

CMSI 370-01

INTERACTION DESIGN

Fall 2015

Assignment I029

OK, it's time to write some *dynamic* user interface code. Resources to help you with this include many of the links listed on the course website, assorted samples from the *bazaar* repository, and the starter files produced in class that are pushed to your respective repositories.

Outcomes

This assignment will affect your proficiency measures for outcomes *3a*, *3b*, and *4a–4f*.

Background Reading

For this assignment, you will want to be acquainted with one or more of the web development resources listed in the course website, particularly the Mozilla Developer Network site, the specifications that can be found with the W3C, the jQuery website, and Bootstrap documentation.

Web browser developer tools will also be quite useful, as would be *jsFiddle* when doing some experimentation or trying out new things.

Textbook reading is centered on the menus, forms, and dialogs interaction style, which would be Shneiderman/Plaisant Chapter 6.

For the assignment itself, the JavaScript textbook will be of help, particularly Chapter 6, Section 7.5, and Sections 8.2.3, 8.3, 8.4.1, 8.5, and 8.7. For the best exposure, read Chapters 6, 7, and 8 in full.

For Submission

A Front End to an Existing Web Service API

The title pretty much says it: put together a front end for a pre-existing web service API. Many such APIs are freely and publicly available, although many do require some degree of signup. For this assignment, you are to:

- Choose such an API (selected examples will be shown in class)
- Design and implement a simple front end that invokes the API's services

Specific functionality will vary depending on the API, of course, but for proper scale, we are looking for the implementation of *at least five (5)* non-trivial web service functions.

A “home-grown” option is available in the form of the *LMU Diabolical* web service that was developed by the CMSI 486 class of fall, 2012, with support from a Google education grant. The service and some documentation is located at <http://lmu-diabolical.appspot.com>.

If you choose this as your back-end service, your front end should implement the following:

- *Display a list of current characters*
- *Spawn/create a new character*
- *View a character*
- *Modify a character*
- *Delete a character* (make sure the user does not delete one by accident)
- *Spawn a random item*: For testing purposes, you may invoke this function directly; in a real game, this would be replaced by events like a character finding, shopping, or trading for items

For other web service APIs, it is highly recommended that you check with me first on the five or more operations for which you plan to supply a front end. Better safe than sorry.

Finally, we do want a good user interface, but don't worry about fancy graphics—we know that those require a different skill set. Instead, look to make good choices in user interface elements, layout, and behaviors. Some visuals are sufficiently easy with CSS; you are not obligated to go beyond that.

How to Turn it In

Commit your front end files under a directory called *front-end/* within your repository. To evaluate your work, I will run a simple HTTP server from my local copy of that directory and use your application from a web browser. As such, you will need to set things up so that your selected web service API can be invoked under that configuration—for security reasons, such setup is not trivial, but by and large doable.