Milestone Guide — Create a Travel Agency Home Page Using HTML and CSS

How to use this document

On this page, you'll find an example of how to break your development project down into steps. You'll find:

- recommendations for successfully completing each step.
- common problems and things to bear in mind.
- an estimate of project progress.

This breakdown is just a suggestion to help you plan your project. You don't have to follow the steps in this precise order.

Bear in mind that your progress through the different stages is only an estimate and will vary depending on your working pace.

General Recommendations

In this project, you're going to use **Figma** to access the desktop mockups for the site. Create an account and log in to Figma so that you can access the elements in the right-hand panel. Please note that you will be able to access the CSS code for each element within the mock-up in Figma. You can use this code as a starting point for your implementation if you like, but it's not complete and will need some changes. For example, the dimensions are all in pixels, but you'll need to adapt some of these to relative values ("em" units).

Step 1: Set up Your Development Environment

5% complete

Once you've completed this step, you should have:

- installed your chosen editor.

Recommendations:

1. Start by installing the code editor. We recommend you use Visual Studio Code, but you don't have to. Sublime Text is also a popular choice because it's very easy to use.
2. Import the Raleway font family from Google Fonts.
   ⇢ Which versions of the font match the mockup? Try typing a word to find out.
3. Here's how to use the Font Awesome library:
   a. Create a Font Awesome kit using the instructions found on the Font Awesome site and add it to the project.
   b. Check that this has worked correctly by adding a Font Awesome icon to your project.

Points to note:

- The code we've provided should give you a fully functional site to start with, even though the code is incomplete. However, if you change the code and run into some difficulties, here are some questions to consider:
  - The CSS style isn't working properly.
    ⇢ Have you linked the CSS file within the HTML code correctly?
  - The font isn't displaying correctly.
    ⇢ Did you import the Google Fonts fonts before using them?
  - The Font Awesome icons aren't displaying correctly.
    ⇢ Did you add a Font Awesome kit to the project beforehand?
- Are there errors, display problems or other things that aren't working in your code? This is something that happens frequently in code development. That's why it's important to know how to debug your code using relevant tools such as the Code Inspector in your browser's Developer Tools. For more details, take a good look at the Test Your Code With Chrome and Firefox DevTools chapter within the Set up Your Front-End Development Environment course we've provided in the resources.

Resources:

- Set up Your Front-End Development Environment course. You can skip the sections on Git.
- Font Awesome official documentation

Step 2: Break Down Your Mockup

10% complete

Before starting this step, you need to:

- set up your development environment.

Once you've completed this step, you should have:
- a mockup broken down into components representing the HTML code structure.

Recommendations:
- Read Part 3, Structure an entire page, of the Build Your First Web Pages With HTML and CSS course.
- Ask yourself the following questions:
  - Have you identified all the elements of the mockup (such as logo, search function, accommodation and things to do cards, etc.)?
  - How are the different elements nested in a hierarchy?
  - Which HTML tag would you use as the container for each element?
- Work out whether to position elements horizontally or vertically. Each block must contain either horizontal elements only (e.g., filters on the navigation bar) or vertical elements only (e.g., the cards from the "Most Popular" section).

Once you've broken down your mockup, discuss it with your mentor. This step will ensure that you're asking the right questions and checking that you haven't forgotten anything.

Points to note:
Don't spend too much time trying to get the *perfect* breakdown. The aim is to get a quick overview of the mockup's main features.

Resources:
- Break Down and Integrate a Mockup course
- Design a Mock-up for Web Development With Figma course

Step 3: Add the Project Header

20% complete

Before starting this step, you need to:
- create a breakdown of the mockup.

Once you've completed this step, you should have:
- the code for the page header.

Recommendations:
- You can use Flexbox to position the Booki logo and the Accommodation/ Things to do navigation links.

Points to note:

- Don’t forget about the blue border that appears when you hover over the navigation elements with the mouse.
- Identify when you need to use a margin property rather than padding.
- It’s preferable to use pixels rather than percentages for margin and padding values.
- HTML tags have CSS properties by default that are set by the browser (for example, an “h1” title is displayed in bold by default). You might need to override these browser defaults for these elements.
- Note that the “body” tag has margins built in by the browser by default. You can override this by setting your own margin properties for the **body** tag in your CSS.

Resources:

- Implement one-dimensional layouts with Flexbox section of the Create Web Page Layouts With CSS course
- CSS Trick: A Complete Guide to Flexbox — particularly the “Flexbox Properties” section
- Flexbox guide on the Mozilla Developer Network
- Flexbox Froggy game, where you can practice writing CSS code

Step 4: Add the Search Form

30% complete

Before starting this step, you need to:

- add the page header.

Once you’ve completed this step, you should have:

- a search form included in the HTML page.

Recommendations:

1. Identify how this form is constructed. It’s made up of three sections, as we can see from the mockup.
2. Then, use CSS to display or hide the word or the icon based on what type of screen is being used.

Points to note:

- Make sure you don’t apply a border to the whole search field, so that it matches the mockup.

Resources:

- HTML Forms tutorial by W3Schools

Step 5: Add the Filters Section

40% complete

Before starting this step, you need to:
- add the search form to the page.

Once you've completed this step, you should have:
- a completed Filters section.

Recommendations:
1. If you've broken down your mockup properly, you should be able to use Flexbox to add this section successfully.
2. Start by adding the filters without worrying about the hover behavior at this stage.
3. Once your filters are set up, you can change the background color when hovering over them.

Points to note:
- Identify when you need to use a margin property rather than padding.
- It's advisable to use pixels rather than percentages for margin and padding values. Using percentages on margins and padding won't give you the desired result, because sizes can vary too much from one screen format to another.

Resources:
- Part 1 of the course Create Web Page Layouts With CSS

Step 6: Create the "Paris Accommodation" Card

50% complete

Before starting this step, you need to:
- add filters to the page.

Once you've completed this step, you should have:
- your first accommodation card. This will come in handy in Step 8.

Recommendations:
1. The "Accommodation" cards are similar to the "Most Popular" cards already provided, so use them for inspiration. Note, however, that they are different in

terms of how the elements are aligned (horizontally for the "Most Popular" cards and vertically for the "Accommodation" cards).
2. If there is no CSS property applied to an image, it will display in its original size. As with many elements, it's a good idea to give an image's width as a percentage. You'll need to define the height in pixels.
3. Once you've provided the image dimensions, it should be distorted. The CSS property "object-fit" will correct this.

Points to note:
- The images must be added via HTML.
- The following elements should be covered by the CSS we've provided, but make sure they exist:
  - The shadow on the card
  - The alt attributes
  - The border-radius properties on the images

Resources:
- How to Create a Card with CSS article with some code snippets for creating cards
- MDN article about the "object-fit" property

Step 7: Manage the Card Display Within the "Paris Accommodation" Container

60% complete

Before starting this step, you need to:
- create your first card in "Paris Accommodation".

Once you've completed this step, you should have:
- most of the page layout complete. You just need to create the "Things to do in Paris" section and the page footer.

Recommendations:
1. For this step, the code we've provided will manage the display of the majority of page elements using Flexbox and with widths expressed in percentages. Even though you don't need to code these parts, make sure you understand the code.
2. Duplicate the card you created before so you'll end up with six, as shown in the mockup.
3. Define the container layout. It should be helpful to use Flexbox and define the card width as a percentage (using existing cards for inspiration).

4. Replace the contents of the duplicated cards with the content shown on the mockup.

Points to note:
- Don't forget the title, the icon, and the "Show more" link.

Resources:
- Implement one-dimensional layouts with Flexbox section of the Create Web Page Layouts With CSS course
- CSS Trick: A Complete Guide to Flexbox — particularly the "Flexbox Properties" section
- Flexbox guide on the Mozilla Developer Network

Step 8: Add the "Things to do in Paris" Container

70% complete

Before starting this step, you need to:
- add the "Paris Accommodation" section and "Most Popular" sections to the page.

Once you've completed this step, you should have:
- almost completed the project. You'll just need to create the footer.

Recommendations:
1. Refer to the previous recommendations we've provided when building the different cards.
2. In this section, the structure is a little different, because it's presented in four columns.
3. The height of each activity is the same. So, you'll need to create an initial activity (as you did in the previous steps) and add the other three based on the first.

Points to note:
- The images must be added to the HTML and not the CSS.
- Don't forget the alt attributes. I know we've mentioned this a few times, but it's really important to remember!

Resources:
The resources provided in the previous step will be helpful here, too.

Step 9: Implement the Footer

80% complete

Before starting this step, you need to:

- add the “Things to do in Paris” section to the page.

Once you’ve completed this step, you should have:

- completed the coding for this project.

Now, you’ll need to check that the code is valid according to the W3C validators.

Recommendations:

1. Once again, you can use Flexbox for the page layout.
2. You really need to follow your mockup breakdown and identify the different elements and blocks.
3. The footer is made up of three columns of the same width.

Points to note:

- If you use “ul” for links, a default “padding-left” property will be applied. Consider removing this, because it will offset the content from what’s displayed above.

Step 10: Manage Size Limits

90% complete

Before starting this step, you need to:

- have a fully functional desktop version of the site that behaves correctly on a screen with a 1024 pixel screen width.

Once you’ve completed this step, you should have:

- a project that is fully compatible with the specified screen sizes.

Recommendations:

1. Remember to define a maximum width for the page content of 1440 px to cater correctly for screens with a higher resolution.
2. Remember also to define a minimum width of 1024 px—you do not need to ensure that the layout of the page remains as specified for screen widths less than 1024px.

Points to note:

- You may experience minor inconsistencies at certain resolutions. So remember to check the behavior of your page at 1024 px and 1440 px, as well as the resolutions between these two values.
- Check the behavior below 1024 px to see whether a minimum value has been correctly set.
- Also check the behavior above 1440 px to see if a minimum value has been set. Make sure that the website content is centered on screens larger than 1440 px.

Step 11: Check Your Code Quality

100% complete

Before starting this step, you need to:

- finish adding the mockups for all screen formats.

Once you've completed this step, you should have:

- completed the project!

You'll just need to prepare for the assessment session.

Recommendations:

1. Focus on any errors encountered. You can look at warnings, but you don't have to resolve them.
2. Pay attention to code naming conventions (e.g., for CSS classes).
3. Use kebab case, e.g., "main-wrapper". This is the most commonly used CSS convention.
4. It's generally advisable to use pixels for margins and paddings and percentages for widths to help manage different screen resolutions. In some cases, it might be relevant to use pixels for widths (e.g., to manage icon size).

Resources:

- Code validation tools:
  - W3C HTML Validator
  - W3C CSS Validator

Project complete!