ADVANCED WEB PROJECTS

14 | production

DESCRIPTION

Students will convert the kinetic mockup from the previous assignment to HTML, CSS, JavaScript, imagery assets, font files, etc., and publish the application to a public web server.

Deliverable 1: all necessary HTML files, CSS files, JavaScript files, image files, video files, audio files, and font files published to GitHub and a public web server.

Deliverable 2: a prepared 15–minute presentation walking us through your web application and documenting your process from beginning to end. What did you learn? What was challenging? If you could do the web application over again, how would you approach it knowing what you know now? What is the future of this web application?

OBJECTIVE

Students will tie together concepts that they have learned throughout the semester, leveraging technological systems to create dynamic and interactive visual and navigational systems. Students will also learn to solve the myriad kinds of visual and behavioral problems involved in authoring kinetic interfaces. Finally, students will gain a deeper appreciation for the relationship between databases of information and the interfaces, both visual and programmatic, through which we interact with that information.

DEADLINE

Both deliverables are due at the beginning of the last class. Please commit and upload your files to the master branch of your GitHub repository in a subfolder named “14 production.” Additionally, please upload the files onto a public web server; static files may be pushed on GitHub Pages. Please share the URL with me by creating a file named “url” inside the “14 production” folder. Websites

CLASS INFO

IDESN 3535
3 Credits
Fall 2016

MEETING TIME

Tuesday
1:15pm – 5:00pm
University Hall
Room L023

INSTRUCTOR

Saul Baizman
sbaizman@lesley.edu
617 863 0136

OFFICE HOURS

By appointment

WEBSITE

baizman.com/awp

ADVANCED WEB PROJECTS

requiring server-side dynamism (such as PHP files or files that use a database like MySQL) will need to be published on a web host of your choosing (to be discussed in class). Finally, please commit and upload your presentation to the master branch of your GitHub repository in a subfolder named "15 presentation."

***Note**: the homepage* must *be named index.html.*

WEIGHT

This assignment carries a weight of 2.

HINTS, TIPS, AND RECOMMENDATIONS

- You are not expected to build out every single screen and every single feature detailed in your mockups. There is a term in the industry – MVP – which is an acronym that stands for Minimum Viable Product. What's the least you can build to show how the concept works? My recommendation is to focus on your narrative scenario and build out the most common journey that a user will take through your application. For example, if there's a user login page and user registration page and password reset page, I would skip building the password reset page.
- There's no simple mapping between the mockups that you made and the HTML/CSS/JavaScript components into which your application will be converted. A good rule of thumb is to look at your application from a distance and determine what is going to be shared between pages / screens and what is going to be unique. Shared stuff should have a single source and be stored in an easily reusable way. For a simple example, you'll probably want all of your CSS to live in a single file, which your other pages reference between head tags. For a more complex example, you may have common headers and footers on every page. Wouldn't it be nice if you could update the headers for all pages in a single location? Using PHP includes, you can.
- Regarding the use of programming languages, you'll want to endeavor to use user-defined functions – in both JavaScript and PHP – as much as possible.

CLASS INFO
IDESN 3535
3 Credits
Fall 2016

MEETING TIME
Tuesday
1:15pm – 5:00pm
University Hall
Room L023

INSTRUCTOR
Saul Baizman
sbaizman@lesley.edu
617 863 0136

OFFICE HOURS
By appointment

WEBSITE
baizman.com/awp

ADVANCED WEB PROJECTS

CLASS INFO
IDESN 3535
3 Credits
Fall 2016

MEETING TIME
Tuesday
1:15pm – 5:00pm
University Hall
Room L023

INSTRUCTOR
Saul Baizman
sbaizman@lesley.edu
617 863 0136

OFFICE HOURS
By appointment

WEBSITE
baizman.com/awp

This may include making a number of very abstract snippets of code. More abstraction means it's harder to write (and read), but often brings with it more power and flexibility. What you want to avoid is updating similar bits of code in multiple locations because it can make maintenance difficult and introduce errors and inconsistencies in your application.

» If you're not sure where exactly to begin authoring your application, it may help to create a list of the various bits of functionality / features in your application. Then rank the items from easiest to hardest to build. Work on the easy parts first, and when you've made decent progress, stop and work on the tough parts for a little bit. Then, when you get stuck on a hard part, switch back to the easier parts. What this strategy is designed to do is to prevent you from getting stuck for prolonged periods of time or from getting discouraged. If you are having serious issues at any point, and you've spent more than 45 minutes trying to solve a problem, please stop what you're doing, commit your work to GitHub, and send me a message on Slack. Speaking of GitHub, don't forget to use branches to experiment and try different solutions! They cost you nothing, and they're more elegant than a thousand Undo commands.

» In terms of authoring, write a little bit of HTML/CSS/JavaScript/PHP, then test it. Maybe even no more than a couple of lines of code at first. It's a lot easier to troubleshoot two lines of code than ten or twenty. Try and create lots of tiny goals that are easily accomplished rather than a couple of large goals that are very difficult to accomplish.

» Don't forget to use the various tools' error identification capabilities. For example, PHPStorm will bark at you if so much as a comma is missing, and JavaScript will whine at you in the Developer Console if it can't find its keys or hat.

» Above all else, be patient, particularly with the complex parts. Most of programming is just syntax and logic. They both need to be correct for your application to work.