Key Steps Guide - Create a Travel Agency Homepage with HTML & CSS

How to use this document ?

On this page, you will find an example of dividing into stages to make your project. You will find :

> • recommendations for success at each stage; • recurring problems and points of vigilance to keep in mind; • an estimate of your progress throughout the project.

This breakdown is simply a suggestion to organize yourself. You are under no obligation to complete the steps in order.

Keep in mind that your progress on the stages is only an estimate and will vary depending on your rate of progress.

General recommendations

In this project, you'll use **Figma** to access the site's desktop, tablet, and mobile mockups. Create an account and sign in to Figma to access item details in the menu on the right. Attention however: you will be able to access the CSS code of each element. You can start from this code for your integration if you wish, but it will be incomplete and will require modifications. In particular, the dimensions are all in pixels, but you will need to adapt some of them in relative values (notably "em" units).

Step 1: Set up your development environment **5% progress**

Once this step is complete, I must have: • your
   code editor (of your choice) installed.

Recommendations:

1. Start by installing the text editor. Visual Studio Code is recommended, but not
   required. Sublime Text is also appreciated for its ease of use.

2. Import Raleway font variants from Google Fonts.
   ÿ Which variants correspond to the model? For the knowledge,
      test them on one word.

3. To be able to use the Font Awesome library: a. Create
   yourself a kit on Font Awesome by following this link And
      add it to the project; b.
      Check that it works correctly by adding a Font Awesome icon to your project.

Vigilant point :

• The code provided should allow you to get the site working right from the start, even
   if it is incomplete. However, if you modify the code and encounter difficulties, ask
   yourself the following questions: • CSS styling is not working correctly.
      ÿ Did you link the CSS file correctly in the HTML code?

   _____

      • The font does not display correctly.
         ÿ Have you imported the fonts from Google Fonts before
            to use ?
      • Font Awesome icons are not displayed correctly.
         ÿ Did you add a Font Awesome kit to the upstream project? • Are there
errors, display issues, or something else not working in your code? This is something
   that often happens in development. This is why it is important to know how to debug
   your code using suitable tools such as the code inspector. To do this, pay attention
   to the chapter "Test your code with Chrome and Firefox devtools" of the course "Set
   up your front-end environment" given in the resources.

Resources: •
   the course " Set up your front-end environment by skipping the parts on Git; • official
      documentation
   to use Font Awesome.

Step 2: Slice your mockup **10% progress**

Before starting this step, I must have:

• set up the development environment.

Once this step is complete, I should have:

• a cut model, representing the structure of the HTML code.

Recommendations:

• Follow the webinar " Cut out a model to prepare for integration » to make the cutout
of the Booki model. • Ask yourself the following
questions:

• Were you able to identify everything that is visible on the model (elements
such as the logo, the search function, the accommodation and activity
cards, etc.)? • How are the different
elements grouped? • For each element, what HTML tag is
associated with it?
associated?

• Define when to position elements horizontally or vertically. Indeed, each block only
contains elements positioned either horizontally (for example, the filters of the
navigation bar), or vertically (for example, the cards in the "most popular" section).

Once you've cut your mockup, discuss it with your mentor. This step will allow you to ask
yourself the right questions and to check that you have not forgotten anything.

Points of vigilance:

Do not waste too much time making a **perfect cut.** The main thing is to have a quick view
of the main challenges of the model.

Resources: •

the webinar " **Cut out a mock-up to prepare for integration** ". • the course "
Create a web model with Figma ".

Step 3: Integrate the project header

20% progress

Before starting this step, I must have: • cut out the
model.

Once this step is complete, I must have: • the
page header code.

Recommendations:

> • To achieve the positioning between the Booki logo and the Accommodation/Activities
> parts, you can use Flexbox (or Grid).

Points to watch out

for: • Be careful not to forget the blue border displayed on hover. • Distinguish
when to use a margin property instead
that padding.
• It is preferable to use pixels rather than percentages for
values of margins and paddings.
• HTML tags have default CSS properties (for example, an "h1" title is displayed in bold
by default).
• You should know that by default, the "body" tag includes margins.

Resources: •

The chapter « Make your layout with Flexbox of the course "Learn how to create your
website with HTML5 and CSS3"; • The little game Flexbox
Froggy to practice writing CSS code.

Step 4: Add the **30% progress** search form

Before starting this step, I must have: • integrated
the header of the page.

Once this step is complete, I should have:
• A search form integrated into the HTML page.

Recommendations:
1. Clearly identify the construction of this form, which consists of
3 parts, as seen on the model.
2. Then, with CSS, it will be necessary to show or hide the word or the icon in
depending on the screen used.

Vigilant point :
• Be careful not to apply a border to the entire field of
research, so that the visual corresponds to the model.

Resources: •
The Create Forms chapter of the course "Learn how to create your site
website with HTML5 and CSS3".

Step 5: Adding the Filters part
40% progress

Before starting this step, I must have:
• integrated page search form.

Once this step is complete, I should have: • the
Filter part fully realized.

Recommendations:
1. With a good slicing of the model, you should be able to integrate
this part correctly with Flexbox.
2. Start by integrating the filters without worrying about the change of appearance on
hover.
3. Once the filters are in place, you can implement the background color change on
hover.

Vigilant point :

- Distinguish when to use a margin property instead
  that padding.
- It is preferable to use pixels rather than percentages for the values of margins and
  paddings. Indeed, the use of percentages on margins/paddings would not
  correspond to the expected result, because the variations in size would risk being
  too great from one screen format to another.

Resources: •

The chapter " Discover the box model of the course "Learn to
create your website with HTML5 and CSS3".

Step 6: Make the "card" present in " Accommodation in Marseille
» 50% increase

Before starting this step, I must have: • integrated
the filters of the page.

Once this step is complete, I should have: • a first
accommodation card. It will be useful in step 8.

Recommendations:

1. The "accommodation" maps are similar to the "most popular" maps already provided,
   so take inspiration from them. Be careful though, they are different, especially in
   the direction of alignment of the elements (horizontally for "the most popular" and
   vertically for the "accommodation" cards.

2. If no CSS properties are applied to an image, it will be displayed in its original size.
   As with many elements, giving the image a "%" width is a good idea. It will be
   necessary to define a height in pixels.

3. When sized, the image should be distorted. The CSS Property
   `object-fit` will correct this.

Points of vigilance: •

Images must be integrated via HTML. • Normally the
following are covered by the provided CSS,
but make sure they are present: • the shadow
cast on the map;

• alt attributes; • the
border-radius on the images.

Resources: •
W3C How to – Cards article with some code snippets for
make a card; • The
MDN article on the "object-fit" property.

Step 7: Manage the display of container "cards" "Accommodation in Marseille" **60% increase**

Before starting this step, I must have: • created a
first "card" present in "Hébergements à Marseille
".

Once this step is complete, I should have:
• Much of the site layout. Only the part "
Activities in Marseille" and the footer to create.

Recommendations:
1. For this step, a good part of the display of the elements of the page is already
managed by the code provided, in particular thanks to Flexbox and the use of
widths in percentages. Even if you don't need to code these parts, make sure you
understand the associated code.
2. Duplicate the "card" made previously in order to have 6
like on the model.
3. Create the layout of this container. The use of flexbox as well as the definition of the
width of the "cards" in percentage should help you (for this, take inspiration from
what has already been done).
4. Replace the content of the different duplicate "cards" with the content
of the "cards" of the model.

Vigilant point :
• Don't forget the title, icon and "Show more" link.

Resources: •
The chapter « Make your layout with Flexbox of the course "Learn how to create your
website with HTML5 and CSS3".

Step 8: Integrate the "Activities in Marseille" container **70% progress**

Before starting this step, I must have: • integrated
the two sections "Accommodation in Marseille" and "The most
Popular" from the page.

Once this step is complete, I should have: • the
project almost finished. There will only be the footer left to do.

Recommendations:

1. Rely on the previous recommendations given for the realization of the different cards.

2. In this section, the structure is a little different since it is a
presentation in 4 columns.
3. The height of each of the activities is identical. It will then be necessary to set up (as
in the previous stages) a first activity, and to integrate the 3 others from the first.

Vigilant point :

• Images should be embedded via HTML and not CSS. • Don't forget
the alt attributes. This point has been repeatedly
mentioned, but it is important to remember it!

Resources:

The resources from the previous step will come in handy here too.

Step 9: Implement the **80% progress** footer

Before starting this step, I must have:

• Integrated the "Activities in Marseille" section of the page.

Once this step is complete, I should have: •
Completed the project code.
It will now be necessary to check if the code complies with the W3C validators.

Recommendations: 1.

Again, you can use Flexbox for layout.

2. You will have to rely on the breakdown of your model, and identify the different elements as well as the different blocks.
3. The footer is presented in 3 columns of identical sizes.

Points of vigilance: •
If you use `ul` to make the links, you have by default a padding-left that applies. Remember to remove it, as it will cause a shift from what is directly above.

Step 10: Manage size limits
90% progress

Before starting this step, I must have: • the
desktop version of the site, fully functional, behaving correctly from 1024 px resolution to 1440 px resolution.

Once this step is complete, I should have:
• a project fully compatible with all desktop screen sizes.

Recommendations:
1. Remember to set a maximum width of 1440 px to manage screens with high resolution correctly.
2. Also consider setting a minimum width of 1024px.

Vigilant point :
• You may have small inconsistencies on some resolutions.
So remember to check the behavior of your page at 1024px, at 1440px, as well as the resolutions between these two values. • Check the behavior below
1024px to observe the correct
establishment of a minimum value
• Also check the behavior above 1440px to observe the correct implementation of a maximum value. Make sure that the content of the website is centered on screens greater than 1440px.

Step 11: Check the quality of your **100% progress** code

Before starting this step, I must have: • finished integrating the models, on all screen formats.

Once this step is complete, I should have: •
Completed the project!
All you have to do is prepare for the defense.

Recommendations:
1. Focus your efforts on reported errors. You can look at the warnings, but you don't have to deal with them.
2. Pay attention to the naming of code (of CSS classes, for example).
    You can use English or French, but avoid mixing the two languages.

3. Use the kebab case, for example `.main-wrapper` . This is THE most widespread CSS convention.
4. Generally prefer the use of pixels for margins/paddings, and percentages for widths, in order to allow good management of different resolutions. In some cases, the use of pixels for the widths can be something relevant (management of the size of an icon, for example).

Resources: •
Code validation tools:
    • W3C HTML validator ; • W3C CSS validator.

Project finished!