|  |
| --- |
| **Make a new document using the naming convention HWFinal-MUid.docx (example: HWFinal-johnsok9.docx).**  **Objective**: The objective of this exercise is to:  Use all of the skills learned in the entire course.  **Submit**: screenshots, your word document, html files and links to your web page(s)  You **may** discuss the concepts with your fellow students  You **may** **not** show or share code with your fellow students  You **may** **not** show or share code with internet sources  You **may** discuss this with your instructor or TA. |

**NOTE: This assignment includes a mandatory presentation by you to the Professor, along with submission to canvas. Any assignments which are not presented will receive Zero for the entire assignment. This must be working on your openstack instance.**

**The programming part of the final project is a three-page website. An introduction (landing) page, a weather page, and a history lookup page. These pages are described below. Any sections which are left out (such as the actual layout of the page) are up to your discretion.**

# Part #1: Create GitLab project

*Estimated time: 10 minutes*

**Exercise:**

* Gitlab
  + Create your final project in gitlab
  + Clone it into your cse383\_projects directory in /var/www/html (will need to use sudo)

# Part #2: Client Front End

*Estimated time: 2-3 hours*

**Exercise:**

The first part will be a multi-page/multi-tab web client using html, JavaScript, bootstrap and CSS. You must make a good looking, pretty, responsive, web page(s) using bootstrap for formatting and your own CSS to make it look great. The page must degrade nicely which I will verify by shrinking it from a large desktop so that it fits down to phone size. Degrading means it must change, for example from 3 columns down to 1.

Note: Everything you write must use the following (from the exam)

* Concepts
  + Html
  + CSS
  + JavaScript
  + Bootstrap (must be responsive)
  + jQuery
  + Ajax
* Style
  + Good looking using CSS
  + **All CSS, ajax, JavaScript must be external files**
  + Every page (including the menu) must include pictures that integrate well into the web page
* Location
  + All code MUST be working on your openstack instance http server
  + All code must be in git
* Ajax
  + **You are to write all code using the jQuery Ajax routines as done in the lecture, labs, and homework. Some examples you see will use other methods, notably the fetch command. You may use those as an example to review, but you MUST use the $.ajax call for credit.**

The menu must provide the following items.

* Menu of pages you are providing (with a mechanism to switch pages) **You are to research and decide on your own mechanism** to provide the front-end menu of pages.
  + This was not covered in class. However, you have already seen web sites for research which included them.
  + **Do NOT just add buttons or links to the top of a page (hint: nav), you must use a navigation element**
* 3 Pages
  + Landing page (default page)
    - Your information (name, class, assignment etc…)
    - Include picture and explanation of your system
  + Weather
  + Directions History and search

# Part #3: Weather App (using TomTom, OpenWeatherMap and your OpenStack rest server using SqlLite)

*Estimated time: 7-12 hours*

**Exercise:**

1. Use the TomTom api you created previously to allow for a fuzzy location (such as “Benton Hall in Oxford ohio” to get the latitude and longitude
2. Call openweathermap to get the 5 day forecast using the latitude and longitude from TomTom
   1. Display the 5 day forecast using your smart phone’s weather forecast as an example
      1. For each day
         1. Date
         2. Day of week
         3. High
         4. Low
         5. Forecast (weather description)
            1. Use icon to display the image
            2. Definition: <https://openweathermap.org/weather-conditions>
         6. Visibility
         7. humidity
3. **Store all information about the request (and response(s)) using the php rest server from lab13**
   1. **Store your own JavaScript object by creating the json version of the object and storing that in the value field**
   2. **Finally – save that entire object to your database along with the tomtom fuzzy search paremeters**

# Part #4: Request History

*Estimated time: 2-6 hours*

**Exercise:**

Provide a page where the user can enter a date and max number of lines and then receive a list of all weather requests made on that date in a nicely formatted output (table), showing all available data, with not more than the requested number of lines.

* You will be calling the updated rest interface on your openstack instance using ajax
* *http://IP/final.php*
  + *Method:getLookup*
  + *Date:yyyy-mm-dd*
* You must show the correct number of returned results (based on the input page)
* For each returned request, use the json object you sent in part 3 to show the following:
  + Date
  + Time
  + Fuzzy search input (such as “Benton Hall Oxford ohio”)
  + Latitude and Longitude
  + Full weather data
* The user should be able to click on one of the lines (make something selectable or add a button)
  + This should then display the same information as in part 3, but using the data from your json object
    - Do **not** call TomTom again for latitude/longitude.
    - Do **not** call OpenWeatherMap again, use the stored results to show what was done on that day

# Part #5: Write a new Word Document

*Estimated time: 45 minutes*

**Exercise:**

* Make sure it includes your name, class, assignment etc…
* Explain (relating to this assignment and the entire course) in **full paragraphs** with **full sentences** for **full credit**.
  + what worked
  + what didn’t
  + how long it took
  + What you liked
  + What you didn’t

# SUBMIT:

* Word Document
* **Mandatory Presentation of working system in zoom session**
* Upload all code
* Git all code
* Paste the URL of your menu page into the submission as a comment