The JavaScript Workshop

Building a Professional Portfolio

If you're a professional developer, or want to become one, a project portfolio is one of the best ways to demonstrate your capabilities. A great resume will help you get the attention of HR managers. Showing someone actual work product, however, speaks much louder than words on a page. For this reason, a portfolio is particularly important, especially if you are new to the profession and don't have a ton of paid work experience.

This workshop is going to guide you through setting up a portfolio and giving you some ideas about what kind of project you might want to include. There are any number of ways you could host your portfolio, but we're going to use the free tools available on GitHub.com. They're free, and they integrate nicely with your hosted code repositories. So, if you're ready let's get started.

1. Set up a GitHub Account

The first thing to do is set up your GitHub Account, if you haven't done so already. TIP: your GitHub username will be part of the URL for your portfolio, so choose one that won't cause you to blush in front of a hiring manager. Simple so far, right?

2. Create a GitHub repository to Host your Portfolio

If you are new to Git, GitHub or to the topic of version control, you'll want to build your skills with this essential technology. There are lots of books and free videos on YouTube. Me personally, though, I prefer something a bit more structured. I am a regular user of Udemy.com, so I can personally recommend the site for inexpensive training courses, many around $10. Before you buy a course, though, make sure to preview it. I have browsed several of the Git courses, and this one seems pretty good -- taking about two hours to cover the key concepts.

If you're ready, head over to GitHub and create a new repository named ***username*.github.io**, where username is your GitHub username. Once you've created the repository, clone in to your computer and create an index.html page in the root folder of the repo. Code in an H1-tage in the body of your page and add a catchy title – your name, or Hello World always works. Commit your changes to the repository and push it up to GitHub. Now navigate to *username*.github.io, and you should see your page. Great!

Now you're ready to begin building your portfolio page, but if you want to know more about the tools GitHub offers for hosting websites, checkout this page.

3. Decide on a Site Structure and Theme

Alright, let's start building our site. To get started you need a couple things. First you need an organizational plan. Will you build a multi-page or a single-page site? Either way, you'll need to map out how your content will be organized. Second, you need a theme or style guide that defines the look and feel of the site. To increase your chances for success, I suggest starting simple then growing your site once you're up and running.

Most developers are not natural designers. Fortunately, the web is full of examples to help you create a professional looking site. To jumpstart the process, three one-page templates are included on this packet of materials, courtesy of the team at Start Bootstrap. Look them over. If you want to work with one as is, great. You're moving right along. If you want to customize it a bit, you can reassemble the features you like into a new template. this is a good way to see how others build interfaces. If you want to go completely custom, that's a great challenge too. Have at it!

4. Showcase your Work

Now it's time to show off your coding skills. You'll want to include a screenshot and title for each of your featured apps. If you have created space in your design, include a short description of the app, the technologies you used and the purpose/problem that gave rise to your app. What's the? You don't have a ton of great coding projects to showcase? No problem! That's what The JavaScript Workshop is all about -- having fun and building cool stuff that shows off your skills. If your looking for ideas, here are some types of apps/tools to think about including.:

- Native mobile app
- Component library
- HTML 5 element showcase
- Chrome extension
- 3rd-party API driven app (using 1 or more API's)
- Data scraper app
- Utility module
- NPM module
- Socket-driven multi-user app (game)
- Dashboard-visualization