CIT 483/583 Lab – Sinatra/HAML

Fall 2018

**Instructions**

Open and save this file in any MS word-compatible format as Lab10\_*Firstname*\_*Lastname*.<ext> and place your answers in that document. Do a **Save-As** and retain all of my content. Keep the document safe in case your submission fails, or you discover an error prior to the due date and wish to re-submit. Submit your document to the Lab10 dropbox in Canvas. The due date and any other pertinent information are noted in the Canvas item.

This lab should be completed on students.cs.nku.edu as a reference implementation and as a means of testing your answers.

**Place the answers in or immediately following each question and make sure your answers stand out from the questions by using a different font color.**

First, copy recursively the directory Sinatra-HAML from my home directory on students.cs.nku.edu to a location under your home directory on students.cs.nku.edu. For Sinatra configuration questions, refer to <http://www.sinatrarb.com/configuration.html>.

**Some of you will still overlook it, but as soon as you learn to change the port for a Sinatra server, please do so to avoid port conflicts with other students working on the lab. Where you see 4567, substitute a randomly-chosen port above 2000.**

1. Concerning hello.rb:

1. First test it by running ruby hello.rb in one session and curl localhost:4567/ in another session. Capture the screen output in the curl session:
2. Alter the code to tell Sinatra to listen on any valid port greater than 2000 at your discretion. Show the code change you made and test it using two sessions and capture the screen output in the curl session. Now that you know how to change the port in a Sinatra server, use it for the rest of this lab to avoid port conflicts.
3. How would you alter the code to change the URL pattern to /hello ? Test it by running ruby hello.rb in one session and curl localhost:4567/hello in the second session. Capture the screen output in the curl session:
4. Run hello.rb as follows: ruby hello.rb 2>/dev/null

Based on the configuration documentation, why did the output disappear in the terminal running hello.rb?

2. Concerning the code under the mwf\_example subdirectory:

(a) There is a problem in the given code in the nesting of the <tr> and <td> elements. Run ruby mwf.rb in one session and curl localhost:4567/mwf in another and capture the output of the session running curl. Notice the <td> elements are not nested properly in the <tr> element:

<!DOCTYPE html>

<html>

<body>

<table>

<tr></tr>

<td>Mon</td>

<td>Wed</td>

<td>Fri</td>

</table>

</body>

</html>

(b) What file would be the source of the error?

(c) Fix it by editing that file to make the ary.each iteration create a new <tr> element that contains a <td> element. Run ruby mwf.rb in one session and curl localhost:4567/mwf in another and capture the output of the session running curl.

3. Concerning the code under the template\_example subdirectory:

1. Run main.rb in one terminal, and run curl localhost:4567/my\_template in a second session and capture the output.
2. Study three files (main.rb and the two .haml files). Why is weather.haml consider the "main" haml file?
3. Does it appear that layout.haml must be specified in the code to be part of the configuration?