Food Me Once (https://foodmeonce.me)
Phase One Technical Report

Useful Links for Project

FoodMeOnce GitLab Repository -
https://gitlab.com/shub95/foodmeonce/
FoodMeOnce Postman documentation -
https://documenter.getpostman.com/view/7777503/SVtPXWHE?version=latest
Food Access Atlas API
https://www.ers.usda.gov/data-products/food-access-research-atlas/download-the-data/
Socioeconomic Estimate API
https://hudgis-hud.opendata.arcgis.com/datasets/acs-5yr-socioeconomic-estimate-data-by-tract
Congressional Data API
https://projects.propublica.org/api-docs/congress-api/

What is the purpose of the website?

Food Me Once is a website geared toward helping users understand the food security and health of their communities. By providing information about food security to users by pulling from health databases, and then letting the users explore the actions of their political representatives and federal legislation from government databases, we aim to create an easy way to understand the causes for levels of food security in various federal districts. Food Me Once will provide a platform to best understand what efforts are being promoted politically to promote healthy food access to people of all demographics.

Motivations

Going to college in Texas, it can be pretty easily seen the disparity in wealth between various demographics, and even entire regions of the state. Within this state alone you can find residents and people with enough distance from a healthy supplier of food to be federally recognized as living in a food desert, as well as those who simply cannot afford to buy healthier food options due to low economic conditions. Our group was motivated to use our unique skills in software engineering to help others make connections between political representation, legislation, and

food security. We hope that with easier access to knowledge surrounding the causes of food security and health disparity, that actions can be taken to improve the situation in not only our community but many others.

User Stories

Customer

We provided user stories for Put it in Park. First two of which have already been closed.

1. It would be useful to see the description of the project along with the aim. Potentially, the details of each model and how they correlate. Additionally, a cool background (not necessary) might be a nice-to-have. Overall, the website can be static for now - it will need to be dynamic in the future phases.
2. A profile picture for each group member. Along with that, a total summary of the entire repository commits + issues would be cool to see. It would not be necessary to make the page look "pretty" for now - just a simple picture + commits & issues.
3. It would be preferred if the model pages had a grid or table with links to the instance pages. Each instance listed on the model page should have 5 attributes. We must be able to click on a link/somewhere to navigate to the instance page. To be clear, if the model does not lend itself to a variety of media (for eg: outdoor activities) - then a table is probably more idea. However, a model like National Parks might have lots of pictures/videos specific to each park, and hence a good candidate for a grid. Regardless, a table is fine for now too.
4. There should be a link to the API documentation - designed by Postman. It would be ideal to have 'GETs' for models + instances.
    1. return a list of models
    2. return the details of the instances (attributes)
    3. return a detailed list of models with their 5 basic attributes. No PUTs/POSTs. There should be the https://api.website.me to communicate with the API.
5. Provide 3 instance pages for each of the 3 models. Each instance page must display the original 5 attributes from the model page + additional data points. Along with that, there must be some sort of media (more than one preferred) on each instance page, that is specific to the instance: links to videos, images, etc. It does not need to be "pretty" or formatted super well.

Developer

After Work was our customers. They put 5 user stories for our group. First two of which have been implemented and closed out of issues.

1. Have the home page explain the name of the website a bit more, since right now it's a bit confusing and it sounds like its a mismatch between the name and content.

Estimated Weight: 1. Implemented such that the splash page explains the name of the website.

2. Switch the layout of legislation to a table view since it doesn't really make sense to have a picture with every new legislation.
   Estimated Weight: 2. Changed to lists.
3. Add a petitions page where we can see what current petitions are going on about certain legislation, maybe requires another API.
   This was out of the scope of this project.
4. I would like to see information about obesity in these districts.
   Estimated Weight: 2. Not implemented yet
5. Can I see the cost of living, income inequality, and homelessness stats on the site?
   Estimated Weight: 2. Not implemented yet.

RESTful API

For phase 1, we have simply designed what calls to our API might look like in the future, although the website currently operates without a RESTful API. In future phases, we will scrape data from at least three distinct databases to populate a RESTful API of our own. Our first major data source will be a health security map of the united states, allowing us to obtain information about the demographics of certain locations, as well as the health and food security of those regions. Secondly, we will scrape data to obtain information about all the political leaders in both the House of Representatives and Congress. Lastly, we will scrape data from Congressional databases revolving Food health and security acts in the United States. We believe by combining these three distinct sources and populating a database of our own, that we can use this information to create a website that allows users to truly make connections between food security and political actions, as well as see what efforts have been made by various leaders to promote higher food access. The API's we currently plan to scrape from can be viewed from the links in the corresponding useful link section of this report.

Models

For this phase of the project, we have designed three model pages. Each model page has a table with three instance data items with five attributes. Above the table, there is a sort by and filter by menu. For now, the menu is just for looks and we will implement it in the later phases. In the district model, we have three congressional districts with attributes relevant to the congressional district. Each district instance page has additional attributes that determine access to healthy food. Also, we have a map of the district and a photo of the representative. So, you

can access the representative instance page from the district instance page. In our representative model, we have three House of Representatives and attributes about their experience and district they serve. In the representative instance, there are additional attributes such as contact information, social media handles and their website. We also have a picture of each representative in the instance page. We have included legislation that they sponsored and a map of the district they serve. Users can access the legislation instance and district through the representative instance.
In the legislation model page, we have three legislations and each legislation is sponsored by one of the three representatives from the representative model. For the attributes, we have included when the legislation was introduced, who introduced it, which party introduced it and its status. We also included the type of bill it is such as Senate bill or House of Representatives bill. In the bill instance, we have an image of the type of bill. We have additional attributes such as who is sponsor, number of cosponsors, the committee it is assigned to, the latest action and summary of the bill. There is a photo of the representative who sponsored the bill and the representative and district instance can be accessed through the legislation instance.

Tools

We used the React JavaScript library to build the front-end of the website. We used Bootstrap as our CSS framework. We used Amazon S3 to host our static website and deploy it. We configured Cloudfront so that the website would deliver efficiently. We configured Route 53 for our domain name which we got using Namecheap. This allowed for our website to be accessible at foodmeonce.me

Hosting

Food Me Once(https://foodmeonce.me) is, at the moment, a static website hosted in AWS S3 bucket. All of the files in S3 bucket are the files generated in the build folder by invoking “npm run build” command. AWS S3 is then connected to Cloudfront to provide the https access. Lastly, we got a pretty domain from Namecheap, then connected the domain to URL generated by Cloudfront, so that the website can be visited using the pretty domain.