Front-End Web 200 With Angular

Lab Manual

Revised 9/1/2020

# Overview

This lab is designed to help you deepen and extend what you’ve learned in the Front-End Web 200 instructor-lead course.

This lab is designed to take between 4-7 hours to complete. You should work with your manager to block off some time in which to focus on this, or sign up for a Workshop class. These one-day classes, sprinkled throughout the ITU calendar, allow you to have access to the instructor to get questions answered and guidance as you need it.

You will accomplish the lab using your assigned Azure Labs Virtual Machine – the same one you used in the classroom. **Please remember to shut down your virtual machine when you are not using it**.

# Log into your Azure Labs Virtual Machine

Using Google Chrome or Edge (not Internet Explorer), go the <https://r.labs.azure.com/virtualmachines>. Sign in with your Progressive email address if asked.

Start the FrontEndWeb virtual machine. When it is running, hit the connect button. The password is: ProgressiveITURocks!

# Beginning the Lab

You may continue with your own code or start with the instructor’s version. The instructor will provide a link to his completed code at the end of your class.

## If you are using the Instructor’s version of the Code

Here is the link to the instructor’s Github repo: <https://github.com/FrontEndWeb200/few200augsep2020>

While logged into Github, go to the above URL and hit “Use this Template”. This will give you a copy of the code. In your copy of this repo, hit the [Code] button and open with Github Desktop. Bring it down somewhere in your c:\dev directory you’ll remember.

If you pull down the instructors code from Github, make sure you run, in the project directory:

npm install

## Otherwise

Open the project folder in VS Code and open an integrated termina. Start your project:

ng serve

Open your web browser to <http://localhost:4200> and inspect the running application.

# The Lab

## Narrative

You will create a new *feature module* in our application to track holiday gift giving. This will be modeled after the structure of the *media feature* we created together in class. E.g. you will have a *routed feature module* that is available from our Nav component. It will supply it’s own navigation routes, according to the list of features below. You will not be required to create effects or interact with an API. We’ll do that in the next course.

Please follow the link at the bottom of the page to see what you’ll create.

### Features

* The user can see a list of recipients.
  + Each recipients name and holiday are listed, along with the date of that holiday.
  + Only holidays in the future (today or later) are shown.
  + Sort the list by date of holiday. The soonest should be at the top.
  + Any holidays that have occurred in the past should be *hidden* by default. A navigation link can show a list of past holidays.
  + In the list, the user can place a check-mark (using a checkbox) to indicate they have sent the gift and/or card.
* The user can enter recipients. Each recipient has:
  + A Name
  + A holiday observed (Christmas, Hanukkah, Kwanzaa, President’s day, etc.)
  + The date of that holiday.
  + You can indicate that each user can get a card, a gift, or both, or neither (maybe you just want to remember to say “Happy St. Patrick’s Day!”)
* The user can see a list of past recipients
  + A route is provided to see recipients for holidays that are in the past.

# For a live demo of what you should build:

<https://few200lab.vercel.app/>