March 16th, 2017

CSE 135
HW#5 Write-Up

Shopping List App
MADE BY: Tianyi Li, Milad Imen, Brian Chao
Link to our app: https://cse134b-all.firebaseapp.com/

For CSE 134b, our team decided to make a tool that would allow a user to browse meals, choose their favorites, and easily create shopping lists. Following this will be a complete documentation of our app.

Technology Stack

Front End	Back End
HTML	Firebase Server
Bootstrap & CSS	
Vue.JS	
Firebase Client	

Our front end client consisted of the basics; HTML, CSS, and JS. We incorporated Bootstrap, a CSS framework, which provides many useful basic CSS classes for common use cases. In order to make our HTML templating easier and managing our front end data, we used Vue.JS. Vue has two way data binding which made it very easy to retrieve data from various sources and have it automatically render our view based on that data. Also, it made it easy to attach functions to different buttons on our site which was the core of our functionality.

For our backend, we used the professor's recommendation of Firebase, which provides an easy to implement authentication system, as well as a persistent NoSQL database which makes it very easy to retrieve and store JSON data. To access the backend, we used Firebase's client framework, which provides useful front end javascript functions that make it easy to connect and interact with the backend.

In addition, we used the edamam API as our source of data regarding food items, recipes, ingredient lists, calorie counts and other information.

Project Structure

This is in regards to the HW5 portion of our github repo.

/views/ - This folder contains all of our HTML templates for each page.
/img/ - This folder contains all static images.
/style/ - This folder contains all of our custom css stylesheets.
/js/ - This folder contains the script files containing Vue and Firebase code for each HTML page.

Functionality

1. Login Page
    a. The login page gives the user three options, all implemented using firebase authentication.
        i. register an account
        ii. login to an existing account
        iii. or use Google to login.
    b. Any error messages relayed back from Firebase are displayed on the page in a special “message” div, which gives the user an idea of why their login did not work.
    c. This page was implemented using vanilla javascript and the Firebase client.
    d. After logging in, you are redirected to the home page.
2. Once you are logged in
    a. You have access to the toolbar. The toolbar can:
        i. Let you logout (deauthenticates you, and sends you back to the login)
        ii. View the home page
        iii. View the shopping list
        iv. View the search page (find meals)
3. Home page
    a. The home page shows you a table containing all of your favorited meals that you added from the “Find meals” page.
    b. For each meal, you can:
        i. see a photo of the meal
        ii. View it’s name
        iii. View the calorie count
        iv. Click on a link to the recipe
        v. Take a variety of actions
            1. reorder the meal on your list
            2. add it to your shopping list
            3. remove it from your favorites.
4. Find meals page
    a. The find meals page allows you to search the Edamam database for recipes.

    b. After searching, a paginated table shows up which shows you 5 search results at a time.
    c. For each food, you can:
        i. View a photo
        ii. View it's name
        iii. View it's calorie count
        iv. Click on a link to the recipe
        v. Add it to your favorites
5. Shopping List page
    a. This page shows you a table consisting of all of the meals that you've added to the shopping list.
    b. For each meal, you can:
        i. View it's name
        ii. View an itemized list of it's ingredients
        iii. Remove it from your shopping list

Limitations

While our app has come a long way, there are definitely places where it could be improved.

1. Styling - The layout and organization leaves something to be desired. Some custom CSS would definitely make things look less bootstrap-py and stale.
2. Form submission - When trying to use forms with Vue functions, we found that form submissions would cause a form submit and cause the page to refresh. This was undesirable, as we wanted it to just call the v-on:submit function which executes asynchronous javascript. Our way around this was taking our form inputs and buttons out of the form tags. The only problem is that you must use mouse clicks to submit any forms; the enter key does not work.
3. Improving asynchrony - A loading indicator while we are loading information from the API or from Firebase would make the page loads seem more seamless. As it is, there is a slight lag from page-load to the tables rendering.
4. Additional Features
    a. Calorie count totals
    b. Multiple shopping lists
    c. Ingredient by ingredient shopping list with a checklist
    d. Custom meals