FoodDex Documentation

Developer: Jason Pon Contact: japon@ucsd.edu

Demo Link: https://fooddex-3afc7.firebaseapp.com/hw5/wireframes/login.html

Pepperoni Pizza		
Carrot	Home	
Fish Taco		
Medium Rare Steak		
Hash Brown		TRIED
Green Beans		
Wild Salmon		
		Entree The classic Pepperoni Pizza, easy to eat, quick for parties. Ate Today Delete Item
Search Submit Query		



Code Architecture

FoodDex was created utilizing HTML, vanilla Cascading Style Sheets (CSS) and vanilla Javascript. The decisions to commit to these tools came during the development process. In build #3 of the web application (before the current "final" build build #5), we experimented with Framework and Vanilla CSS, ultimately committing 4 hours and 14 hours respectively to build a workable styled application. The Framework-styled (BootStrap) website was ultimately bloated with unnecessary libraries that were unused, the structure of page elements were unorganized, and there were too many number of things that needed to be fixed. Ultimately, the greater amount of polish for the vanilla version and the amount of learning needed to correct and make the BootStrap version efficient led us to proceed with vanilla CSS.

In terms of Javascript choice (VueJS, jQuery, VanillaJS), our team decided on VanillaJS. In build #4 of FoodDex, a prototyping phase, we ultimately did not have the chance to prototype different Javascript frameworks. During that phase we experimented with VanillaJS and partially created working implementations of CRUD (Create, Read, Update, Delete) operations. The different ways other javascript libraries worked (e.g. \$ selectors for jQuery) required additional learning that was beyond the scope of the team's capabilities at the time. For the current build (#5), we proceeded

implementation with vanilla Javascript. Of particular note during development is that some operations might have been easier to implement with the use of other Javascript frameworks, but so far, there haven't been any hurdles in not being able to implement a specific feature with VanillaJS.

In regards to javascript organization, all scripts are currently inlined within the HTML page the script is used on. Our team experimented with referencing scripts as a separate javascript (.js) file, but we've had no success in making this implementation work, as moving the scripts to an external file broke the implementation of working functions and we have no idea how to make it workable. There were other factors to consider including the added time/processing it would take for the network/system to reference the external file that would make an external reference file undesirable. Also, most scripts had function implementations that were specific to the page they are located in, so there are few cases where code are duplicated and the development time/effort to make functions modular was not worth investing in.

For CSS organization, most elements are styled uniquely and so are done inline. We have an external stylesheet for repeating styles for similar elements, and in some cases, override/overload styles referenced in the stylesheet.

2				
3	<pre>chtml lang="en" xmlns="http://www.w3.org/1999/xhtml"></pre>			
4				
5	(smeta charset="utf-8" />			
6	<pre><title>FoodDex - Login</title></pre>			
7	<pre>ink rel="stylesheet" href="/stylesheet.css" type="text/css" /></pre>			
8				
9	<script src="https://www.gstatic.com/firebasejs/3.7.0/firebase.js"></script>			
10	<script src="https://www.gstatic.com/firebasejs/3.6.10/firebase-app.js"></script>			
11	<script src="https://www.gstatic.com/firebasejs/3.6.10/firebase-auth.js"></script>			
12				
13	□ <body></body>			
14	<hl><hl>class="siteName">FoodDex</hl></hl>			
15	<pre>form action="#" onsubmit="loginAccount()" style="text-align:center; border:solid 2px; width: 20%; position:relative; left</pre>			
16	Welcome to FoodDex! > Please login or signup to start enjoying all the features we offer!			
17				
18				
19	email:			
20	<input autofocus="" id="email" required="" type="text"/>			
21	<pre> </pre>			
22	password:			
23	<input id="password" required="" type="text"/>			
24	 			
25	torgot password?			
26				
2/	<pre>cinput type= submit value= login ></pre>			
20				
30	<pre><ur><pre>cur></pre></ur></pre>			
31	<pre></pre>			
32	<pre></pre> <pre> </pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre>			
33	<pre></pre>			
34				
35	ESC: Return to Index			
36				
37	<script></script>			

General organization of code

<head> Reference stylesheet, libraries </head> <body> HTML elements, and styling Javascript scripts </body>

Organization of Files



Current File Structure

Files are current organized by content. HTML pages have their own directory, image content have their own directory and commonly referenced files like stylesheets are located at the root directory.

Concerns, Limitations, Caveats, TODOs

Every effort has been made to prioritize as many working features, fixes, and improvements to FoodDex in the limited amount of time remaining. Unfortunately, there is still a list of tons of improvements and implementations to work on.

Among the chief concerns right now is security and speed. Perhaps the web application is not secure enough if text fields accept malicious input, if implementation details are currently exposed, etc. Security hasn't been a big focus thus far so it may be important to consider in the future. In terms of speed, in the implementation of certain features, our programmers implemented features with algorithms they came up with at the time, involving loops and other resource intensive structures. In the future, it may be better to research if better algorithms could be used to provide the same implementation more efficiently.

For the limitations of FoodDex, it is currently not built to scale. It's database and implementation currently isn't setup to support multiple users. We focused on making the web application work for a single user before expanding our scope of users.

TODOs

- Improve the visual design of the website and make it responsive for different devices
- Create more image assets for different foods
- More checks on user input
- Improve security of website: hide implementation details
- Expand scale of web application to support multi-users
- Finish initially planned & in-progress features.