**CS 628 Full-Stack Development II**

**HOS07A: Reverse Engineering Practice 3 for the Home Page**

July 19, 2020

12/29/2020 Updated by Min

School of Technology & Computing (STC)

City University of Seattle (CityU)

**Before You Start**

* The directory path shown in screenshots may be different from yours.
* Some steps are not explained in the tutorial**.** If you are not sure what to do:
  1. Consult the resources listed below.
  2. If you cannot solve the problem after a few tries, ask a TA for help.

**Resources**

* Holmes, S. & Harber, C. (2019). [Getting MEAN with Mongo, Express, Angular, and Node, Second Edition](https://login.proxy.cityu.edu/sso/skillport?context=148121). Manning Publications. (ISBN 9781617294754)
  + 4.5 Taking the Data out of the Views and Making Them Smarter
  + Appendix C: Dealing with All the Views
    - Listing 4.10: The homelist controller, passing hardcoded data to the view
    - Listing 4.11: Final view for the homepage: app\_server/views/locations-list.pug
* Pug, <https://pugjs.org/api/getting-started.html>

**Course Concepts**

* Understand the roles of Views and Controllers.
* Apply the process of taking the data out of the Views
* Make the Views smarter if necessary.

**Activities**

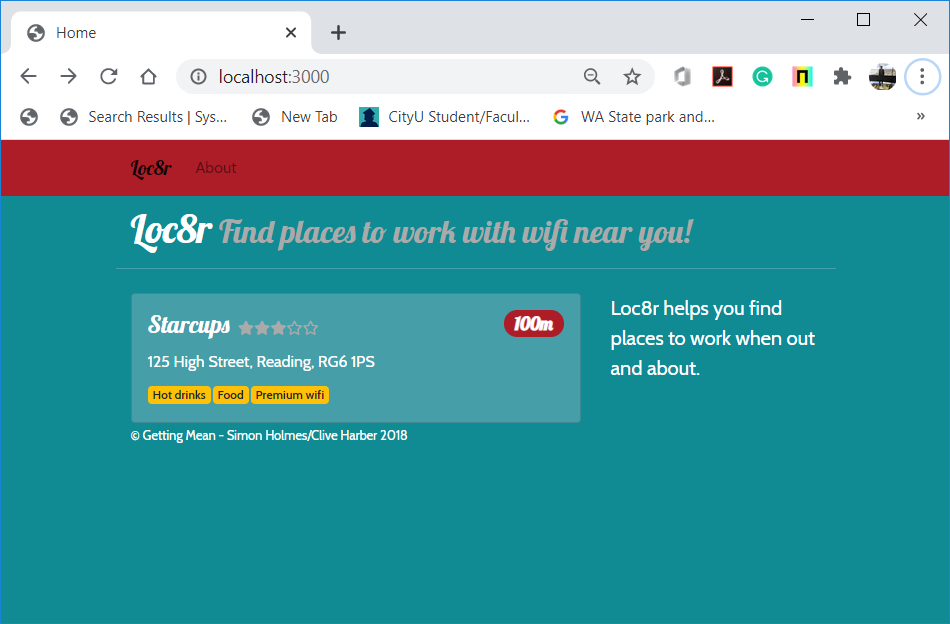
* Preparing for a MEAN project environment
* **Reverse Engineering the Home page**
* Source Code Management and Deployment

**Preparing for a MEAN project environment**

1. Copy the “module06b” directory to the “module07a” one.
2. Open the “module07a” folder in the VSCode.  
   Close the “Welcome” window.  
   Open the terminal (Ctrl+`).
3. Test whether the application works correctly.  
   >>>npm start

|  |  |
| --- | --- |
| <http://localhost:3000> | <http://localhost:3000/about> |
| <http://localhost:3000/location/> | <http://localhost:3000/location/review/new> |

**Reverse Engineering the Home page**

1. Discover what data need to be moved to the about controller from <http://localhost:3000>You can find 3 areas for reengineering: A) Page header, B) Location, and C) Sidebar  
   Area B needs more detail programming in Pug.  
     
   

A)

B)

C)

1. **Page Header**
2. Look at the data in the locations-list.pug.

extends layout

block content

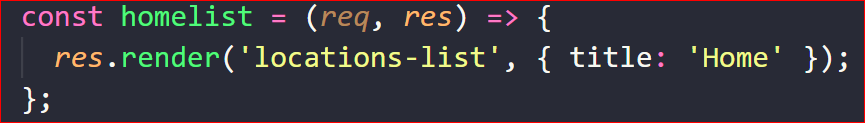
*.row.banner*

*.col-12*

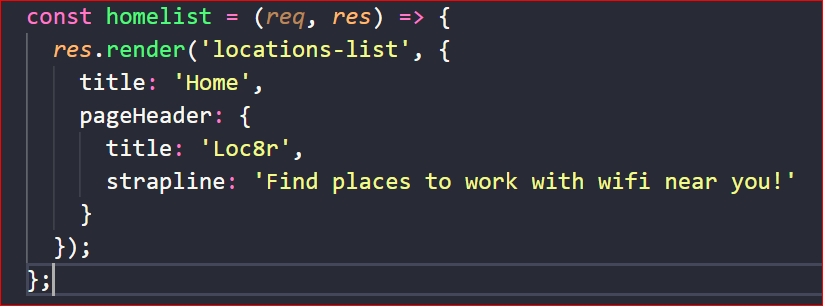
      h1 Loc8r

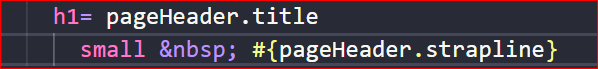
        small &nbsp;Find places to work with wifi near you!

1. Create a structure for that data in the controller – homelist in locations.js.  
   >>>Before



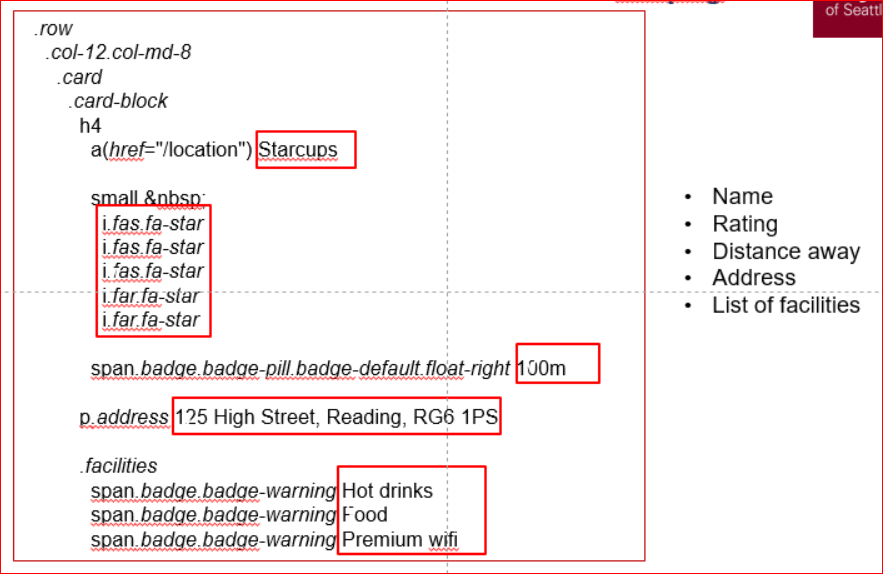
1. >>>After  
   For full information, check Listing 4.10: The homelist controller, passing hardcoded data to the view

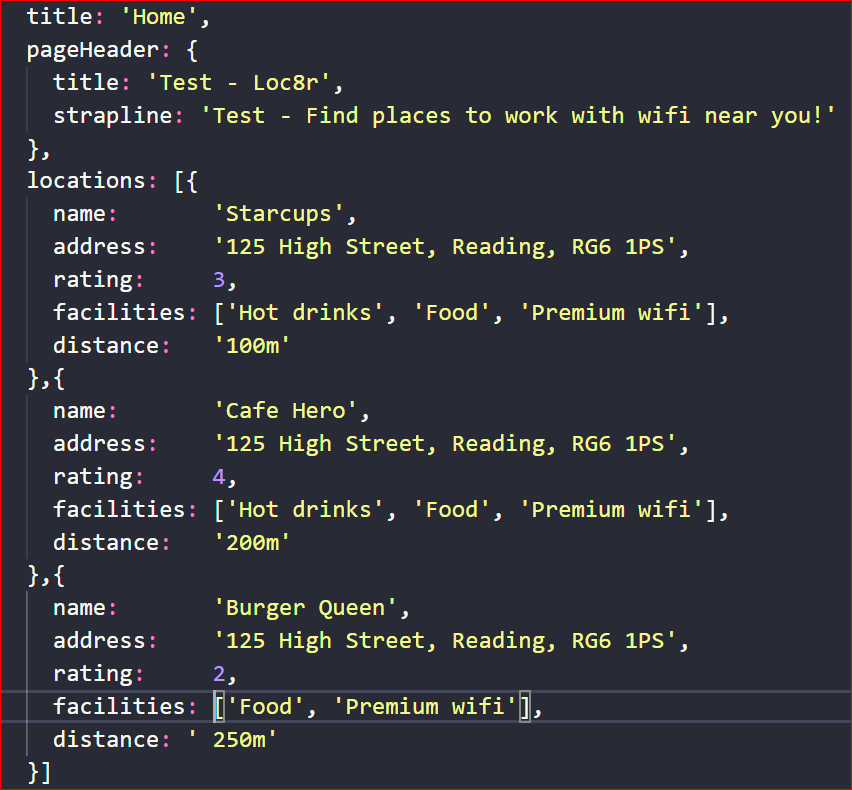


1. Replace the data in the view with references to the controller data – locations-list.pug.  
   For full information, check Listing 4.11: Final view for the homepage: app\_server/views/locations-list.pug  
     
   

1. Test whether the application works correctly.  
   No interfaces were affected.  
   >>>npm start

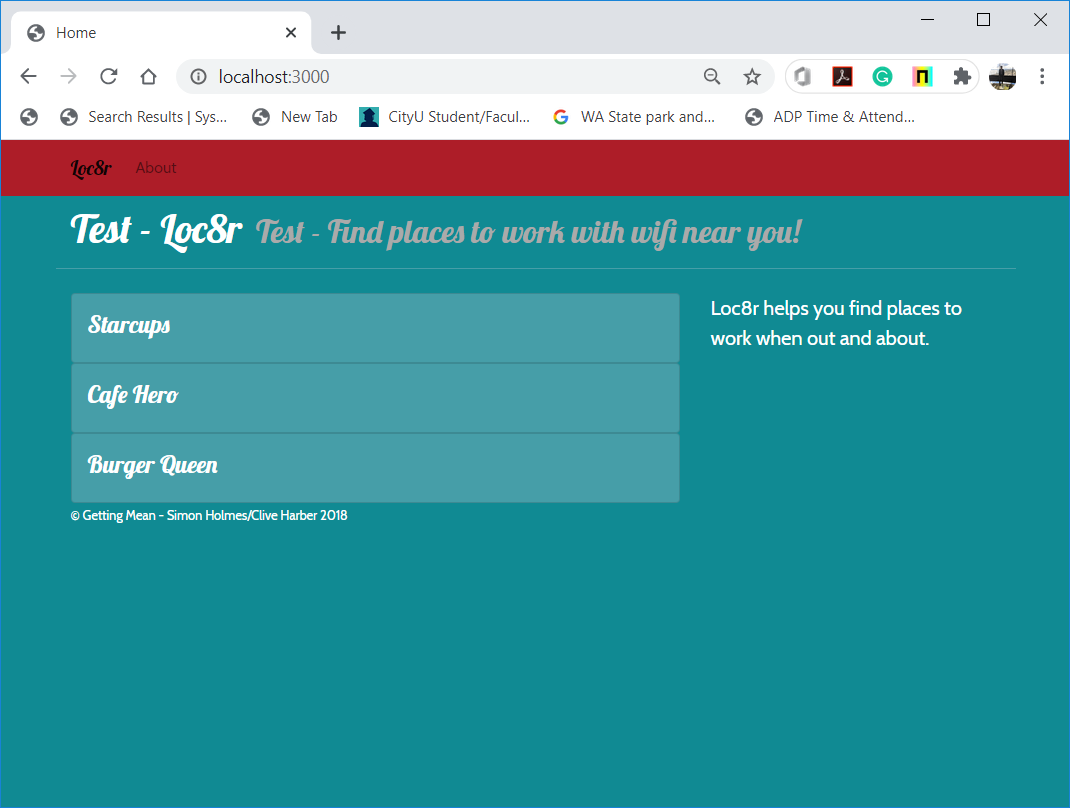
|  |  |
| --- | --- |
| <http://localhost:3000> | <http://localhost:3000/about> |
| <http://localhost:3000/location/> | <http://localhost:3000/location/review/new> |

1. **Location**
2. Look at the data in the locations-list.pug.  
     
   
3. Create a structure for that data in the controller – honelist in locations.js.  
   You can see that there are two more locations added for demo.  
   For full information, check Listing 4.10: The homelist controller, passing hardcoded data to the view



1. Replace the data in the view with references to the controller data – locations-list.pug.  
   For full information, check Listing 4.11: Final view for the homepage: app\_server/views/locations-list.pug

After each update, a quick test is recommended.

* 1. Location array & location name  
       
       
       
       
       
       
       
     ****

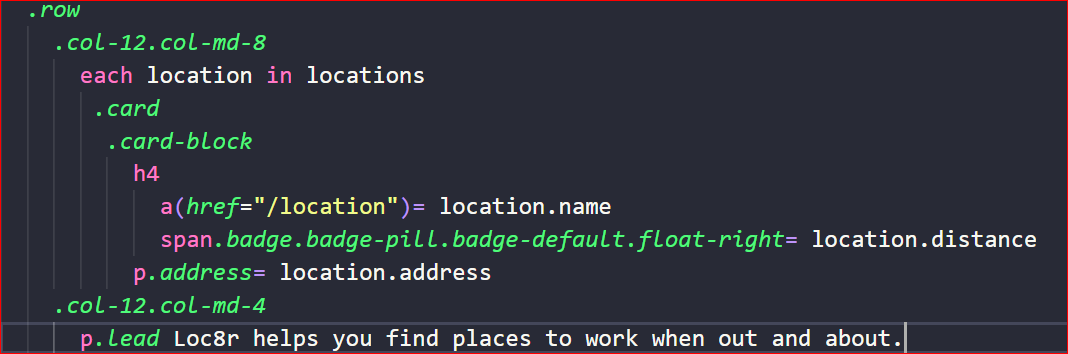
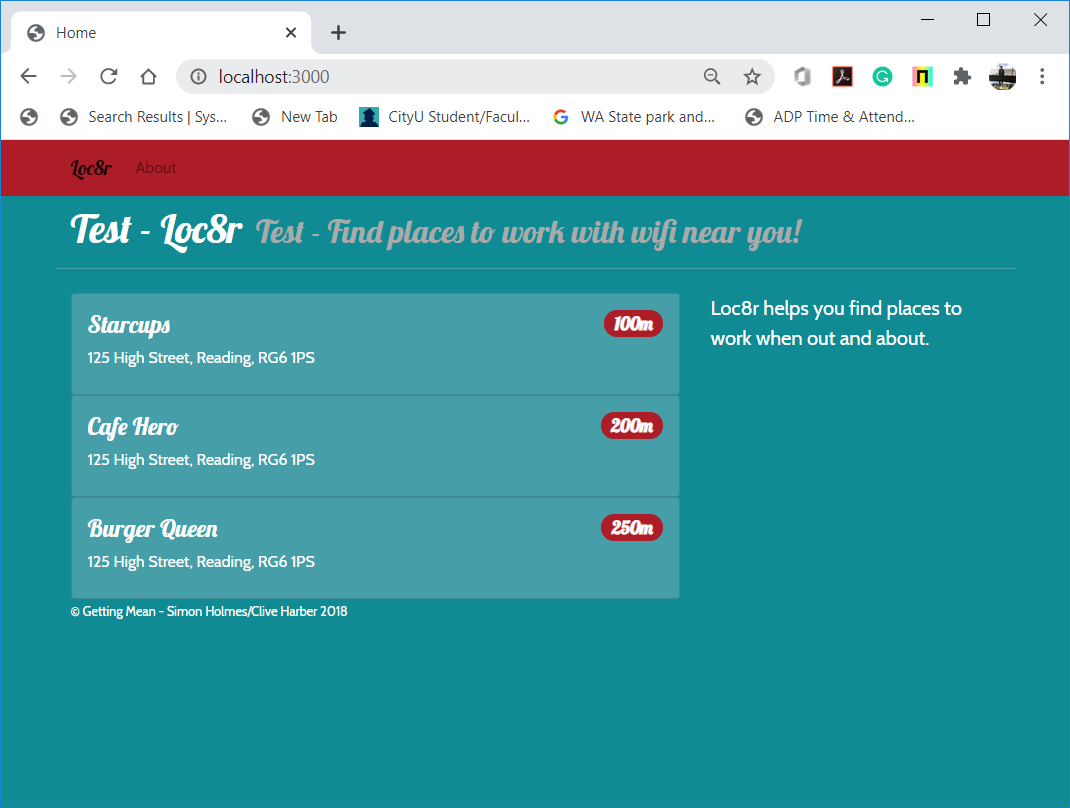
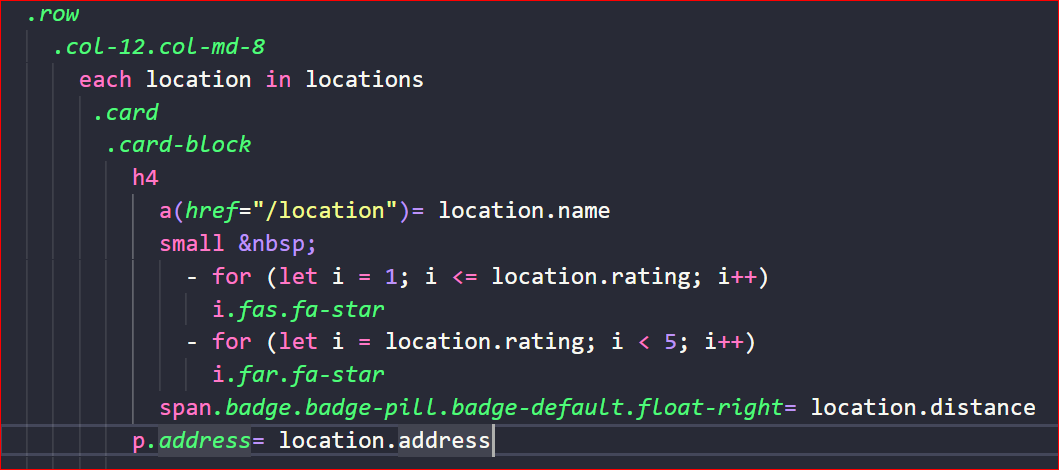
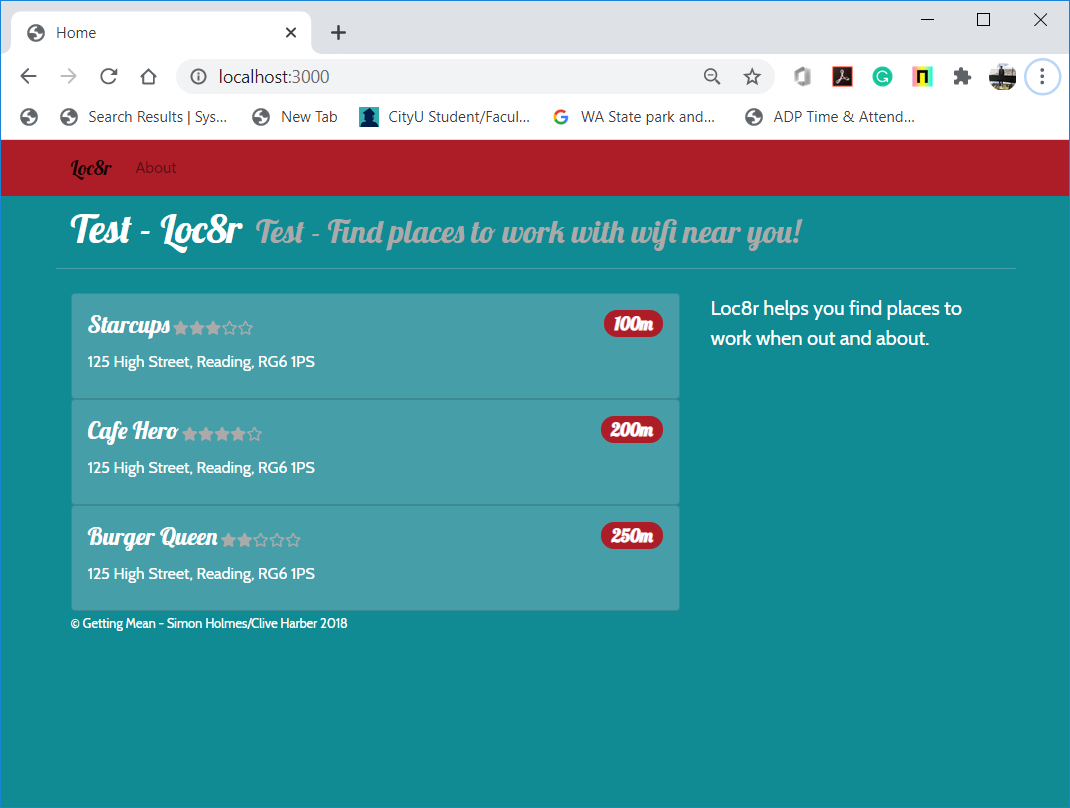
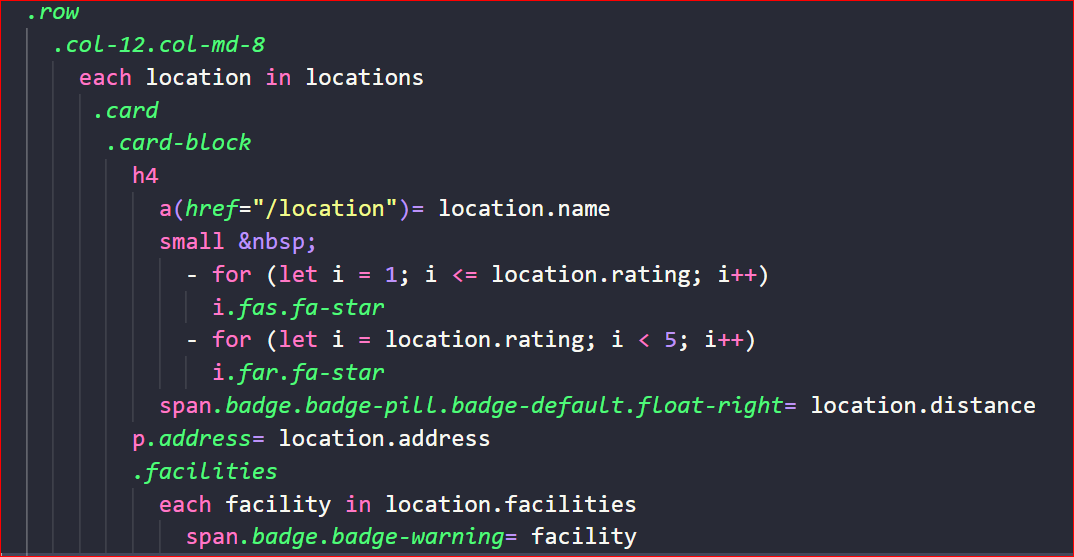
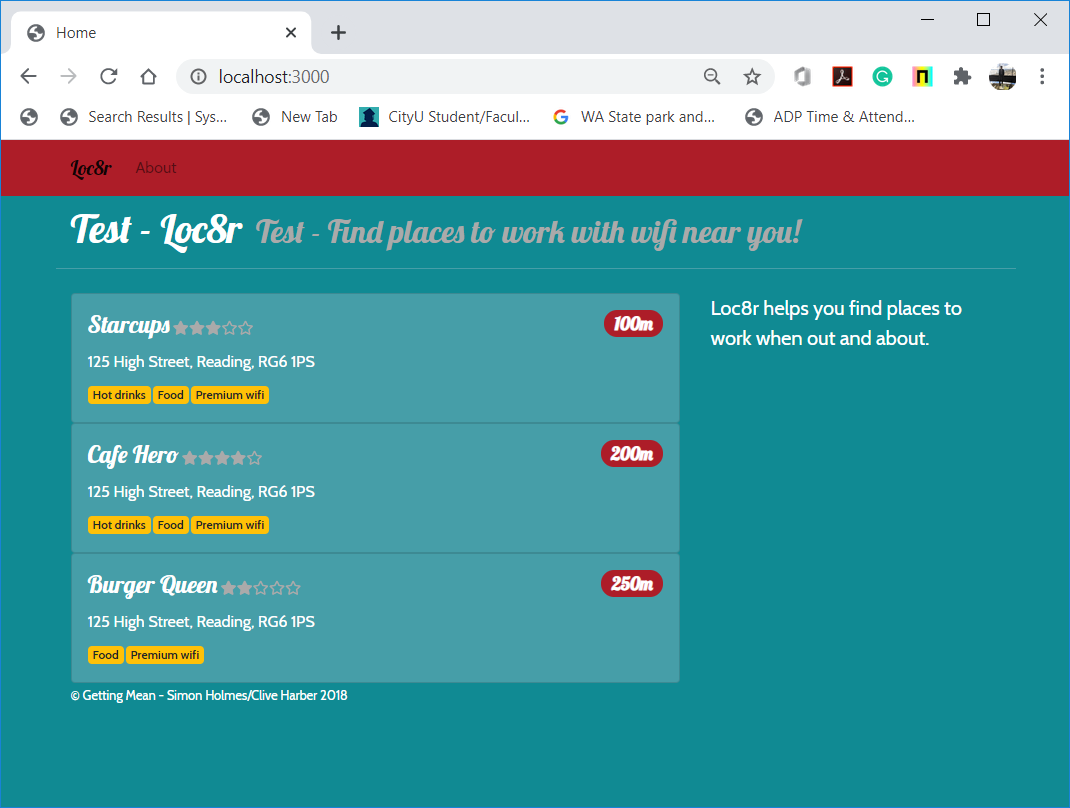
**each location in locations**

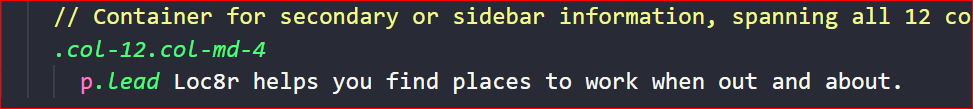
*.card*

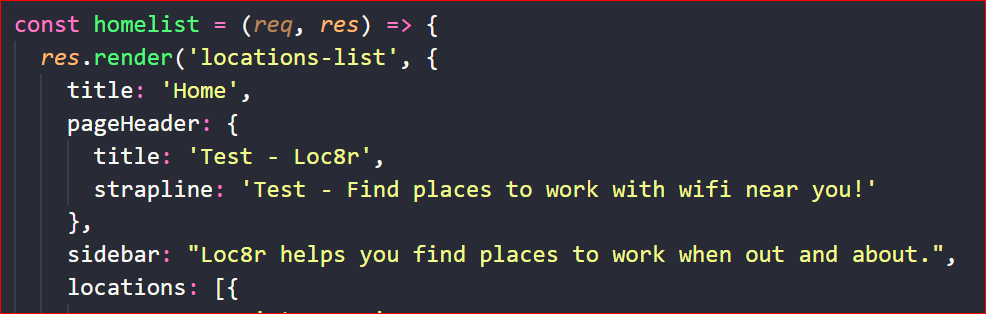
*.card-block*

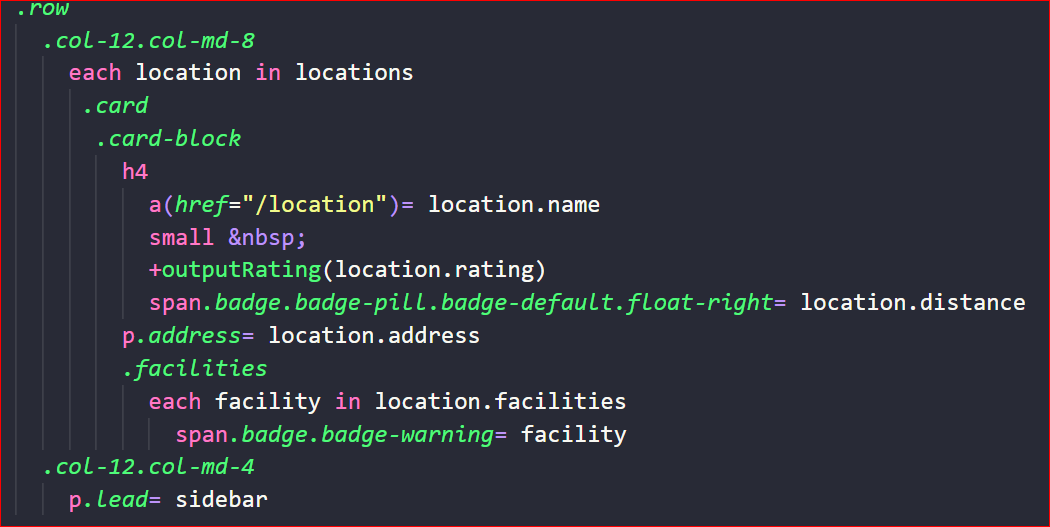
          h4

**a(*href*="/location")= location.name**

* 1. Location distance and location address  
       
       
     
  2. Location rating  
       
       
       
     
  3. Location facilities  
       
       
     

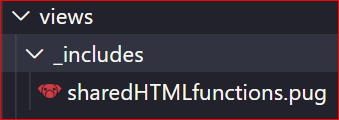
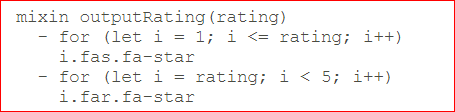
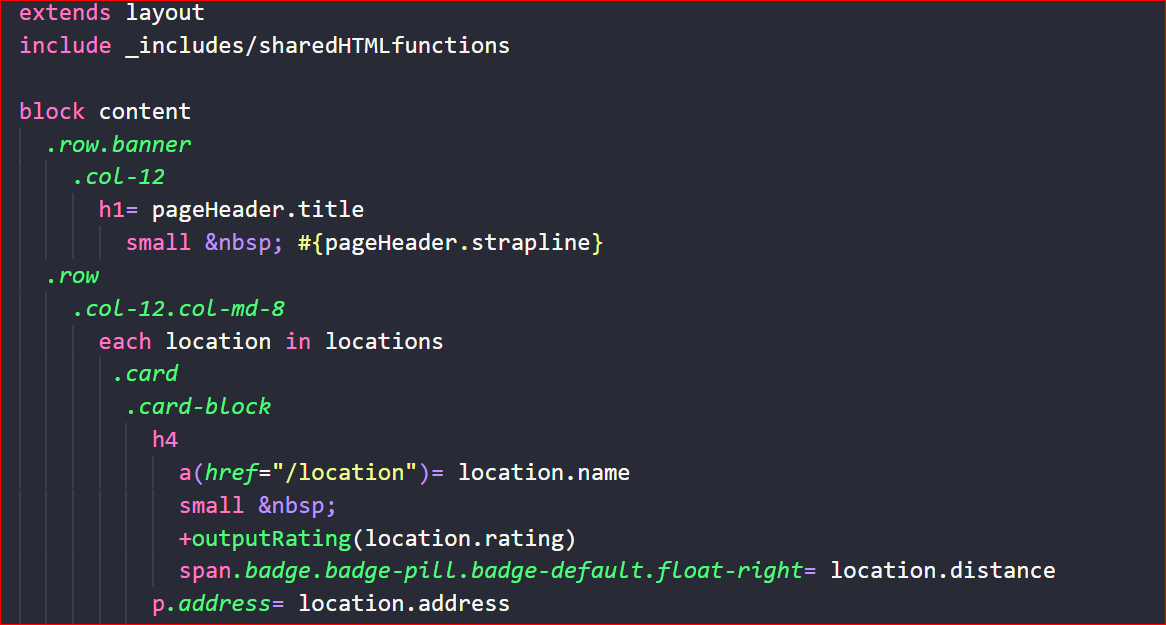
1. **Sidebar**
2. Look at the data in the locations-list.pug.  
   
3. Create a structure for that data in the controller – homelist in locations.js.  
   For full information, check Listing 4.10: The homelist controller, passing hardcoded data to the view



1. Replace the data in the view with references to the controller data – locations-list.pug.  
   For full information, check Listing 4.11: Final view for the homepage: app\_server/views/locations-list.pug  
     
   
2. Test whether the application works correctly.  
   >>>npm start

|  |  |
| --- | --- |
| <http://localhost:3000> | <http://localhost:3000/about> |
| <http://localhost:3000/location/> | <http://localhost:3000/location/review/new> |

**Look for opportunities to reuse code.**

1. Create a directory called “\_includes” under the “views” directory.
2. Under the “\_includes” directory, a create “sharedHTMLfunctions.pug” file.  
     
   
3. In the file, create a mixin outputRating() by copying the location rating source from the “locations-info.pug” file.  
     
   
4. Revise the “locations-list.pug” file by using the mixin.  
     
   
5. Test whether the application works correctly.  
   >>>npm start

**Source Code Management and Deployment**

**Push your work to GitHub**

Run the following commands to push your work to the GitHub repository:

Open the terminal from the VSCode by hitting the control + ~ key and type the following command:

>>> git add .

>>> git commit -m “Submission for HOS07A --yourname”

>>> git push origin master

**Push your work to Heroku**

>>> git push heroku master

If you cannot remember your branch name, run the command “git status” to check.