

🔗 harishparuchuri / lab-angular-calory-tracker

forked from FACEPrep-ProGrad/lab-angular-calory-tracker

No description, website, or topics provided.

Edit

[Manage topics](#)

7 commits

1 branch

0 packages

0 releases

1 contributor

Branch: master ▾

New pull request

Create new file

Upload files

Find file

Clone or download ▾

This branch is even with FACEPrep-ProGrad:master.

Pull request

Compare

FACEPrep-ProGrad Update README.md

Latest commit 876996c 21 days ago

README.md

Update README.md

21 days ago

README.md



ProGrad Lab | Calory Tracker

Introduction

At this point in the course, we know you have a tight schedule and your day is all packed up. Some of you might infact be reminiscing about times before the course when you use to work out everyday and were in shape.

Well, let's not regret about it anymore. Think of what could you do to stay fit at present.

Our solution: Eat Healthy 🍌🍇🍓🍌🍇🍓

Now, how would you know what you consume is healthy or not or how many calories does it contain?

C'mon ProGrad, you no longer are suppose to have such questions in mind. Let's quickly build a Nutrition application that helps you get all your answers.

In the starter code, we've provided a list of foods in the form of a TypeScript array of objects. This has already been imported into the foods-list component provided to you.

Learning Goals

After this lab, you will be able to:

- Create a static Angular application with Angular CLI.
- Build an Angular application with one component.
- You must use at least 1 of each of the following:
 - component
 - pipe
 - ngFor
 - ngIf
 - ngModel

Requirements

- Fork this repo.
- Clone this repo.

Submission

Upon completion, run the following commands:

```
$ git add .  
$ git commit -m "done"  
$ git push origin master
```

Navigate to your repo and create a pull request from your master branch to the original repository's master branch. In the pull request name, add your ProGrad id, name, and last name separated by a dash "-".

Deliverables

You need to generate the starter code and fill it with the necessary code to satisfy the requirements described below.

Starter Code

To generate the starter code, follow the steps given below

- To create a new application,
 - Open your ubuntu or cmd terminal and execute the following command
 - `ng new app-name`
 - for example, `ng new super-wars`
 - To create a new component, execute the command
 - `ng generate component component-name`
 - example, `ng generate component contacts`

How to run

- To run the project go to your ubuntu terminal or VScode editor
 - open the ubuntu or cmd terminal or inside the vscode editor
 - run the command following command
 - `ng serve --open` or `ng serve -o`

Progression 1: Display Foods

In the `app-food-list` component, display a list of the foods. This should include the food's image, its name, and its calorie count.

Notice, we've already imported an array of objects containing food into the component.

Make sure the image's `src` property is properly bound so that any time the value is changed in the Typescript class, the change is reflected in the DOM.

Progression 2: Search Foods

Create an input box to search through the foods by name. You're going to need to generate a custom *pipe* to do this. Refer the example from the lesson on pipes, as it will be very helpful or you can also refer this link: <https://angular.io/guide/pipes>

Progression 3: Add New Foods

Create a button to add new foods.

When a user clicks the button, a form will appear with fields for a name, number of calories, and an image.

When the user clicks submit, the food will be added to the list. The form should disappear when the user clicks the submit button.

Progression 4: Today's Foods List

Create a button next to each item. When a user clicks the button, the corresponding food should be added to a special list of foods which are "today's foods".

Somewhere on the page, display a list of today's foods, with a total calorie count.

Progression 5 (Bonus): Quantities

In addition to the "Add to today's list" button, create an input for quantity. Whenever a user enters a number into the input, you should add those many items to their list.

The default quantity should be 1.

Progression 6 (Super Bonus):

If the user has added more than 1 item of the same type to their list, don't display it twice. Instead, display the item's name, and the number of times it has been added.

For instance:

- Banana x2
- Salmon x5
- Cake

Expected Output:

Calory Tracker

Search

Add New Food

Name

Calories

Image

Add Food



Example Name 250



Apple

Calory - 81


Add



Banana

Calory - 105

Add



Dates

Calory - 228

Add



Grapes (1 cup)

Calory - 114

Add




Biscuit

Calory - 103

Add

Today's Foods 2102 cal


- Apple 81
- Banana 105 x2
- Dates 228 x3
- Grapes (1 cup) 114 x3
- Biscuit 103 x5
- Strawberries 45 x6



Chocolate

Calory - 208


Add



Strawberries

Calory - 45


Add



Potato

Calory - 220

Add



Noodles

Calory - 159

Add

Happy Coding ProGrad ♡!