How to make a Kick-Ass first Milestone

(Part 1.)

Things to know before you get started

Choosing a topic. Brief vs Guidelines.

The Guidelines are what you need to follow for your site in order to pass. The Brief is there for an idea, but it is not what you HAVE to develop.

Choosing a topic for your milestone that is something you are interested in and know a lot about will make your project much easier to work on.

- o You already will know where to find the media and resources you need for the content.
- o Your knowledge of the topic will make it much easier to choose its design, fonts, images etc.
- o You will want to work on it more, even when the code is not working the way you want!

What is the first milestone you marked on?

Each of these are marked out of 5 to make a total of 95. That means each of these is just over 5% of your total mark.

Project Purpose.	Appropriate use of CSS.
UX design.	Directory Structure and File Naming.
Suitability for purpose.	Version Control.
Navigation.	Testing implementation.
Ease of Use.	Testing write-up.
Information Architecture	Readme file.
Responsive Design	Comments.
Image Presentation.	Deployment implementation.
Colour scheme and typography.	Deployment write-up.
Appropriate use of HTML.	

Check out the project Checklist that Code Institute provides in the LMS before starting your project.

Importance of your README.md

- If you want great marks you need to dedicate time to your readme file.
- Your readme file is where you give the examiner most of the information about how you planned and executed your work. The reasoning behind it etc.
- Make sure your readme file is properly formatted using markdown.

Importance of mobile first approach to designing your page(s).

- Remember to use a responsive grid for your project. Bootstrap's built in media queries will help to make sure your website looks good on all screen sizes. Your site needs to be responsive to be graded well.
- Useful links for understanding the bootstrap grid in this Slack comment.

Tips to make your life easier.

Refresh the browser cache when you are checking your code.

- The MOST common problem new student's encounter is their code not showing up in the browser when they make changes.
- Use **CTRL + F5** on a PC or **CMD-SHIFT-R** on a Mac. This will force a refresh and clear cache when doing it and things should be up to date

If your CSS is still not working, I made a little troubleshooting page to help you track down what's wrong.

Habits that will get you marks

- **Plan your project** before you write even one line of code.
  - Use the planes section from the CI course to plan your project.
  - Use wireframes to build examples of how your page(s) will be laid out.
  - Have a reason behind the fonts, colours, design choices you make. Note them down in your readme.

- **Commit often!** Everything you add to your project should have its own commit.
  - You cannot commit too often in the eyes of examiners.
  - Make sure your commit messages are clear about what you changed.

- Sign up for one of @Eventyret_Mentor 's Git conference calls to learn more about the git flow. He tends to run one once every couple of weeks. Check the #i-dont-git-it channel on Slack.

- **Keep track of the resources you use for your project**.
    - In your readme you are expected to link to every place you got a piece of media from.
    - Get in the habit of adding these links to your readme file every time you pick up a new media for your website.

- **Add credit whenever you use code you found somewhere else.**
    - Add credit for where you got code from in comments with the code AND in your readme file.

- **Add meaningful comments in 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.

- **Do not be afraid to ask for help in Slack**.
    - This resource from a fellow student is excellent for learning how to ask the right questions to get answers faster.

Final note
- Make it easy for another developer to work on your project, because in doing this you are building great habits AND you are making it **easy for the examiner to see where you tick the boxes** for your project.
- Explain your code as you go along in the comments and in the readme.
- Break down why and how you did things.

Handy resources
- Free access to Balsamiq for wireframes for CI students.
- Bootstrap 4 documentation. Use the search bar on this page, it is very handy for learning bootstrap.
- Understanding the Bootstrap grid. Useful link here.

- Example README.md file from Code Institute
- Markup cheat sheet for formatting your readme file.