Block 2 Challenge Proposals

Start Assignment

Instructions

1. Design and build one or more web applications to explore and apply the things you have learned thus far.
2. As this is the primary work you will do over weeks 10-14 you should plan on spending at least 30 hours on these challenges.  Work with the instructor to manage the scope for your ideas.
3. Once you have settled on an idea or two write it up and submit it here. Because these should be more complex challenges, you should spend more time on this than you may have for the first block to produce a more formal proposal.  A sample outline might include the following sections:
   1. Purpose
   2. Audience
   3. Data sources
      * External API
      * localStorage
      * local JSON file.
      * etc
   4. Initial Module list
   5. Wireframes for each view of your application
   6. Colors/Typography/specific Element styling
   7. Schedule to provide yourself mile markers along the way to help you stay on target.

Reminders

Final Project Grading Rubric

This will let you focus on the topics this semester.

Meaningful usage of the following topics

**30%        Third party APIs**

More than a wimpy Chuck Norris API is not going to cut it ([https://api.chucknorris.io/jokes/random (Links to an external site.)](https://api.chucknorris.io/jokes/random)). There are MANY great APIs for weather, movies, travel, sports, NASA, movies, music, Anime, search, GPS, maps. Check out [https://rapidapi.com/marketplace (Links to an external site.)](https://rapidapi.com/marketplace). I expect an API should return a good rich JSON result set. Something with an array you can iterate through. Weather for the next 8 days, 10 matching movies, song list for an artist, books from an author, pictures from a Mars rover, sports scores. BTW, if you do this you will also get the points for JSON and it is impossible to do a good job here without some serious JavaScript. APIs+JavaScript+JSON and you already have 80% of the points!!

**30%        JavaScript**

I expect some honest logic going on here. Validating the screen data, looping through an array of JSON data to display to the screen, creating and using events, changing element styles with JS, changing element classes to use different CSS rules.

**25%        CSS**

Use Transforms to round the edges of your input fields, add shadows. Use Transitions to enlarge an input field on focus, and shrink it on blur, wiggle an input field when bad data is entered. Add borders. Don’t go overboard. CSS should subtly add style to a page, not whack you over the head.

**20%        JSON**

JSON is the *lingua franca* of our time. Don’t use it at your peril.

**15%        Events**

Use events to enhance the user experience; increase the size of the input field on focus or add a shadow. React to a button click. Initialized the page with data once the onload event triggers.

If you have CSS rules for pseudo-state-selectors (hover, focus, checked, after, before) I’ll recognize these as events as well.

**10%        LocalStorage**

LS is a nice to have. By design it is local to current device. It is not available as you move around from phone, to tablet, to laptop. But it is nice to know how to implement. It can add value to an app to remember the user from their last visit. Not a lot of points but fairly easy to add to an app.

Yes, this adds up to more than 100%. You can get 90%+ without using APIs. But you can see how biased this is towards using them. They are extremely useful and are ubiquitous in enterprise development.

Can I do some Server-side code in my app?

You are graded ONLY! On the client side. Some of you might want to have a server side with a database to drive your front-end. That is OK but you will **not** get any points for that part of your application. It could be 1000 lines of Node.js but this course is Web **Frontend** Development II. I just want to set expectations so when you are planning your project you will invest your time where you will get the most points.