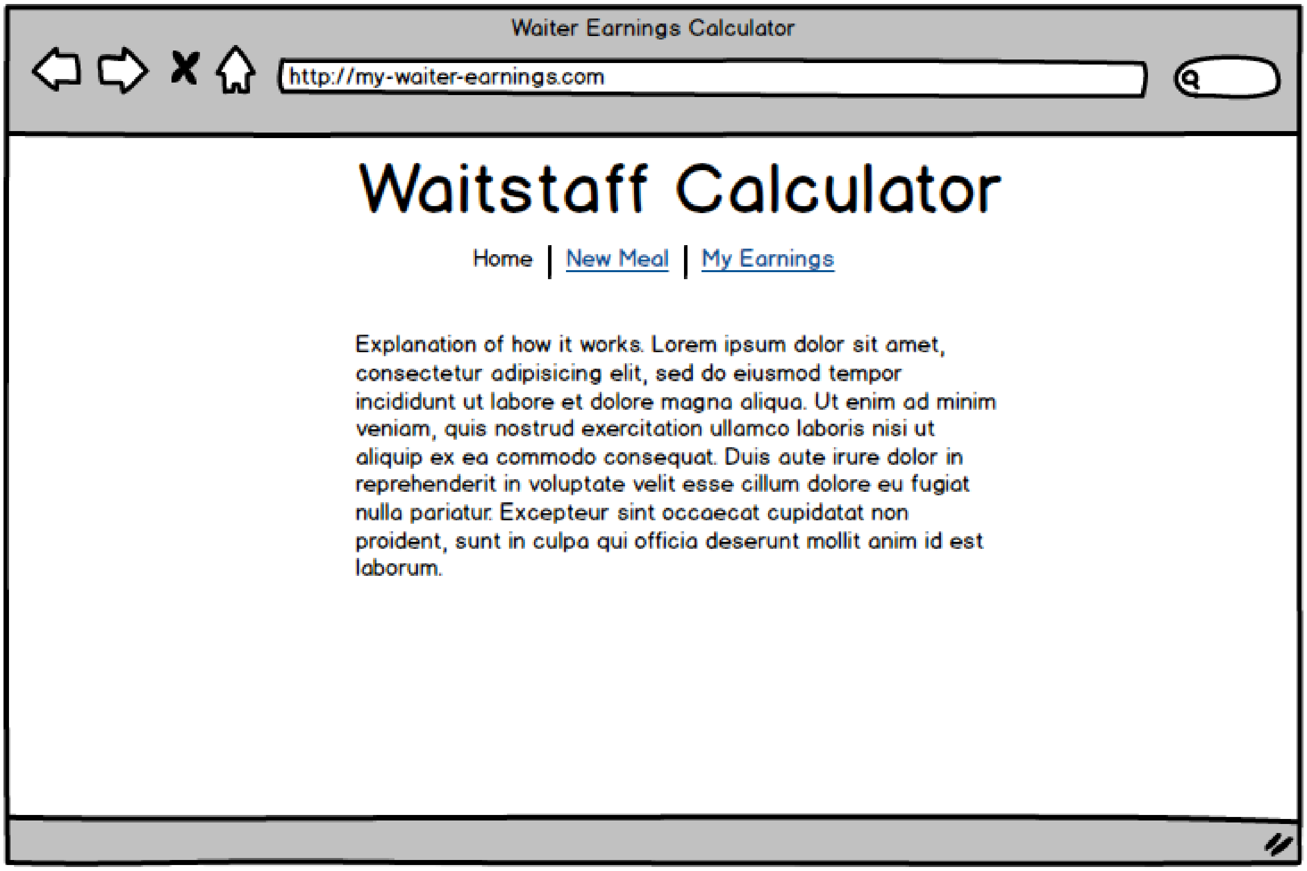
Add Routes to Waitstaff Tool

Estimated Time: 90 minutes

It's time to practice routing and views in Angular. For this project, we'd like you to revisit your the Waitstaff tool from Unit 1. When we last left off, this app had a single route and view. For this assignment, we'd like you to use ngRoute to generate three routes and views for this app. You'll have one view for the index page, one for entering the details of an individual meal, and one for seeing your earnings for a shift.

Index View

This view should load when users visit the root URL for the site ('/'). Here's a sketch:


Waiter Earnings Calculator
http://my-waiter-earnings.com
Home | New Meal | My Earnings

Waitstaff Calculator

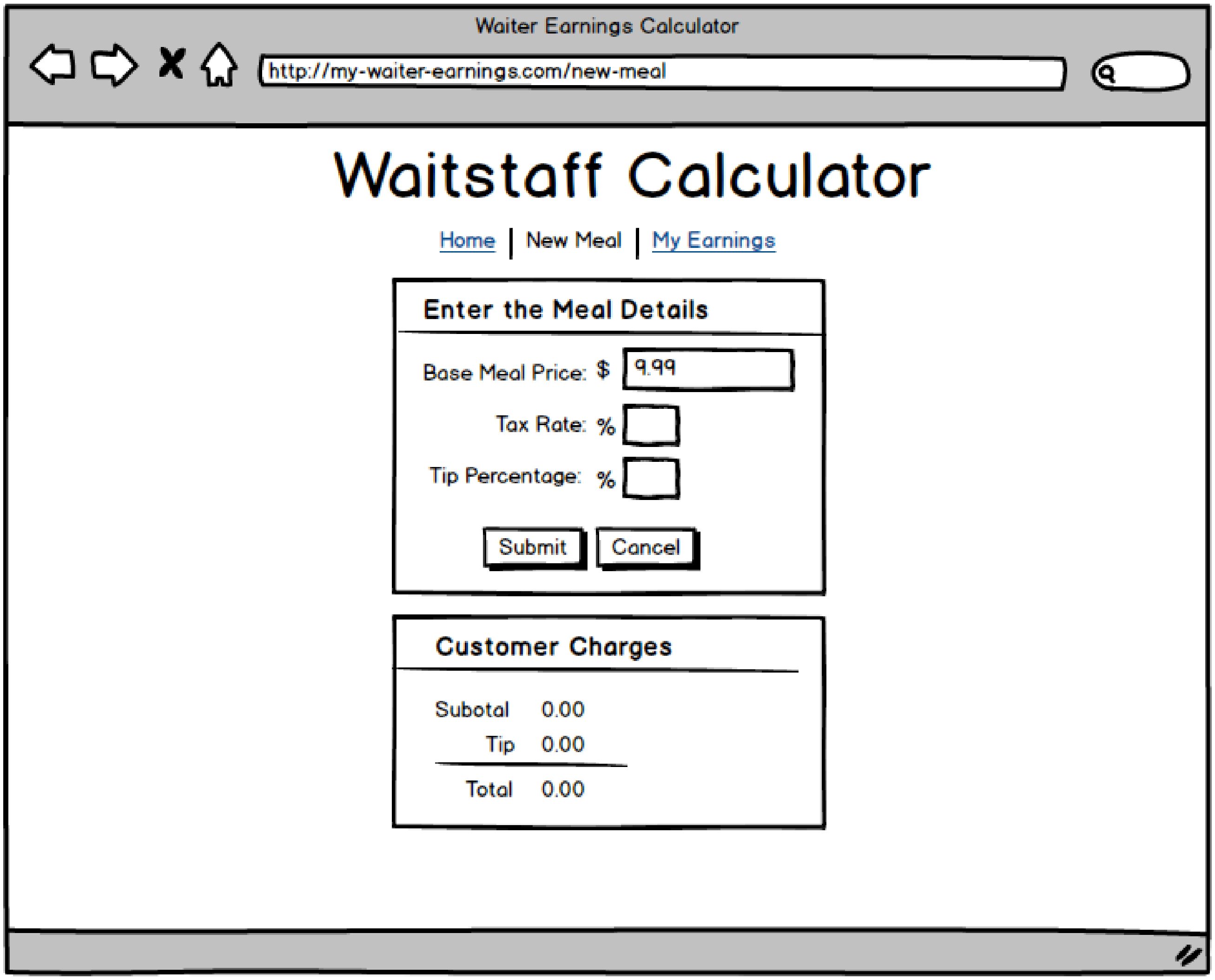
Explanation of how it works. Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

As you can see, this page doesn't do too much. We have some placeholder text for describing the app, and we have a link bar just under the title. From the link bar, you can access each of our three views.

Note that the title and navbar appear in each of the three views, so you'll want to write that code in index.html.

New Meal View

This view should load when users visit '/new-meal'. Here's a sketch:



Waiter Earnings Calculator
http://my-waiter-earnings.com/new-meal
Home | New Meal | My Earnings
Submit
Cancel

Waitstaff Calculator

Enter the Meal Details

Base Meal Price: $ 9.99

Tax Rate: %

Tip Percentage: %

Customer Charges

Subotal	0.00
Tip	0.00
Total	0.00

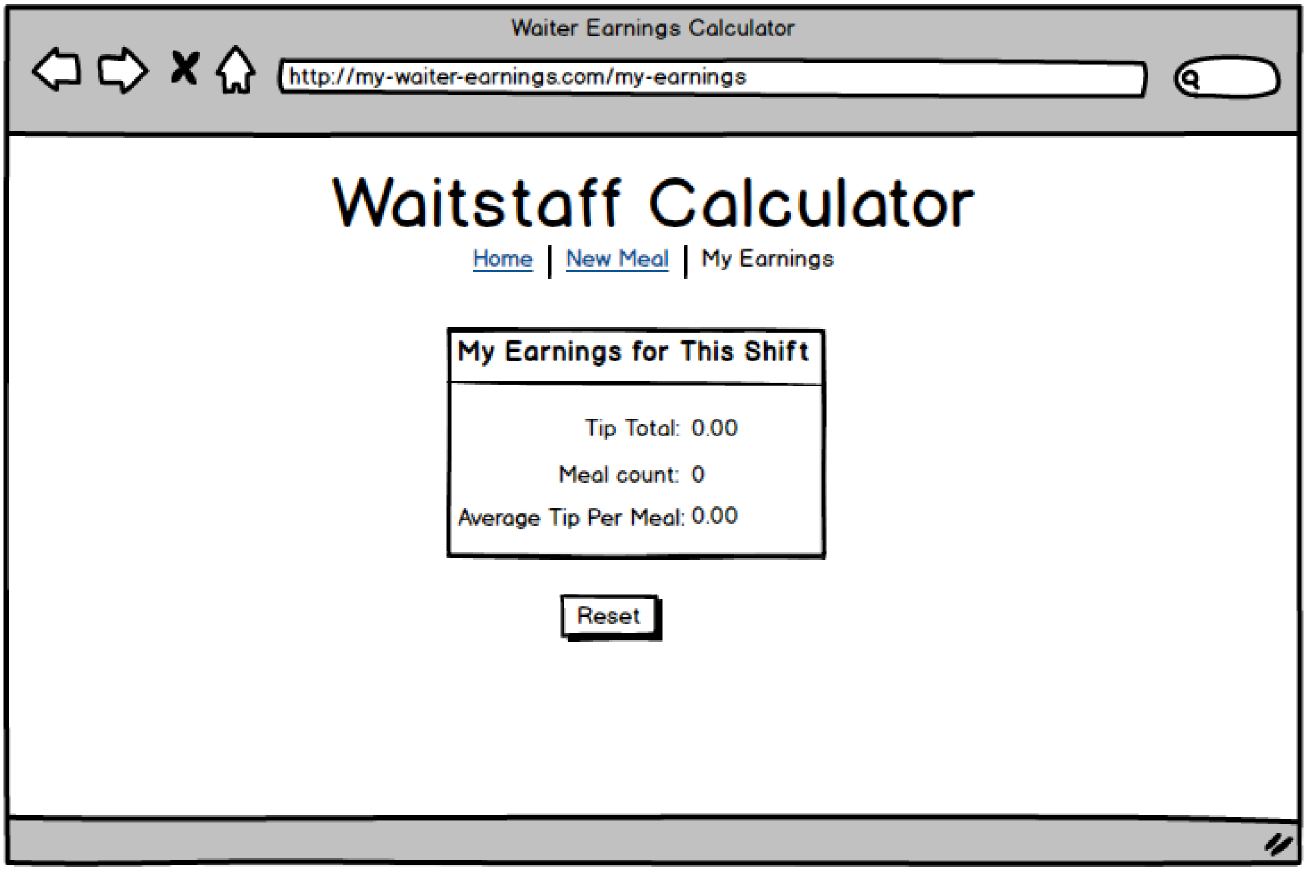
The behavior of this part of the app should be the same as before, with a few additions.

> The user enters details for a new meal in the "Enter the Meal Details" panel. When they click submit, the app validates that the value in each field is numeric. If the form is valid, the "Customer Charges" panel and data model for the My Earnings view should update accordingly.
>
> In the "Customer Charges" panel, *Subtotal* is the value of the base meal price plus tax. *Tip* is dollar value of the tip, given the subtotal and tip percentage. *Total* is equal to subtotal plus tip.

If the user clicks cancel, the form should be reset to its initial state.

My Earnings View

This view should load when users visit '/my-earnings'. Here's a sketch:


Waiter Earnings Calculator
http://my-waiter-earnings.com/my-earnings
Home | New Meal | My Earnings
Reset

Waitstaff Calculator

My Earnings for This Shift

Tip Total: 0.00

Meal count: 0

Average Tip Per Meal: 0.00

This view tracks the tip and meal totals for the waiter on a given shift. The data presented in this view should change as users add new meals. If the user clicks the "Reset" button, the cumulative earnings data should e zeroed out.

Additional Requirements

If users try to visit any routes other than the three we have explicitly defined, they should be redirected to the index of the site.

Be sure to use Git and Github for this assignment. We'll be adding another feature to this app in the final assignment for this lesson, so it's important you save your work as you go.

Your app need not be a pixel perfect rendition of these sketches. As long as your app provides the functionality indicated above, you're free to style the app as you please.

When you've completed this assignment, be sure to share a link to your work with your mentor. You should provide a link to the source code and a gh-pages web page.