How to make a Kick-Ass first Milestone

(Part 3.)

Common mistakes made in first milestones
things to check before you submit

Check your file structure is correct

- Your project needs an **assets** folder, inside that it should have a **css** folder with your css file inside. If you have images they belong in an **images** folder, which also goes in the **assets** folder
- All your html files belong in the top directory.
- If you have to change these, be aware that your file paths to your images may have to change too. See the video link below for understanding file paths.

Check your file names fit industry standard.

- Files names should be in lower case. For example rosie.jpg NOT Rosie.jpg
- The only exception to this is the README.md file which should be in capitals.
- File names should not have – in them if it can be avoided. For example contact.html, NOT contact-form.html

Deploy your website to GitHub Pages

- How to do that here
- Check that everything works the way it should on your github pages page.
- Often links to image and files do not work. This is because GitHub pages does not like absolute file paths, it needs relative paths.
  Check this video for the difference between the types of paths, and how to change yours to relative ones.
- Every time you push your code to github your github pages page will be updated. (sometimes it needs 5 mins to update though so be patient)
- You will use your github pages link to submit your project

Add a favicon to each of your html pages

- Here is how to do that

Make sure your css works on all browsers

- Use the Auto-prefixer to make sure your css has all the prefixes it needs to work on all modern browsers.
- Don't forget to add a link to the auto-prefixer in the Technologies Used section of your README.md

Check your contact form is validating input

- This means that your form should check that it has the correct input before allowing the user to submit the form.
- Bootstrap 4 documentation on forms here.

Check you do not have any overflow on your pages

- Overflow is when some of the content is wider than the screen size, which causes a horizontal scroll bar to appear at the bottom of the browser.
- Most of the time this is caused by bootstrap containers or rows adding padding by standard. The following css code usually fixes it:

```
row, container, container-fluid {
    padding: 0;
    margin: 0;
}
```

- If that does not fix it try using devtools to find what is causing the problem, and if you can't narrow it down ask in Slack for some help.

Use devtools device mode to check your sites responsiveness

- Make sure your site looks good on all major device sizes using devtools device mode
- If your site is not responsive, learn more about how to use the bootstrap grid and check out the demo website I made to explain the grid.

Make sure your README.md documents all the important information

- Document your UX / UI choices, explain why you laid things out the way you did.
- List your user stories, explain what they are looking for on your website.
- Document your testing, go through all your user stories and explain how you met the needs of your users.
- Document your deployment process properly. Full details on how to do that in this slack comment.
- Include links to any wireframes you made during the design process in your readme.
- Link to all the resources and technologies you used

Format your README.md using markdown

- Use this markdown cheat sheet to display headings, links, bullet points, images etc correctly.

Add enough comments to your code

- Imagine you are explaining your code to another developer who will be working on it in the future. Explain the WHY of your code, less of the WHAT it is for.
- Remove all commented out code in all html and css files.
- Comment (with links) in your code any time you used code by someone else. For example from stack overflow or w3schools.

Make your html sematic where possible

- Use this semantic html cheat sheet to change your s to more semantic html types where applicable.
- Be aware some semantic html will change the appearance of your site, so change one thing at a time and check it before moving on to the next one.

Validate your code using validators

- Use this validator for your html code
- Use this validator for your css code
- Make all corrections suggested so that your code passes

If you only do one thing:

Post your project in #peer-code-review for feedback

- Find the #peer-code-review channel in slack and paste in your github pages AND your github repository links there, ask for feedback on your project, be specific about what you would like checked.