

## HW5

## App Overview

The app I made this week is just a jQuery mobile site that lists the information on the IS core classes (250, 272, 371, etc) on different pages. It can be found on the following site:

<https://mobile-bihequ.backliftapp.com/basic/index>.

## Pages

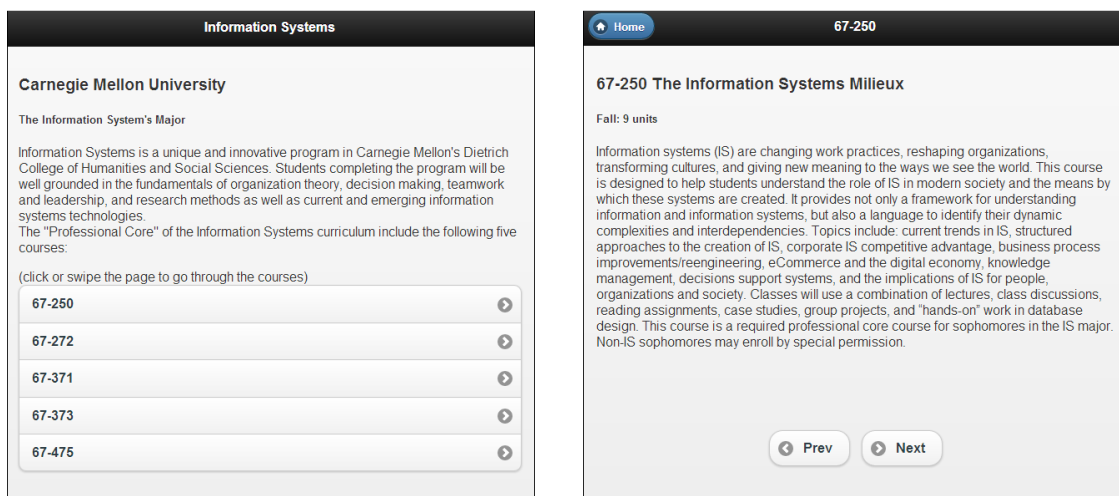
The index, 250, and 272 pages are all loaded together in the same file. When the index.html file is loaded these pages are all loaded too.

The other course pages, 371, 373, and 475, are all loaded in via AJAX using jQuery mobile when either their link is clicked or they are swiped to.

## jQuery Mobile Components

I used several jQuery mobile tools in this app, most of which I got from the jQuery mobile demo site:

1. There is a page header on top of each page, which includes the name of the page and also a button to go back home, to the index.



2. That home button in the header uses the “home” icon provided by jQuery mobile. I used some other icons for the nav buttons, the arrow-l and arrow-r icons.
3. The home button has a different theme (it’s blue) which is provided by jQuery mobile.

4. The buttons on the index page are contained within a linked list, which stacks them and gives them those little arrows on the side.



5. I implemented swipe events so that a user can swipe between the course pages, kind of like an ebook.
6. The transitions from page to page differ depending on which pages you go between. If you are traversing the pages sequentially (like by swiping) the transition is a slide left or right, depending on which way you swipe. If you are jumping to a “non-adjacent” page it uses the flow transition. I hoped that the sliding would give the user a good mental picture of the courses, and that they should be viewed in a certain order.

## Mobile Terms

### Progressive Enhancement

Progressive enhancement is the idea of building an application with only the components that are compatible with all types of browsers or devices, and then making more advanced features available based on what the application detects the device can handle.

JQuery mobile says, on its website, that it is built on a “progressive enhancement approach brings core content and functionality to all mobile, tablet and desktop platforms and a rich, installed application-like experience on newer mobile platforms.” I’m not exactly sure what this means, but I’m guessing it means that jQuery mobile plays nice with all the browsers because it only makes features available based on the browser and its settings, like if it has Javascript enabled or anything like that.

### Responsive Design

Responsive Design is the idea of designing a web page or application to be able to adapt to whatever device is running it, and to do so in an appealing and nice way. So instead of making separate pages for different kinds of devices, you equip one with the structure and tools to change based on what device it is on.

In my app the only responsive design, besides the auto resizing feature provided by jQuery, is the navigation buttons found on the bottom of the page. These buttons appear based on the browser window width. This is done by doing a media query in the CSS file:

```
@media all and (max-width: 620px) and (min-width: 0px) {  
    .nav-buttons{display: none;}  
}
```

This is a simple way to track browser size, and here I'm kind of assuming that narrow browsers are mobile devices, but there are other ways to see what features are available on the device and to see what kind of device is being used.

## **Semantic HTML**

Semantic HTML is the practice of using HTML tags to label parts of a webpage correctly so that it is structured better and it is easier for others to see and work with your code. This might be really important if you are using a webpage framework with custom CSS and javascript, like bootstraps or jQuery Mobile, because labeling parts of the webpage translates to certain styles and actions. So having the wrong label might really mess up your page layout and other stuff. Also, it is better to format all your tags with CSS, and not the style attribute.

In my application, I followed jQuery Mobile's naming structure, which includes many attributes that have names like "data-role" and "data-theme" that are all formatted by jQuery's CSS and javascript, which is a use of semantic HTML. My HTML looks very simple, but that is because all of the styling is done by jQuery's CSS and js files, so I only have to label my content correctly and the framework will do the rest, which is the goal of semantic HTML anyways. I did, however, use my own CSS to do a media query which decides to show the nav buttons or not depending on browser width. For this, I labeled the div class "nav-buttons" so that they could be grouped together.

## **Mobile First**

Mobile first is the idea of developing an application or website or whatever to be displayed and utilized by mobile devices (either as a native app or on a browser). The idea behind doing this is that mobile device use is growing vastly, and that most people will be using tablets or smartphones to use applications so it should be that developers focus on making fully functional mobile apps first since most of their users will be on them. It is certainly true that more tablets and smartphones are being used, but it is also the case that many features available on mobile devices can add to an application. But the application can still function without it, meaning a desktop version won't be hindered by running a mobile application. Also, a lot of times mobile applications are slimmed down to only the essential functions and maximize user understandability, which can help many development teams decide what is really important for their app and what is not so important in case they do decide to make a desktop version.

Since my app uses jQuery Mobile, it is already set to a “mobile first” agenda. JQuery Mobile uses a design that is mobile friendly, in that it adjusts to the size of the browser window and features many functions, like transitions and finger-event handlers that utilize the mobile platform features. In my app you can see the swipe event being handled by swiping the pages left and right. It is there for mobile users to easily navigate the page. Desktop users can also swipe, but it can be awkward which is why there are “next” and “prev” buttons, so that swiping is not the only way to navigate. My pages also rescale to the size of the browser window, thanks to the mobile framework.

