# Sprint Challenge: User Interface and Git - Multi-Page Website

This challenge allows you to practice the concepts and techniques learned over the past week and apply them in a concrete project. This Sprint explored User Interface and Git. During this Sprint, you studied Semantic HTML, CSS Fundamentals, CSS Flexbox Module, and Git. In your challenge this week, you will demonstrate proficiency by creating a multi page website that has some missing HTML elements as well as CSS specificity problems that need to be solved. You will also create an additional web page that will be linked to from a navigation you will build.

## Instructions

\*\*Read these instructions carefully. Understand exactly what is expected \_before\_ starting this Sprint Challenge.\*\*

This is an individual assessment. All work must be your own. Your challenge score is a measure of your ability to work independently using the material covered through this sprint. You need to demonstrate proficiency in the concepts and objectives introduced and practiced in preceding days.

You are not allowed to collaborate during the Sprint Challenge. However, you are encouraged to follow the twenty-minute rule and seek support from your TL if you need direction. Your work reflects your proficiency in user interface and your command of the concepts and techniques in semantic HTML, CSS fundamentals, CSS flexbox module, and git.

You have three hours to complete this challenge. Plan your time accordingly.

## Commits

Commit your code regularly and meaningfully. This helps both you (in case you ever need to return to old code for any number of reasons and your project manager.

## Description

In this challenge, you build a missing header (navigation and image) on the home page, update some CSS styling on the home page, and create an additional page (About) which will link from the navigation you created.

In meeting the minimum viable product (MVP) specifications listed below, your web page should look like the solution screen shots of the home and about pages (found in the design-files folder):

[Click here for the home page example](https://tk-assets.lambdaschool.com/39a49225-8ac9-43da-aa90-514fd60ae99a\_sprint-challenge-ui-home-example.png)

[Click here for the about page example](https://tk-assets.lambdaschool.com/ede1bb1a-63ff-4801-8c02-3efa2f603190\_sprint-challenge-ui-about-example.png)

## Self-Study Questions

Demonstrate your understanding of this week's concepts by answering the following free-form questions.

Edit this document to include your answers after each question. Make sure to leave a blank line above and below your answer so it is clear and easy to read by your Team Lead

1. If you were to describe semantic HTML to the next cohort of students, what would you say?

I would reference it as a markup language, providing framework on-which to build a website, that you can later use to style through CSS.

2. Name two big differences between ```display: block;``` and ```display: inline;```.

Display: block takes up a whole line and adds white space above and below.

Display: inline only takes up the width it needs, without forcing new lines.

3. What are the 4 areas of the box model?

Margin, border, padding, and content.

4. While using flexbox, what axis does the following property work on: ```align-items: center```?

The “cross” or vertical axis.

5. Explain why git is valuable to a team of developers.

Primarily, the ability to branch and merge development of the project quickly and smoothly.

You are expected to be able to answer all these questions. Your responses contribute to your Sprint Challenge grade. Skipping this section \*will\* prevent you from passing this challenge.

## Project Set Up

- [ ] Create a forked copy of this project.

- [ ] Add your Team Lead as collaborator on Github.

- [ ] Clone your OWN version of the repository (Not Lambda's by mistake!).

- [ ] Create a new branch: git checkout -b `<firstName-lastName>`.

- [ ] Implement the project on your newly created `<firstName-lastName>` branch, committing changes regularly.

- [ ] Push commits: git push -u origin `<firstName-lastName>`.

Follow these steps for completing your project.

- [ ] Submit a Pull-Request to merge <firstName-lastName> Branch into master (student's Repo). \*\*Please don't merge your own pull request\*\*

- [ ] Add your Team Lead as a reviewer on the pull-request

- [ ] Your Team Lead will count the project as complete by merging the branch back into master.

## Minimum Viable Product

Your finished project must include all of the following requirements:

### Home Page

[Review the provided design file for the home page](design-files/home.png). Notice the navigation and header images are missing.

\* [ ] Build the HTML and CSS to create the missing navigation and header.

\* [ ] Link the `About` navigation item to the [about.html](about.html) page

You will also notice there are 10 boxes on the home page that need background colors. Use this list below to correctly style each box:

\* [ ] box1: `teal`

\* [ ] box2: `gold`

\* [ ] box3: `cadetblue`

\* [ ] box4: `coral`

\* [ ] box5: `crimson`

\* [ ] box6: `forestgreen`

\* [ ] box7: `darkorchid`

\* [ ] box8: `hotpink`

\* [ ] box9: `indigo`

\* [ ] box10: `dodgerblue`

### About Page

[Review the provided design file for the about page](design-files/about.png). You have been provided the HTML wrapper, footer, and page content for the about page. Create the rest of the missing HTML and CSS to match the design file.

\* [ ] Copy and paste your home page navigation and header into the about page

\* [ ] Update the header image with the about page image

\* [ ] Link the `Home` navigation item back to the `index.html` page.

\* [ ] Build the rest of the about page layout to match the design

In your solution, it is essential that you follow best practices and produce clean and professional results. Schedule time to review, refine, and assess your work and perform basic professional polishing including spell-checking and grammar-checking on your work. It is better to submit a challenge that meets MVP than one that attempts too much and does not.

Note: Please make sure you are using flexbox to layout your website. Floats, inline-block, tables, etc, should not be used for layout.

## Stretch Problems

After finishing your required elements, you can push your work further. These goals may or may not be things you have learned in this module but they build on the material you just studied. Time allowing, stretch your limits and see if you can deliver on the following optional goals:

\* [ ] refactor your HTML, make sure it's indented properly, clean, readable, you have written appropriate comments where necessary and that all attributes (required and encouraged) are filled out correctly.

\* [ ] Ensure your CSS is organized and readable, you've seperated your code by section and that you are using descriptive class names and adding classes in your HTML where styles repeat rather than rewrting the same styles over again

\* [ ] Use a flex item property of your choice when laying out a section of your website, ensure you can explain how and why you've used this property