

 README.md

Nutrition Value

Milestone project from Data Centric Development

Data

The data is the nutritional value per 100g (or 100ml for liquids).

- Energy (kJ)
- Energy (kcal)
- Fat
- Saturated Fat
- Carbohydrates
- Sugars
- Fibre
- Protein
- Salt
- Classification
- Shop
- Notes

Others might include: B12 and Calcium.

I want to allow for data which is Not Available (NA) and values which are marked as less than (<).

Pages

1. I will have a home page with some description of the project and instructions for using the application.
2. I will have a page to add data [C]. This will be reached from a menu item.
3. I will have a page to display [R] the items. I will use an accordion structure. I can do one of two things:
 - i. There will be a button to lead to a more details. Here there will be a button to delete the item.
 - ii. All the details will show once an accordion fold is open. There will be a button to delete the item on the accordion item header.
4. In any case the delete [D] button will either open a modal or take the user to another page for confirmation.
5. I will have an edit page [U]. It will be structured like the add page but will be prepopulated with the data for the particular item to be edited. There will be a button to save the changes or leave without saving changes.
6. There will be a Dashboard with DC.js graphs with filters and the possibility to select an item and go to the item's detail page. (Or the accordion fold of that particular item.)

Deploy to Heroku

Name of app: food-nutrition

URI: <https://food-nutrition.herokuapp.com/>

I am using gunicorn server. I found that I will have to set less settings myself.

The differences include:

1. I do NOT need to run `heroku ps:scale web=1`
2. I do NOT need to manually set the IP and PORT variables in settings > Reveal Config Vars.
3. I do NOT need to Restart all Dynos from More.

In all, using gunicorn makes deploying to Heroku much easier.

The commands needed are: (using this app as example)

1. Install gunicorn

```
pip3 install gunicorn
```

2. Create Procfile

```
echo web: gunicorn run:app > Procfile
```

3. Create/Update requirements.txt

```
sudo pip3 freeze --local > requirements.txt
```

4. Create the Heroku on the website (allows me to set Region to EU)

5. Log in to heroku (locally, need email and password) and check the apps

```
heroku login  
heroku apps
```

6. Initialise git, if not already done and set a remote for heroku

```
git init  
heroku git:remote -a food-nutrition
```

7. Push to Heroku

```
git push heroku master
```

DONE. -- Open the app at <https://food-nutrition.herokuapp.com/>

Addition

To access the files on Heroku,

1. go to More > Run console.
2. run the command

```
heroku run bash
```

3. Use normal bash commands.

.

Database

Create a MongoDB database on mLab.

Database name: food

MONGODB VERSION: 3.6.6 (MMAPv1)

- add user
- add collection: nutrition100
- created 3 documents with some data.

I had to do some changes in the code I used in previous projects.

MONGO_DBNAME and MONGO_URI had to be used as app.config:

```
app.config["MONGO_DBNAME"] = getDbName()  
app.config["MONGO_URI"] = getURI()
```

The port argument complained about the int() method, so now it is:

```
app.run(host=os.getenv('IP'), port=os.getenv('PORT'), debug=True)
```

Add Food Item

Implemented add food item.

Now updating Heroku.

The following code:

```
from connection import getDbName, getURI  
.  
.  
.  
.  
app.config["MONGO_DBNAME"] = getDbName()  
app.config["MONGO_URI"] = getURI()
```

does not work in Heroku.

Instead, in Heroku set environment variables for MONGO_DBNAME and MONGO_URI with the correct information.

Classification

I created another collection in the database that will hold the different classifications of food. This will help to give unity to the data.

I implemented the code to populate the classification input field with classes to select from in the add food item page.

I implemented adding new classes. I also added the functionality of returning to the calling page if the process is cancelled.

Implemented editing a class.

Backups

I implemented two routes that are used either to get a backup of the database, or replace the database from a backup file. This will help me in two cases.

1. I mess up the data during testing where it is quicker to replace the lot rather than edit individual entries
2. The data has been messed up by someone (possibly intentionally) - I am not currently implementing a log in system.

Dashboard

Drafted a dashboard with 2 scatter plots - fats vs sat, carbs vs sugar.

I plan to have 4 scatter plots and a bubble chart:

1. All Fat vs Saturated
2. All Carbohydrates vs Sugar
3. Energy vs Protein (have not decided which Energy value I will use)
4. Sugar vs Salt
5. All Fat vs Carbohydrates with the following additions:
 - size of bubble - Protein
 - color - changes with quantity of Sat fat
 - border color - changes with sugar content.

I want to be able to cancel each individual filter and all filters. I would like to be able to transfer the filters to the `get_food_items` view, if it is possible, to display only the selected foods in the accordion.

There is something that need to be changed in the database. Currently, the values are stored in the database with the units attached. This is not what crossfilter etc wants. Numbers should be numbers. I can do the processing in javascript(jQuery) to remove the units before passing the data to crossfilter or I can store the numbers without the units. This is the way I am choosing to do. As a result, the display of data in the accordion view will show without units unless I change the code to add them for display. This is the next step. I also have to remove the code which was adding the units when adding a new food item.