Internet and Web Technologies (CSCI 355)
Spring 2019
Term Project
(Revised: February 12)

Overview:

The term project for the Spring 2019 semester is to build a "Search And Reporting Application" (SARA). The project will be implemented in the following series of phases (that require an increasing skillset of web technologies):

1. Create a website which will be used to house SARA, as well as other pertinent information (see "Website/Homepage" below).
2. Display hard-coded search results about a topic of your choice.
3. Display search results read from files stored in CSV, JSON, and XML formats.
4. Display search results obtained from using the Google Search API (see e.g. https://developers.google.com/custom-search/v1/overview)
5. Display search results from your own search engine which indexes the entire content of webpages. (Details about the search engine will be provided later in the semesters when we discuss databases and server-side programming.) Administration screens related to this search engine will also be needed.

   In addition, each displayed search result should have a checkbox next to it, and users should be able to save whichever search results they have checked to any of the three file types mentioned above. (The standard "Select All" and "Deselect All" options should be provided.) This write feature should be implemented beginning with Phase 3 where reading from such files is supported, but should also work for Phases 4 and 5.

   Regardless of the phase – i.e. the source of the search results (hard-coded, file, Google API, or custom search engine) – the presentation styling should be the same. This should all be controlled using an external stylesheet (CSS).

Guidelines and Methodology:

- Students may work individually or in pairs ("peer programming"). Individuals enrolled in different sections of CSCI 355 (taught by Teitelman) may work together as a pair.
- Each student (or pair) must complete a form - to be made available soon - which includes their names and their project URL (on Venus or other site). This form will also be used by the instructor to track progress on the project, both in terms of satisfying requirements and meeting milestone dates.
- In addition to the option for peer-programming, we are using an "agile" approach which has incremental, increasingly more difficult deliverables (prototype, spiral models), and also encourages regular in-person communication with the "user" (instructor) to get feedback on the project/product in progress and evaluate modules completed to date.

Website/Homepage:

The home page should have at least the following four high-level menu options.

- **Course**, with a submenu item to point to the course Zybook. Additional course-related submenu items may be added later.
- **Search** – with submenu items to the various search modes specified in the Overview. This will obviously be the most important functional area of the site.
- **Browser** – with submenu items linking to pages with information about the browser, window, screen, location, geolocation, etc. (see https://www.w3schools.com/js/ and scroll down to "JS Browser BOM" in left nav bar) to get ideas as well as relevant code snippets.

- **About** – with submenu items to descriptions of the student-developers of the page and a "Contact Us" to contact them by e-mail. (If you don't want your real email posted, you can create a Mailinator or other emailing alias system that will forward to your real email.)

The website should use "responsive web design" (RWD) – so that it can be easily viewed on mobile devices, as well as on larger screens such as laptops and desktops.

Here are some helpful links with regards to menus generally and responsive ones in particular:

https://www.w3schools.com/css/css_navbar.asp
https://www.w3schools.com/howto/howto_js_topnav_responsive.asp
https://getbootstrap.com/docs/4.0/components/navs/
https://getbootstrap.com/docs/4.0/components/navbar/

Scoring, Due Dates, and Lateness:

The total score for all components of the project will be out of 110, which will be divided by 5 to obtain the 22% of the final course grade allocated to the term project.

Rather than penalizing for individual lateness (and dealing with individual requests for extensions), 20 of the 110 points will be allocated to *general punctuality* in meeting milestone dates. Thus, lateness on any one phase will have negligible impact, but consistent lateness will indeed bring down one's final grade in the course.

The remaining 90 points will be approximately allocated as follows:

- (20 points) Phases 1 and 2 – Website and Results Display, due date: March 7
- (20 points) Phase 3 – Download/Upload , due date: March 28
- (15 points) Phase 4 – Search via API, due date: April 16
- (35 points) Phase 5 – Search Engine, final due date: May 9

Hosting:

Each student will need a home page in their Venus account, or another web-accessible server of their choosing, to serve as their course dashboard for the various deliverables mentioned above. If you are using Venus, see the instructions in this section. For other hosting options, you will need to adapt these instructions to your hosting service.

Step 1: Login to the account on Venus.

(For remote access, Windows-users will generally use SSH (see https://www.ssh.com/ssh/download/) while Mac-users will generally use Terminal.)

These are the connection credentials:

- host-name: venus.cs.qc.cuny.edu
- user-name: LLFFNNNN
  (where LL = first two letters of your last name, FF = first two letters of first name, NNNN = last four digits of CUNYFirst Id)
- password: the full eight digits of your CUNYFirst Id

Step 2: Create and set permissions on the relevant directories

Grant access to the root with the command:

```
$ chmod 755 $HOME
```

Create a new directory and grant access with the commands

```
$ mkdir public_html
$ chmod 755 public_html
```

Switch to the public_html directory, create a subdirectory “cs355” (lowercase “cs”) for our course, and set permissions with the commands:

```
$ cd public_html
$ mkdir cs355
$ chmod 755 cs355
$ cd cs355
```

You can use “ls –l” (that is: el es minus el) to see the file/folders and their permissions.

Step 3: Create the homepage placeholder

Open an editor of your choice
Paste in this text

```
<!DOCTYPE html>
<html>
<title>CS 355 Home Page</title>
<body>
<h1>This is my home page for CS 355</h1>
</body>
</html>
```

Save the file as “index.html” (all lowercase)
If you are editing on your own computer, upload this file to the cs355 directory created in Step 3.
Change the permissions on this file too:

```
$ chmod 755 index.html
```

Step 4: Test your webpage

Open a browser of your choice
Type http://venus.cs.qc.cuny.edu/~LLDDNNNN/cs355/ (where LLDDNNNN is as defined in Step 1)
You should see your homepage.
If you get an error about access/permissions, there is probably an issue with what happened in Step 2.