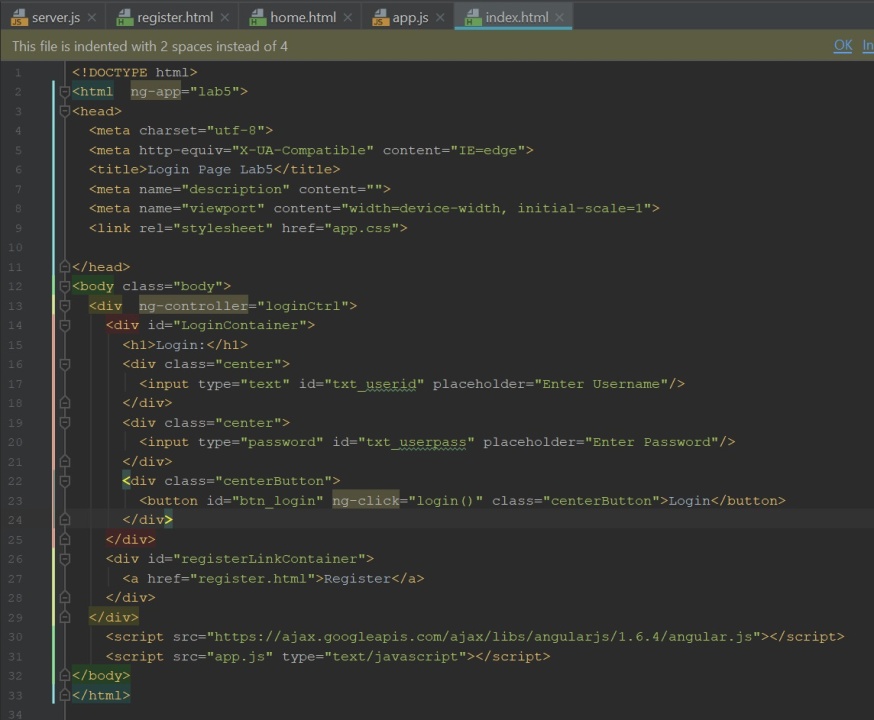
* NodeJS application set up in a client/server model
* MongoDB connected to NodeJS
* Heroku cloud deployment of application
* NPM module associated with semester project

**Steps**

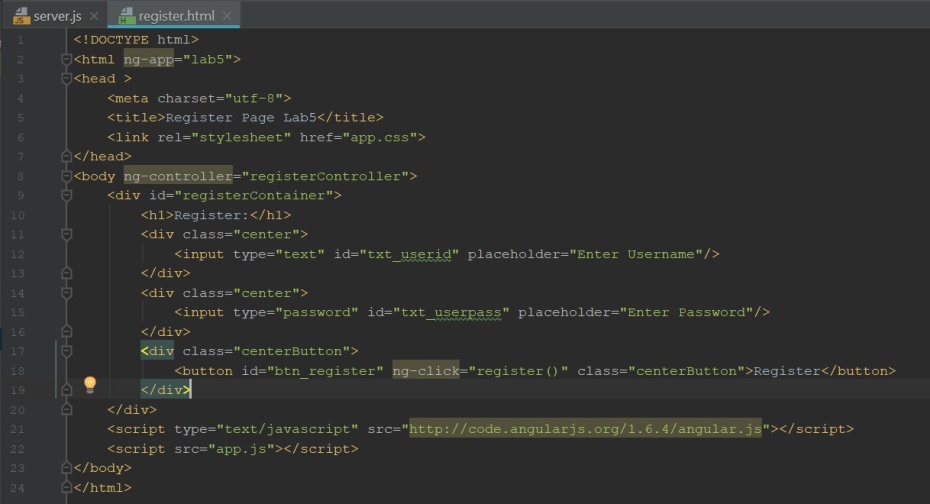
**Step 1 - Create and initialize a Git repository for Heroku, create a NodeJS project, and create a MongoDB database**

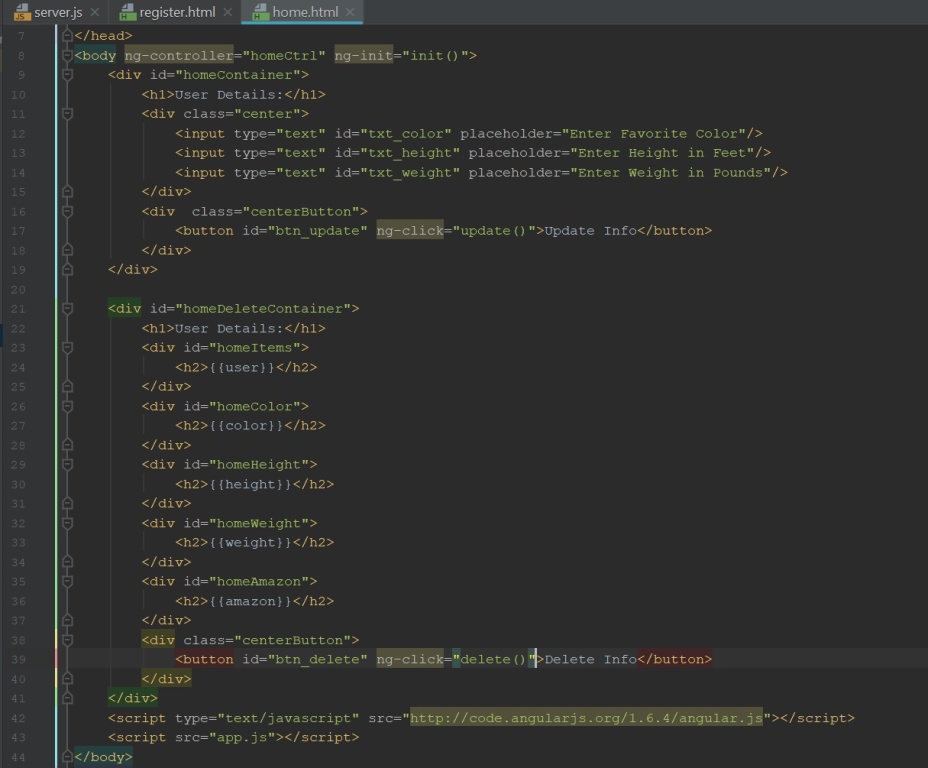
This image shows that Heroku is connected to one of my GitHub repositories, and uses automatic deployment when that repository updates 

**Step 2 - Create a login page, register page, and home page**

This image displays some of the HTML for the login page 

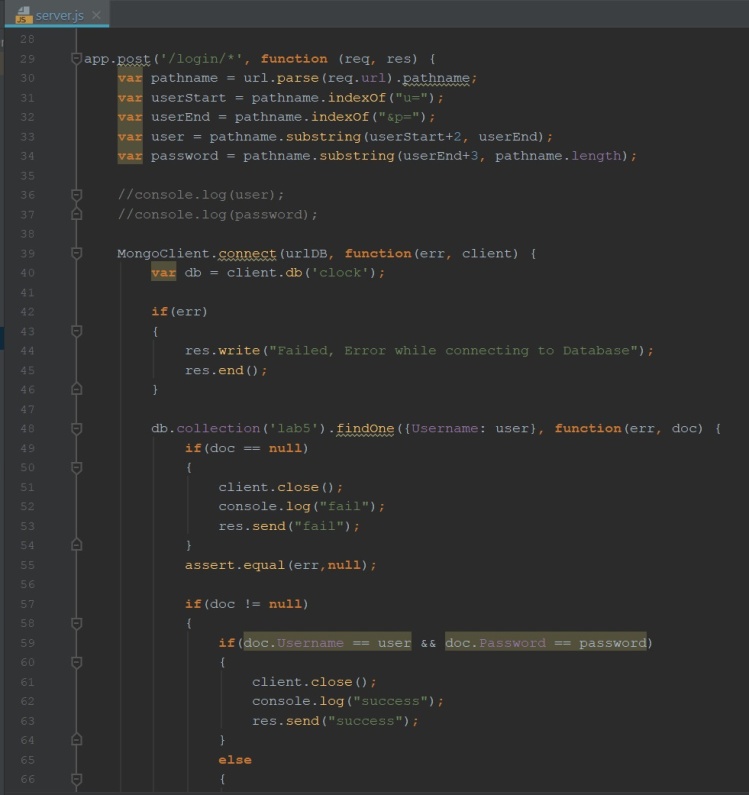
[Full Login Page HTML Code](https://github.com/camlecuyer/CS5551_11-1_LabAssignments/blob/master/Lab-5/Source/public/index.html)

This image displays the HTML for the register page 

This image displays some of the HTML for the home page 

[Full Home Page HTML Code](https://github.com/camlecuyer/CS5551_11-1_LabAssignments/blob/master/Lab-5/Source/public/home.html)

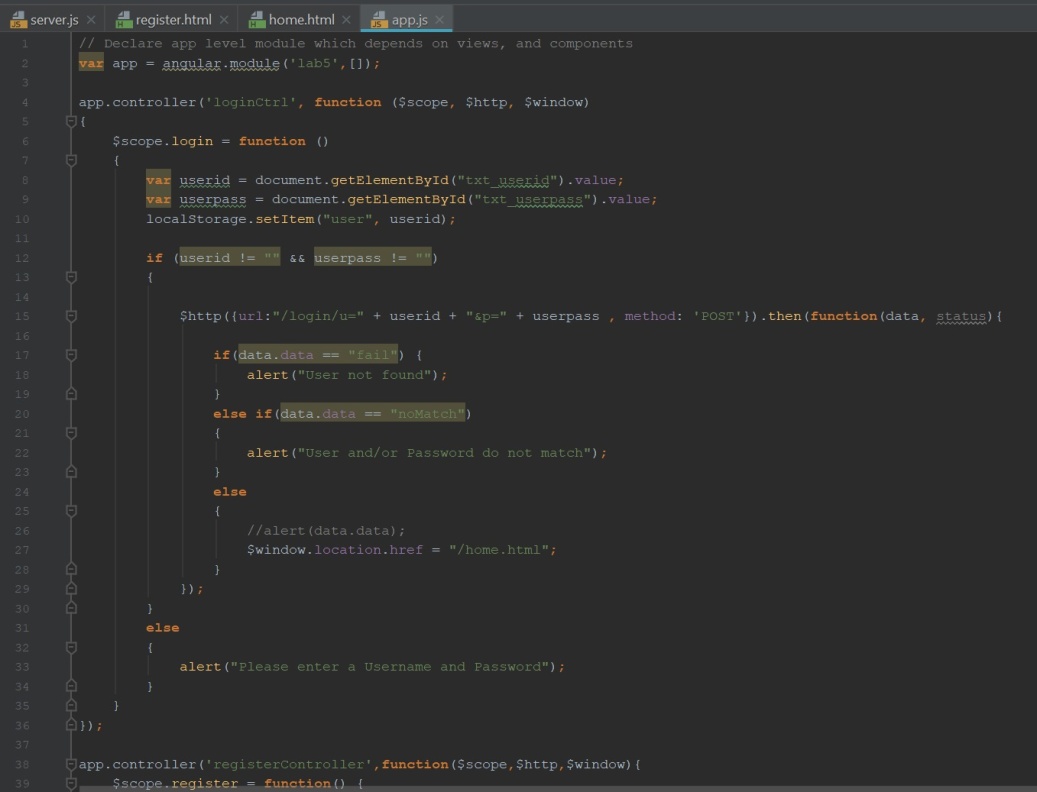
**Step 3 - Create the server JS file and add Procfile for Heroku deployment**

This image displays some of the JS code for the server side of the application, this code handles talking to MongoDB and creating, updating, reading, and deleting data from the MongoDB database, it also deals with the module that is related to our project 

Inside the update post call one of the functions referenced is the Geo-Amazon module, this module can return the correct Amazon URL based on the country code, it would be useful to our project to know the correct Amazon if it was used outside the United States.

[Full Server JS Code](https://github.com/camlecuyer/CS5551_11-1_LabAssignments/blob/master/Lab-5/Source/server.js)

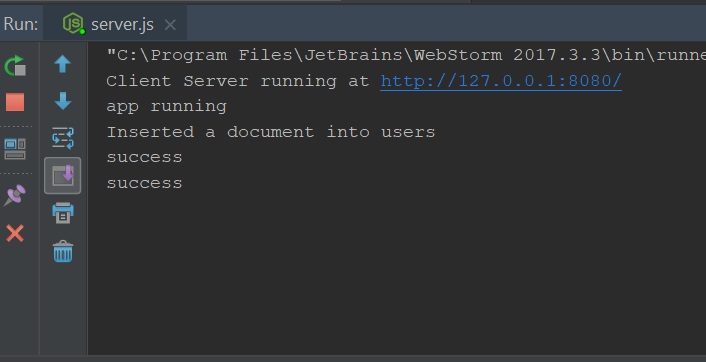
**Step 4 - Add the Angular code to connect the front end and server code, and CSS files to style the web pages**

This image displays some of the code for app JS 

[Full App JS Code](https://github.com/camlecuyer/CS5551_11-1_LabAssignments/blob/master/Lab-5/Source/public/app.js)

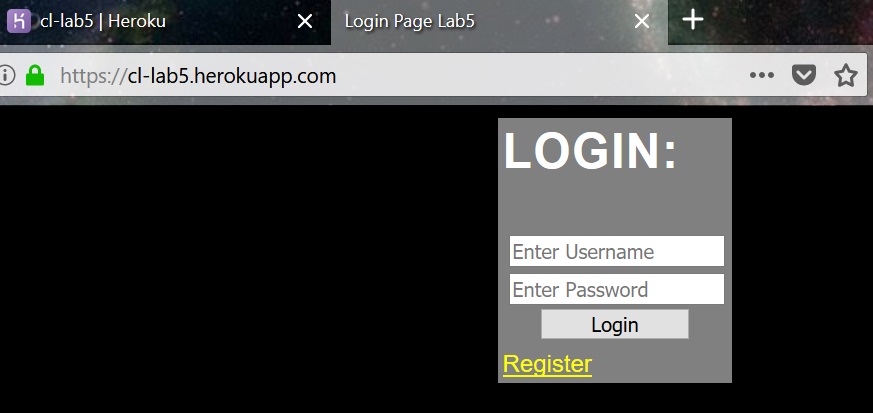
[Full CSS Code](https://github.com/camlecuyer/CS5551_11-1_LabAssignments/blob/master/Lab-5/Source/public/app.css)

**Step 5 - Run and test the server before deployment**

This image shows the server running, and successfully performing several actions 

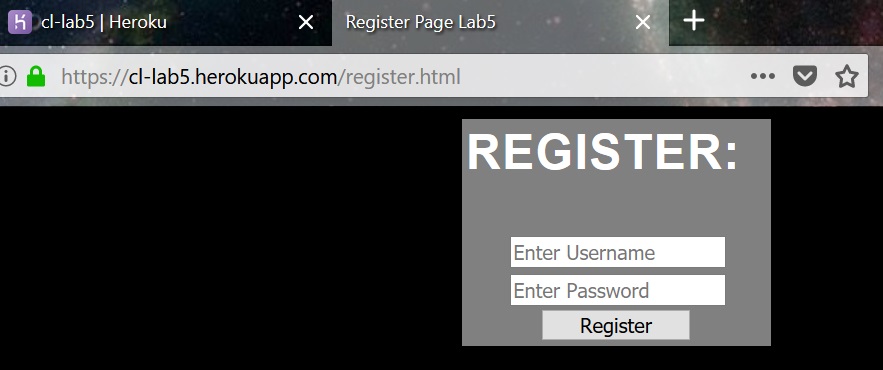
**Results**

**Login Page**

This is the Login page of the app, there is a link to the Register page, for login to be successful, a user must register successfully with the MongoDB database, and the login must successfully find the user and match their password 

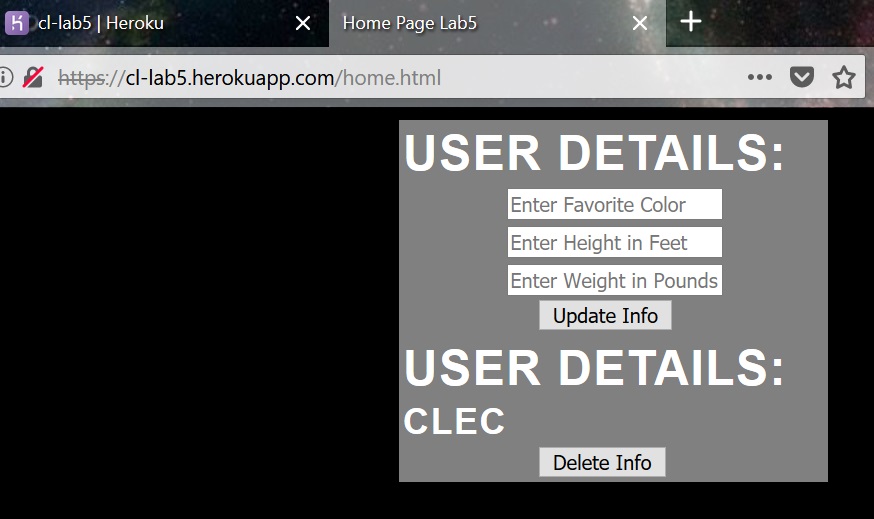
**Register Page**

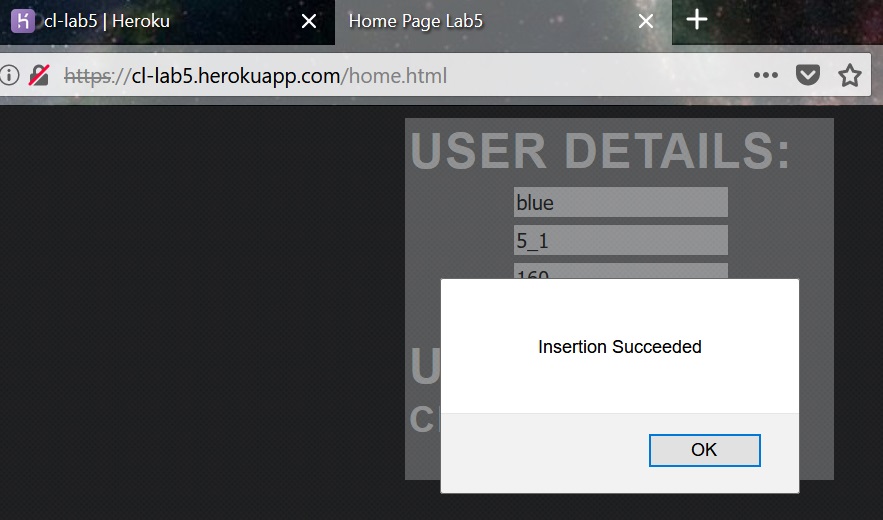
This is the Register page, once successfully registered, the page will redirect back to the Login page

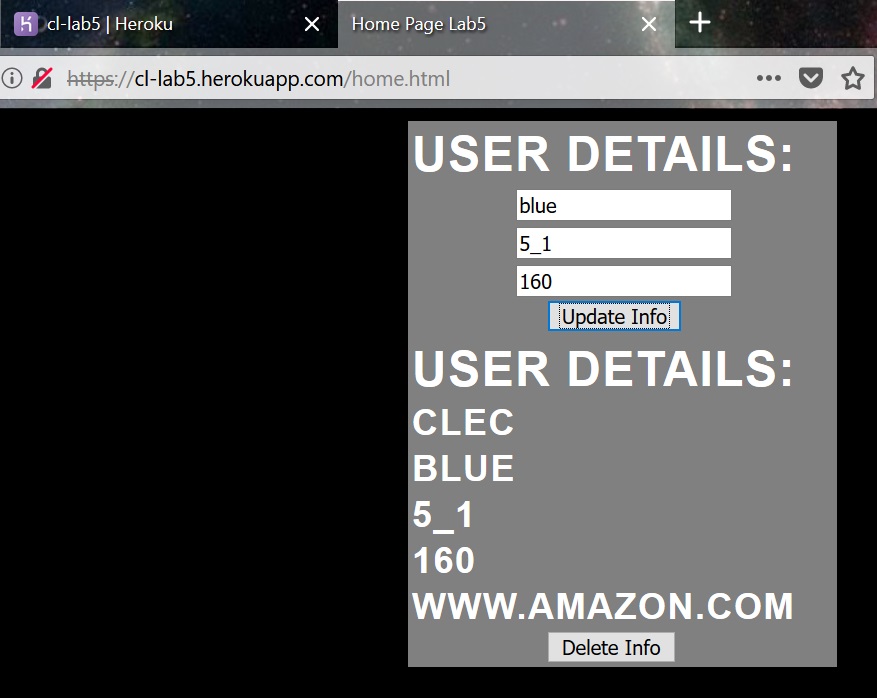


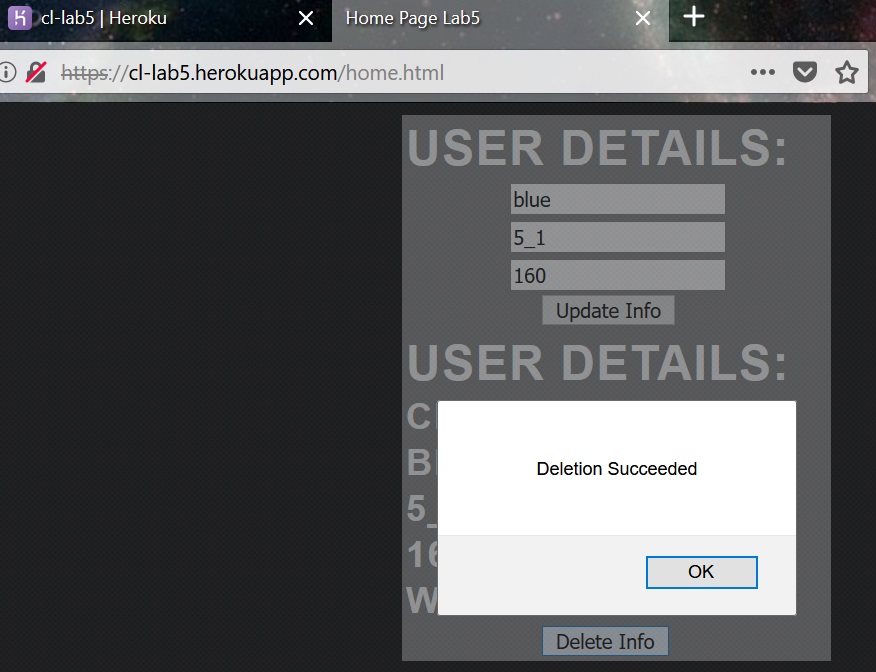
**Home Page**

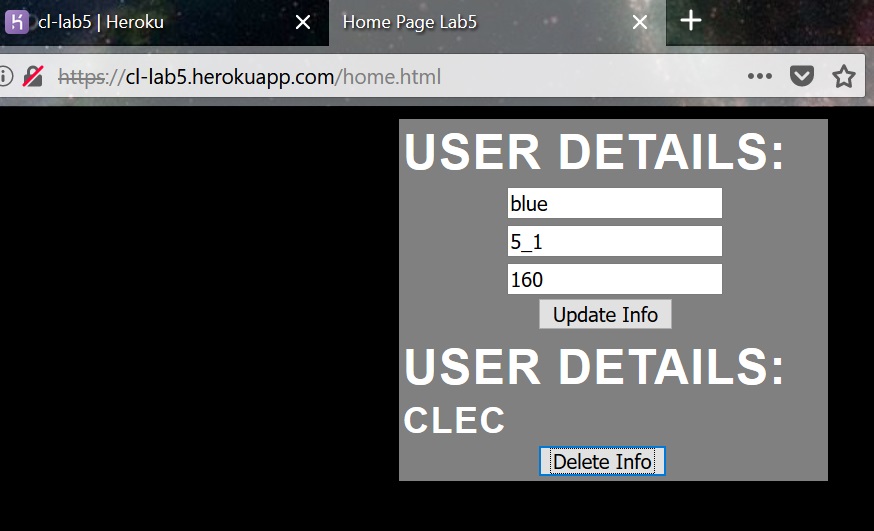
This is the Home page, it displays the Username of the person by using localStorage when moving between pages



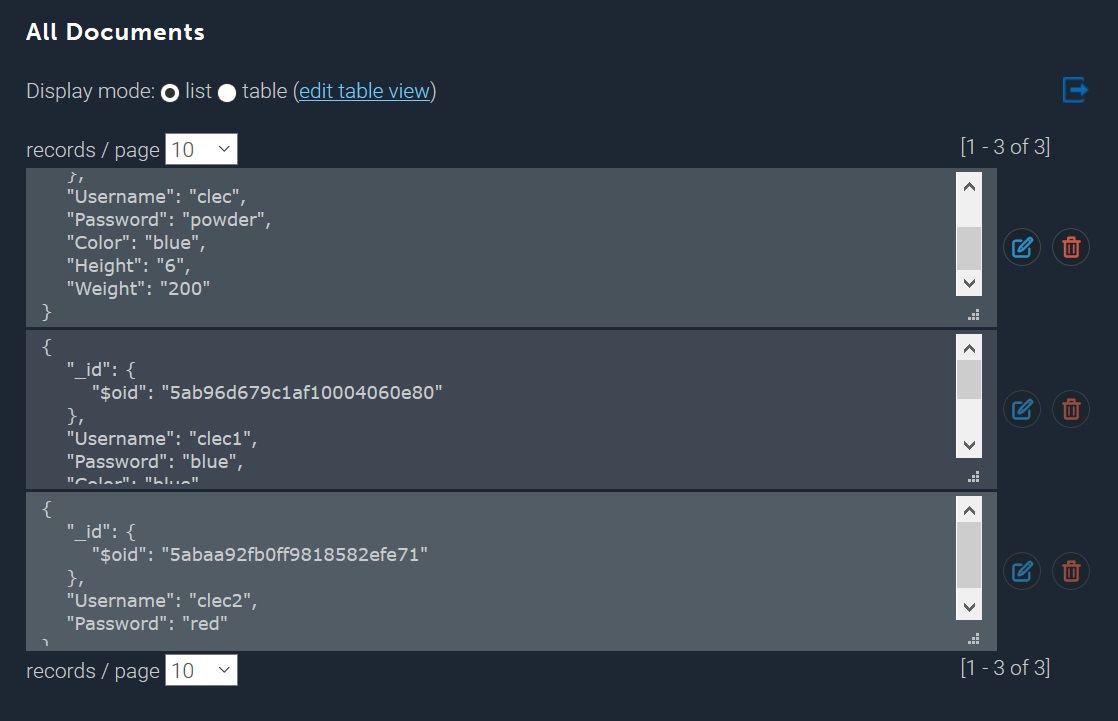
This is the Home page displaying an update successful message, once the message is cleared the page shows the updated user information, which is the following image 



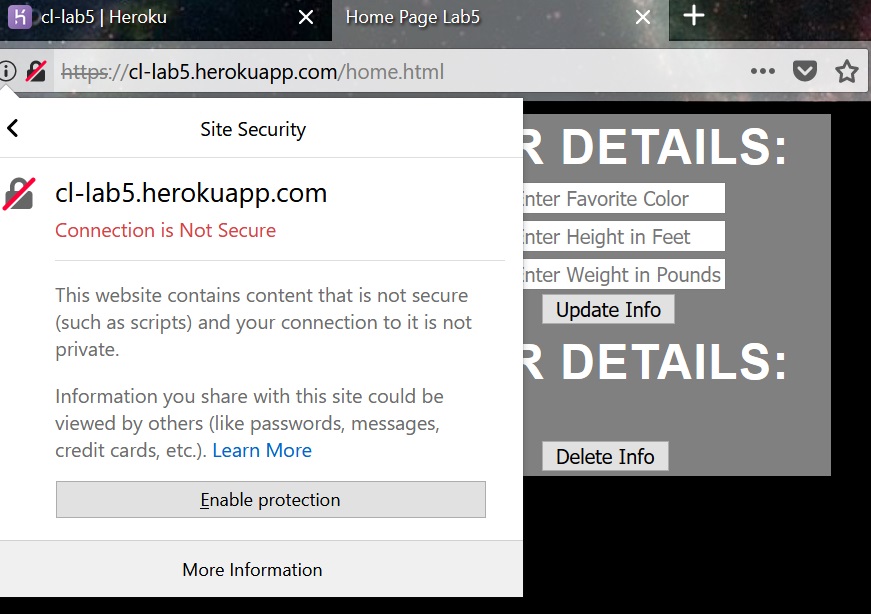
If the user clicks the delete button, the user first gets a message telling if the deletion was successful, and the page no longer displays the updated info, which is the following image 



**MongoDB**

This image shows several users, some with added info, others without the added info, as it was either deleted, or the account is new 

**Issues**

Firefox blocked some of the code from running due to it being mixed types, but if secure connection was disabled, it worked again 

**References**

Geo-Amazon NPM: <https://www.npmjs.com/package/geo-amazon?activeTab=readme>