# Unit 02 CSS and Bootstrap Homework: Responsive Portfolio

Responsive design ensures that web applications render well on a variety of devices and window or screen sizes. As a developer, you will likely be asked to create a mobile-first application or add responsive design to an existing application.

## Directions

First, you will use the Bootstrap CSS Framework to create a mobile responsive portfolio. How do you deliver this? Here are some guidelines:

\* Create the following files files: `index.html`, `portfolio.html` and `contact.html`.

\* Using Bootstrap, develop your portfolio site with the following items:

\* A navbar

\* A responsive layout

\* Responsive images

\* The Bootstrap portfolio should minimize the use of media queries.

\* Screenshots are provided as a reference in the `Assets/Images` folder. Your app does not need to be \_exactly\_ like the images. Use Bootstrap to create a similar, responsive layout.

### Hints

\* Use Bootstrap's grid system (containers, rows, and columns).

\* On an `xs` screen, content should take up the entire screen. On `sm` and larger screens, you should have some margins on the left and right sides of the screen. Check out various sites on your mobile device vs. your computer to see examples of these differences.

\* Use an HTML validation service to ensure that each page has valid HTML.

### Minimum Requirements

\* Functional, deployed application

\* GitHub repository with README describing the project

\* Navbar must be consistent on each page.

\* Navbar on each page must contain links to Home/About, Contact, and Portfolio pages.

\* All links must work.

\* Must use semantic html.

\* Each page must have valid and correct HTML. (use a validation service)

\* Must contain your personalized information. (bio, name, images, links to social media, etc.)

\* Must properly utilize Bootstrap components and grid system.

### Bonus

\* Using Bootstrap, make a sticky footer and use sub-rows and sub-columns on your portfolio site (\*\*Hint:\*\* Check out the Bootstrap documentation).

## Commit Early and Often

One of the most important skills to master as a web developer is version control. Building the habit of committing via Git is important for two reasons:

\* Your commit history is a signal to employers that you are actively working on projects and learning new skills.

\* Your commit history allows you to revert your codebase in the event that you need to return to a previous state.

Follow these guidelines for committing:

\* Make single-purpose commits for related changes to ensure a clean, manageable history. If you are fixing two issues, make two commits.

\* Write descriptive, meaningful commit messages so that you and anyone else looking at your repository can easily understand its history.

\* Don't commit half-done work, for the sake of your collaborators (and your future self!).

\* Test your application before you commit to ensure functionality at every step in the development process.

We would like you to have well over 200 commits by graduation, so commit early and often!

## Submission on BCS

You are required to submit the following:

\* The URLs of the deployed applications

\* The URLs of the GitHub repositories

- - -

© 2019 Trilogy Education Services, a 2U, Inc. brand. All Rights Reserved.