Client-Side Web Technologies – Homework Assignment 5

Due: March 6th @ 11:59 PM

Value: 160 points (16% of total grade)

Overview

In this assignment you will be creating a simple contact management application using AngularJS. To accomplish this we will be making use of several 3rd party modules that we will include in our application via Bower.

I have created seed project for you that has the following:

- The npm package.json file that will install Bower and the lightweight HTTP server we used in class
- The bower.json file that has all the packages you will need already configured.
- An index.html file that has the necessary script tags for the packages you will use
- An app.js file that sets up some module configuration options and sets up routing
- Two empty view templates that correspond to the routes

This structure is contained in the seed.zip archive.

Similar to the GitHub repo code we looked at in class, you will just need to run `npm install` inside the directory to get up and running. To start the HTTP server, run `npm start` and navigate to http://127.0.0.1:8000 in your browser.

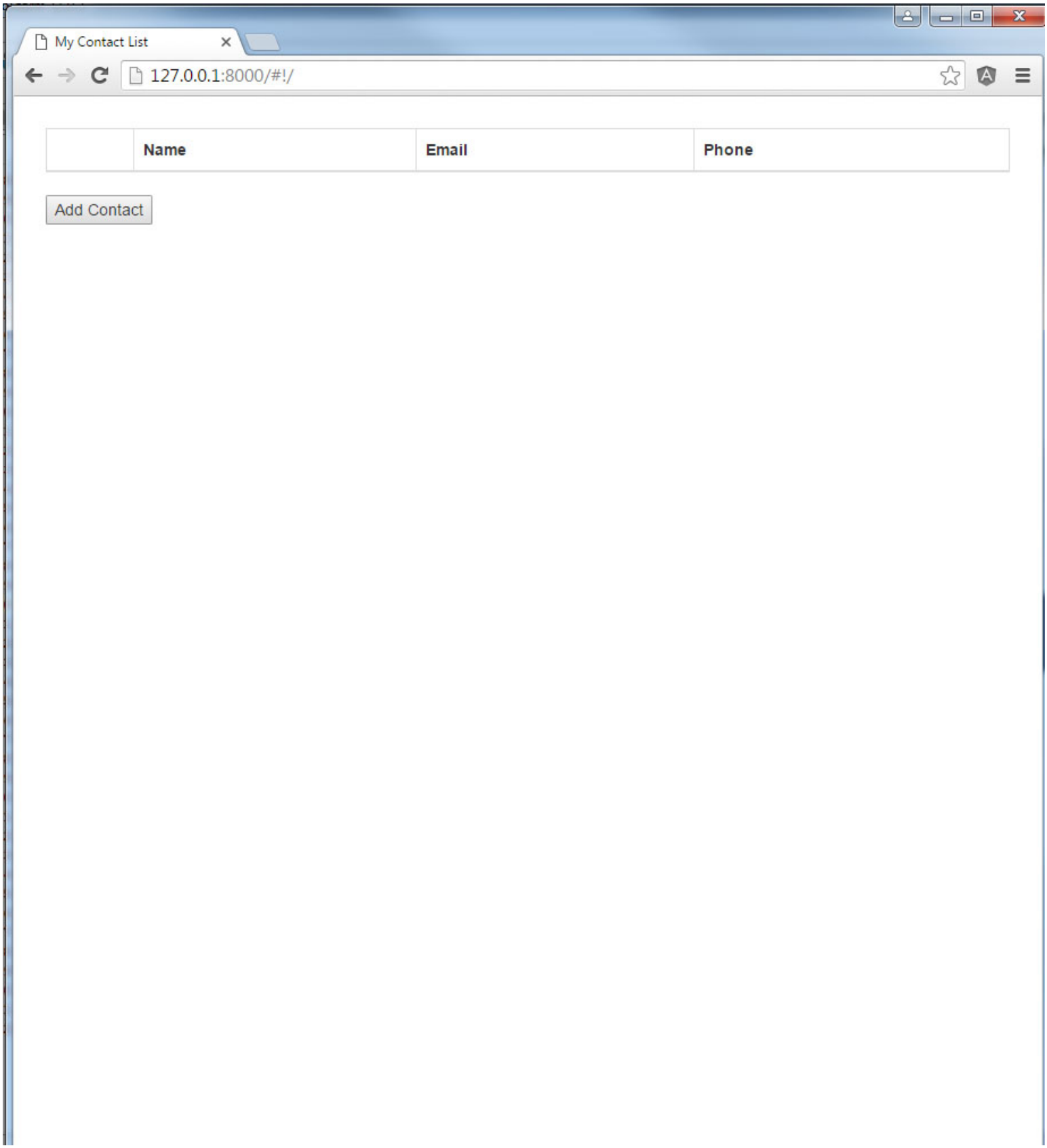
Resources

The following are resources that will be very useful:

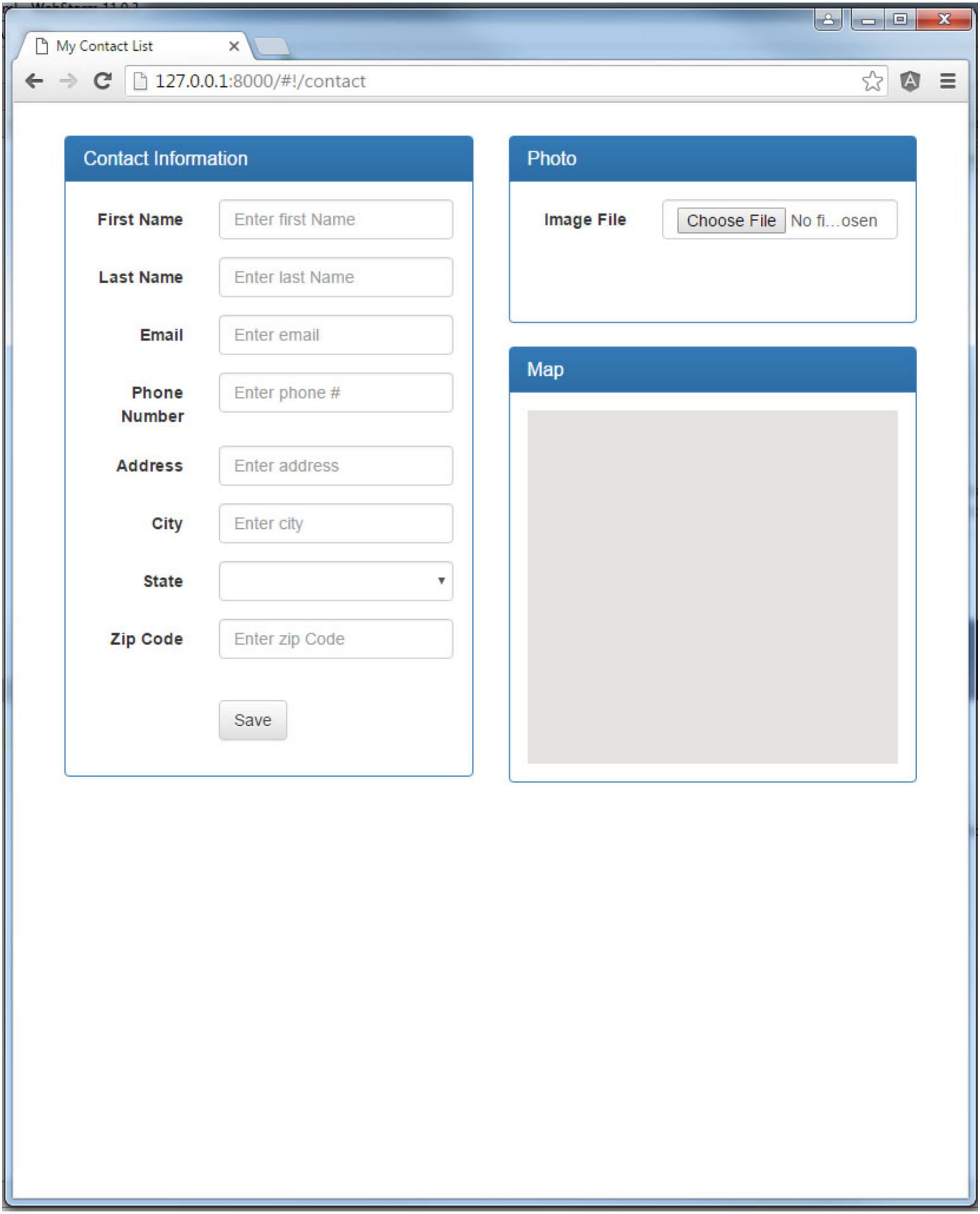
- http://www.w3schools.com/html/html5_webstorage.asp
- https://github.com/grevory/angular-local-storage
  - Specifically the set, get, keys, and remove methods. There are others that you may find useful.
- https://github.com/jsmodules/angular-file-data-url
  - This is similar to the custom file upload directive we went over in class but a little simpler to use
- https://github.com/allenhwkim/angularjs-google-maps
  - Specifically this: http://ngmap.github.io/#/!marker_with_dynamic_address.html

Requirements

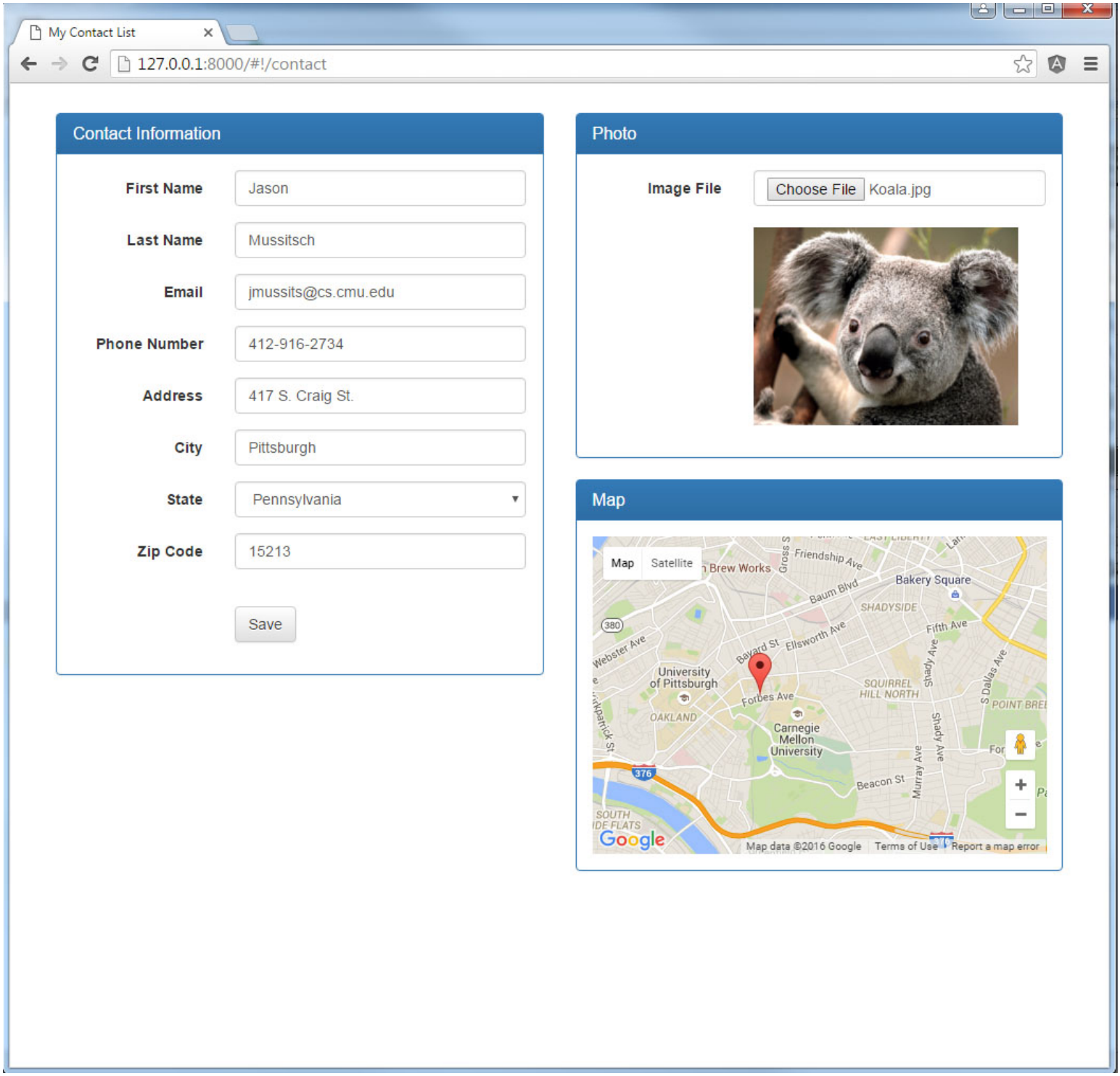
This will be a simple contact management application. When you first navigate to the application you will not have any contacts so there will just be an “Add Contact” button.



Once we have contacts they will be displayed in the table. You can show the table header as I have when there are no contacts or you can hide the table completely. It is up to you. When you click the “Add Contact” button you will be directed to the contact page.

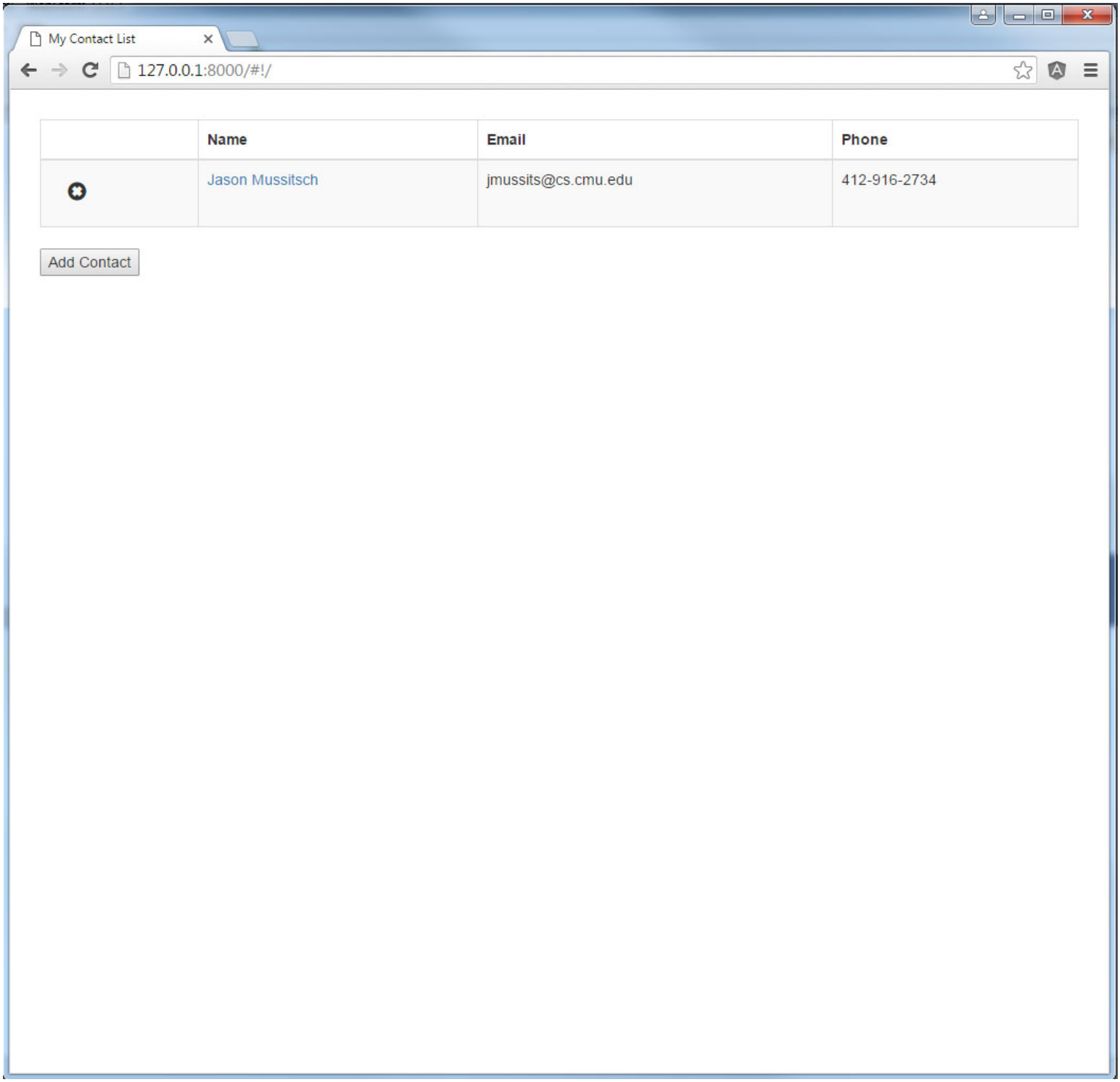

My Contact List
127.0.0.1:8000/#!/contact
Contact Information
First Name
Enter first Name
Last Name
Enter last Name
Email
Enter email
Phone Number
Enter phone #
Address
Enter address
City
Enter city
State
Zip Code
Enter zip Code
Save
Photo
Image File
Choose File
No fi...osen
Map

This will be a page that has the form inputs shown on the left as well as a “Save” button. On the right is a Photo section that uses the 3rd party file upload directive. When a photo is uploaded it should immediately be displayed below the input as shown in the screenshot below. Below the photo section is a map section that uses the 3rd party map directive. When an address is entered the map should zoom in and mark the address as shown in the screenshot below.



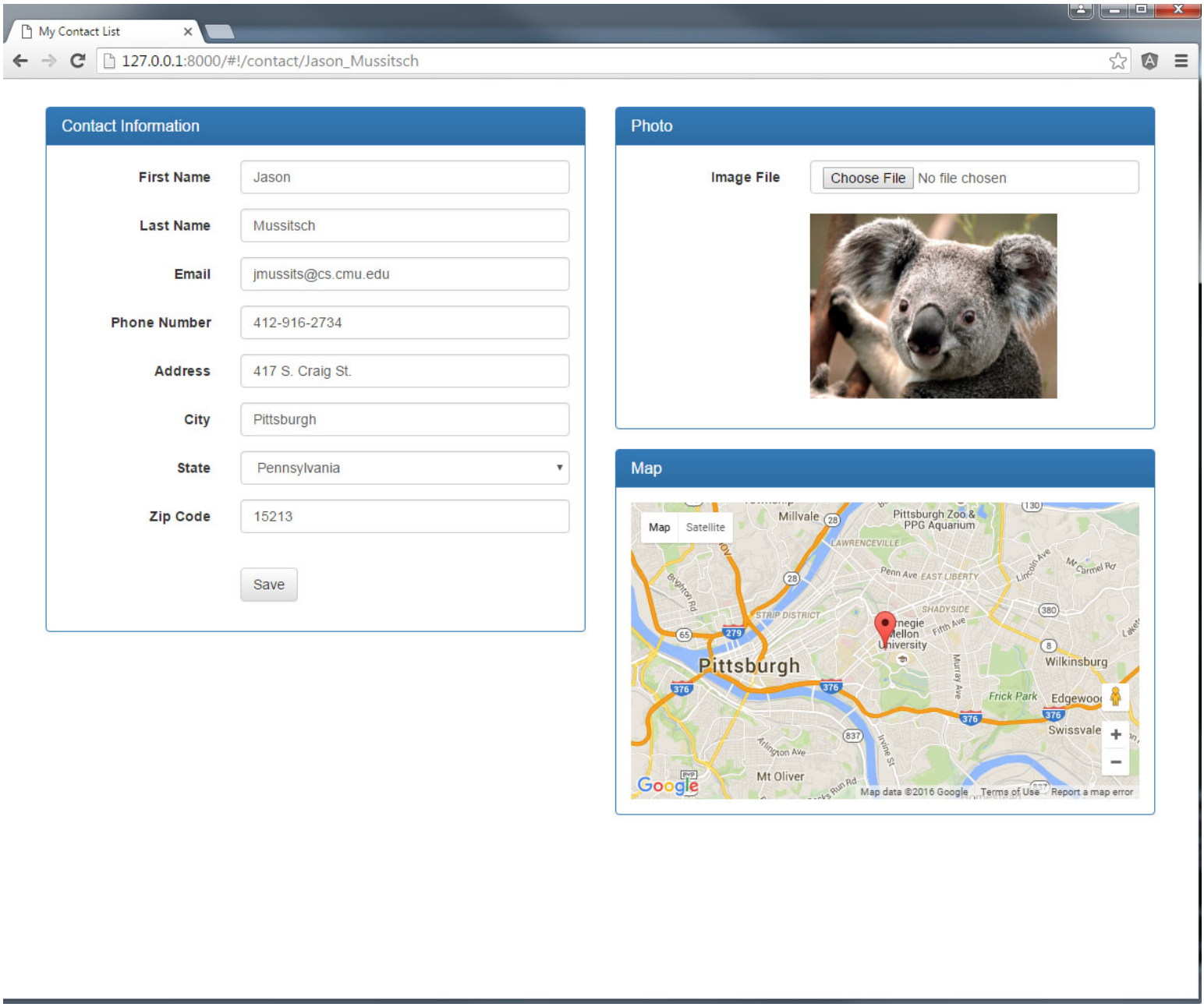
Your layout and styles do not need to look exactly like shown. I used Bootstrap but you are not required to. This project does not have any CSS requirements.

When the “Save” button is clicked you will save all the contact information including the photo image to local storage using the 3rd party local storage service. Local Storage works by storing data using key/value pairs so you will need a key for each contact. You can generate some id or you can use first name and last name as the key. Your application will allow for editing of contacts so if you use the name as a key you must handle this either by not allowing name to be edited or by removing the old key and adding the new key when names change. Once the data is stored your application must switch back to the initial page that lists contacts. See the screenshot below to see how it should look when there are contacts.



Name, Email, and Phone should be displayed as well as a clickable X icon on the left that when clicked removes the contact from the table and from local storage.

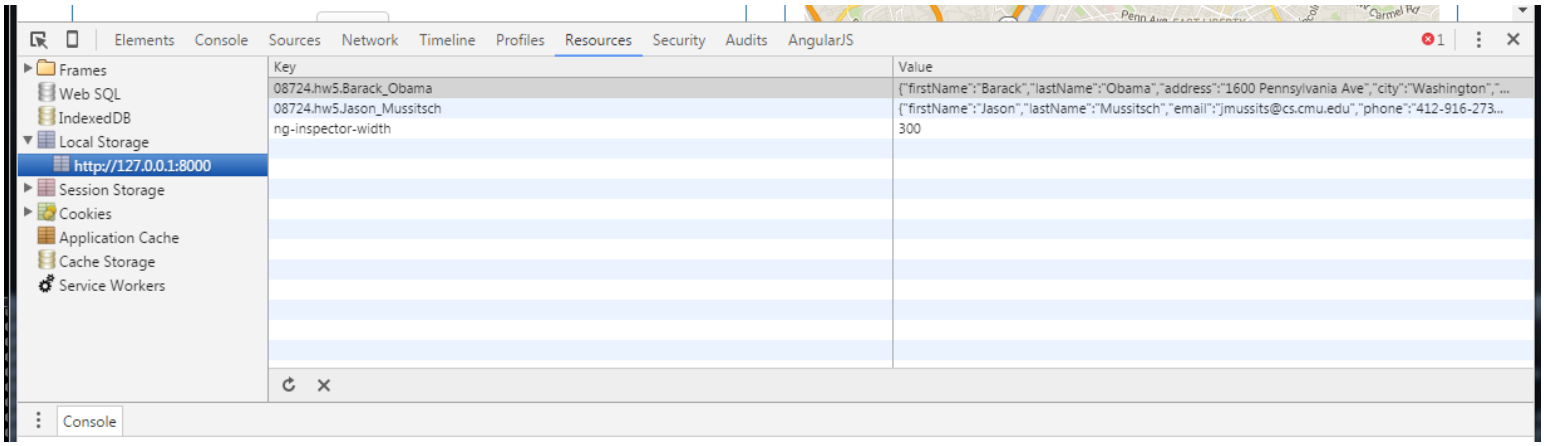
The name column should show first name then last name and be clickable. When clicked, the contact page should load and be populated with all the contact details for that contact as shown below.



Making changes and clicking “Save” should update the contact in local storage and direct back to the list page.

Viewing Local Storage

Using Chrome's dev tools we can look at what is in local storage. Go to the Resources tab and then expand the Local Storage node on the left. Click on the domain for your server and you will see all the key/values currently stored by the browser. Local Storage saves the data even after you completely close the browser. So you should be able to close your browser and reopen to your application and see all the saved contacts.



Submission

To submit your assignment, add your files to a ZIP archive, name the ZIP file ***Homework5_[andrewid].ZIP***, and upload to Blackboard under Assignment 5 by the due date/time above. For example, my ZIP file name would be ***Homework5_jmussits.ZIP***.

Please do NOT include your node_modules folder or your bower_components folder. I do not need these and they will make your submission too large.

To grade your application I should be able to run from your unzipped directory npm install and then npm start and then navigate to http://127.0.0.1:8000.

Grading Rubric

This assignment is worth 160 points (16% of your total grade). The following is how the assignment will likely be scored:

- Functionality works as described above **[100 points]**
- AngularJS is used correctly (e.g. no DOM access unless in directives, $scope used properly, injection patterns followed correctly, etc.) **[60 points]**