

# Second Project: Advanced HTML and CSS

Sprint 2



# Introduction

The first thing you'll notice about this project is that it's similar to Project 1. Although you'll be building Project 2 from scratch, you're free to reuse blocks from your previous project and adapt the HTML and CSS accordingly. Finding similarities and reusing your code will help a lot! When planning out your project, consider how the structure and styles differ. This brief will give you all the dimensions, which should simplify things.

You'll notice that this brief is less detailed than the last one. This is intentional — after all, your skills have become more advanced.

As a budding web developer, you may be interested in learning more about working with web designers and why certain designs look the way they do. Here's a comment from our designer:

*This is the website of the fictional coffee shop in the Practicum Library. The design follows the style of the Library website; however, since the coffee shop itself is less formal, this website is more colorful and uses more graphics. The circles in the background symbolize coffee tables and cups. I suggest animating one or two of these circles. The last circle in the "About" section should pulsate, representing the heart of the coffee shop.*

— Danila, Practicum designer

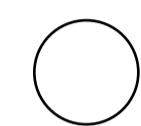
## Table of contents

Assignment overview .....	03
Stage 1. Building the basic structure .....	06
Stage 2. Step by step implementation .....	09
1. Header .....	10
2. Section: Menu .....	13
3. Section: Recipes .....	16
4. Section: Reservation .....	19
5. Section: About the coffee shop .....	23
6. Footer .....	27
Stage 3. Self-review .....	30

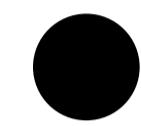
# Assignment overview

Here is how your webpage should look by the end of the project.

## Main Colors:



#ffffff — main background, contrast font color



#000000 — main font color



#2F80ED — accent, button



#838383 — input placeholders

The screenshot shows the homepage of the Practicum Coffee Shop website. At the top left is the logo 'Practicum Coffee Shop' with a coffee cup icon. At the top right are navigation links: Menu, Recipes, Book a table, and Contacts. Below the header is a large image of a study zone in a library, featuring a black round table, blue chairs, and bookshelves. To the left of the image is the text: 'Specialty coffee in the Practicum Library'. Below this is a welcome message: 'Welcome to our study zone! We've got the coffee to energize you and the cozy vibes to help you get creative.' At the bottom left, there is information about hours: 'Hours: Monday – Friday (10:00 – 19:00) Saturday – Sunday (11:00 – 18:00)'. At the bottom right, the address is given as '200 Success Avenue, The Town of Practicum'. The footer is dark blue and contains a 'Menu' section with a description: 'We brew coffee from locally roasted, Fairtrade certified beans and use biodegradable cups. All teas from our collection are \$2/cup.' It also includes 'Filter' and 'Espresso' sections with a single item each: 'Regular (12 oz) \$1' and 'Macchiato \$2.25'.

Large (16 oz)	\$1.25	Cortado	\$2.5
XL (20 oz)	\$1.5	Mocha	\$3
Party-size (24 oz, decaf)	\$1.75	Sunrise in the Bay Area (decaf)	\$10
Iced Coffee (16 oz)	\$1.25		

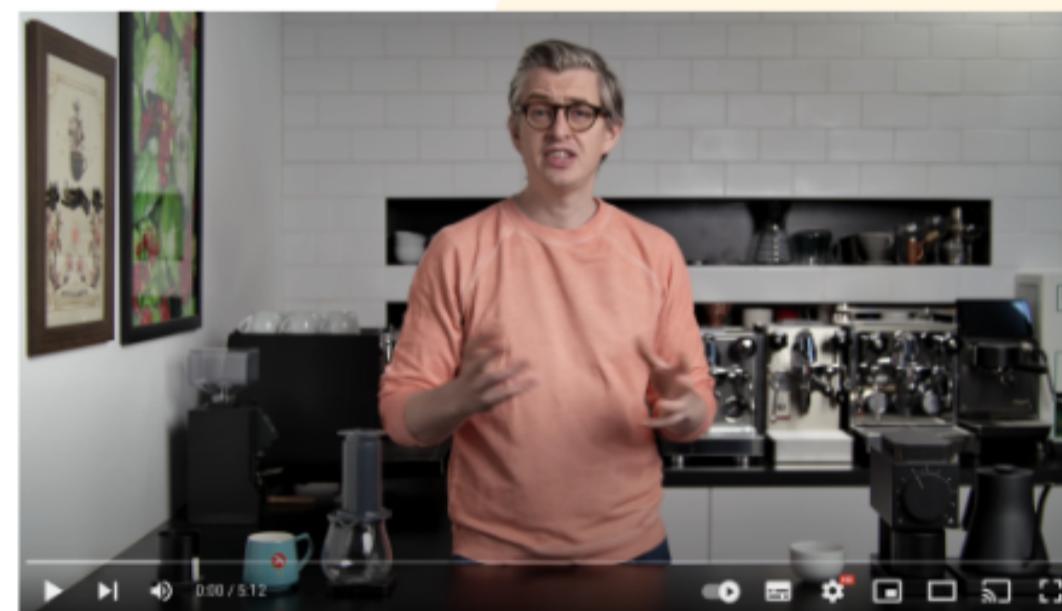
Assignment overview

#### Baked Goods

Almond Croissant	\$2
Banana Bread	\$1.75
Key Lime Pie	\$3
Blue Velvet Cake	\$3.75
Web Cookie	\$2
Fresh Bug'ette	\$0.25

## Recipes

Check out some recipes we've collected for your home-brewing convenience:



Aeropress recipe



French press recipe

~5 min

~15 min

## Book a table

Name \*

Name Surname

Number of guests \*

1-8

Date & time \*

dd.mm.yyyy

Your email \*

email@email.com

Book a table

I agree with the terms of use

Assignment overview

## About the coffee shop

Our coffee shop is located in the Practicum Library's hall. It features 6 tables and free WiFi.

Our main mission is keeping guests cozy and energized.

Besides that, we try to reduce our environmental impact — so we don't sell single-use cups. But you can buy one of our reusable cups!



© 2022 Your name

### Social media

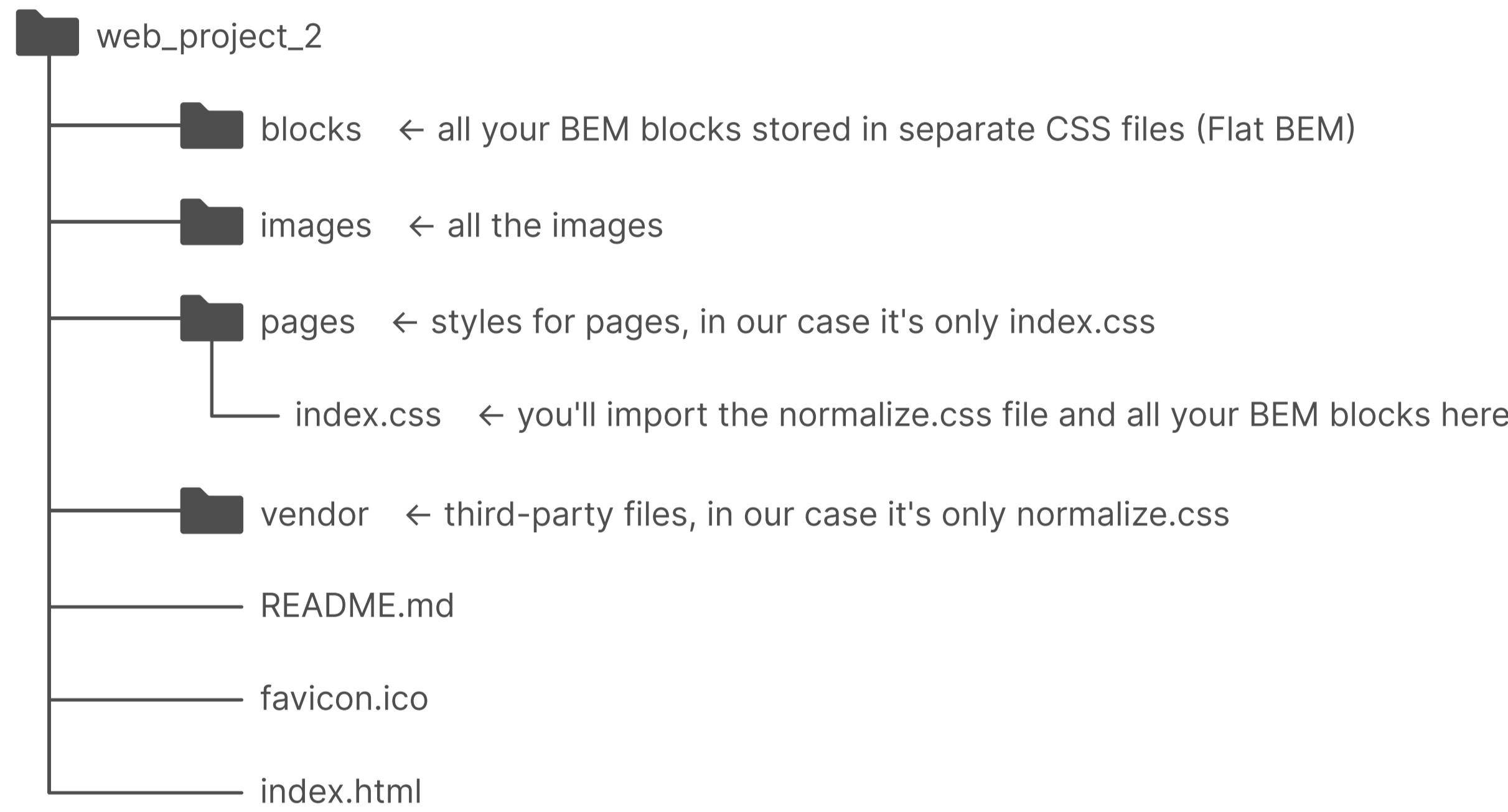
Facebook

Instagram

# Stage 1. Building the basic structure

## 1. Creating the file structure

First off, let's create the main folders and files for the project. The structure should look like this:



Make sure you've downloaded the necessary files from the Project 2 page. Add the files to the corresponding folders based on the structure above. Note that, according to [common practice](#), the favicon file should be placed in the root folder.

## 2. Creating the HTML page structure

Open [index.html](#). Outline the basic document structure: the `<!DOCTYPE>` declaration and the `<html>` tags with `<head>` and `<body>` tags nested inside. Specify the webpage title: you can simply use "Practicum Coffee Shop" or try to be creative. Next, set the page language as English.

### ! Pro tip:

We can use [Emmet](#) abbreviations to quickly produce some common HTML and CSS code.

Take the `!` command, for example. Entering `!` in the code editor will add boilerplate HTML code, complete with meta tags. Boilerplate is code that is usually the same across pages/projects, so it's useful to be able to quickly add it. We can then add code to our page using Emmet abbreviations, usually just by hitting the Tab key. To make sure it works, your cursor needs to be active and placed right after the abbreviation.

## 3. Don't lose your `<head>`

- To kick things off, let's provide some useful info for the browser and search engines that we learned about in Sprint 1 (see Advanced HTML and CSS Capabilities). Use `meta` tags to set the following:
  - The character encoding to UTF-8.
  - The `viewport` is defined so that 1) the `width` of the page matches the user's `device-width`, and 2) `initial-scale` is equal to `1`.
  - For SEO (Search engine optimization): `description` (1-2 sentences), `author` with your name, and `keywords`.
- Now, go ahead and connect `index.css` using the `<link>` element.
- Finally, in the files we've provided, you'll find the favicon prepared by our designer. Choose the one you like and add it to the project root folder. Connect the favicon using `<link>`.

## 4. Importing `normalize.css`

In the previous project, we connected the `normalize.css` file using the `<link>` tag. However, there's also another way. You just learned how to connect the CSS styles of BEM blocks using the `@import` at-rule, so now let's apply this knowledge to `normalize.css` as well. Import this file to `index.css`. Note that all other CSS files must be imported after `normalize.css`.

## 5. Fonts

1. For this project, you'll use Google Fonts.

In your next project, we'll ask you to figure out the fonts by yourself. This time, however, we'll give you a list. Using the instructions right below the list, you'll need to connect two fonts with the following styles:

Inter

- Regular 400
- Medium 500
- Bold 700

Noto Serif

- Regular 400

For each font, go to [fonts.google.com](https://fonts.google.com) and search for the font family.

Scroll down and select the specific style(s) you want. You'll see a list of the selected styles to the right, and the site will generate the code automatically, you just need to add it in your `<head>`. Make sure to connect it via `<link>`.

2. When you assign font styles to webpage elements, make sure to specify fallback fonts:

- For Inter, use Arial as a fallback and use the generic sans-serif family in case the user doesn't have either of the other fonts.
- For Noto Serif, use Times New Roman as a fallback and use the generic serif family in case the user doesn't have either of the other fonts.

## 6. Outlining the main blocks of the webpage

1. Just like you did in the first project, we'll start by creating a wrapper for the entire page. Create a `<div>` element with the `page` class, then style it so that:

- The block has a minimum width of `1100px` and a maximum width of `1600px`.
- The block is centered for screens over `1600px` wide.

Next, let's specify the font settings for the `page` block:

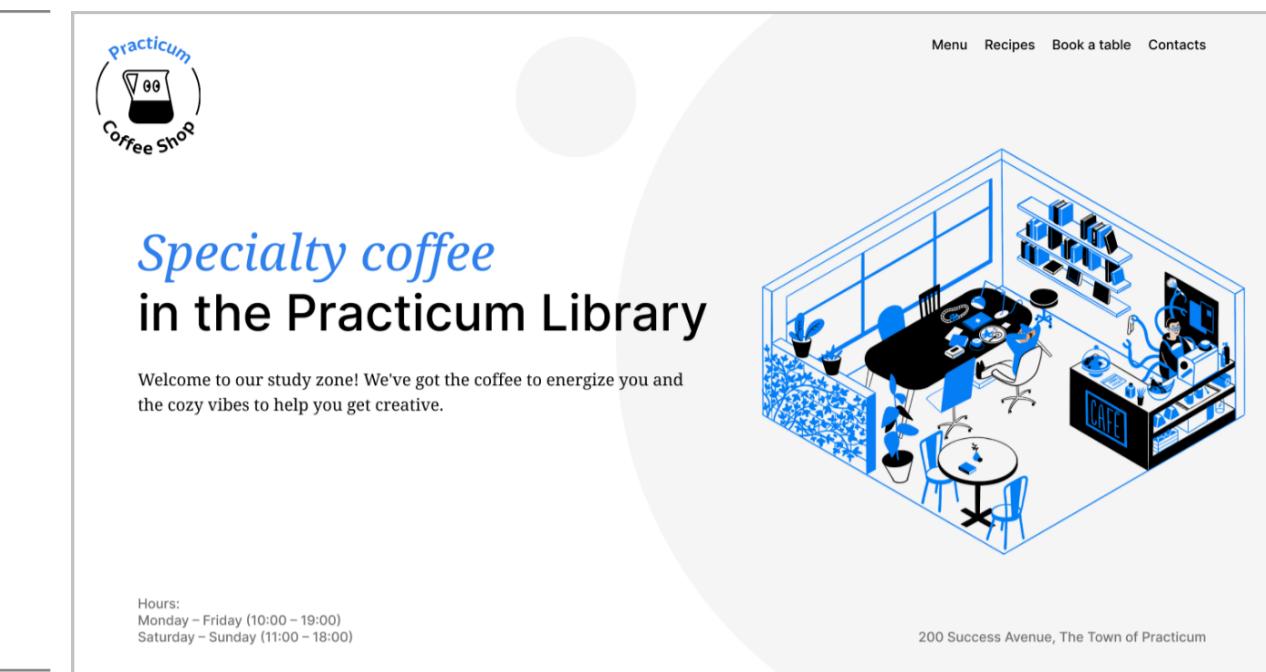
- Set the font family to "Inter" (remember the fallback fonts).
- Set the font size to `16px`.
- Set the line height to `20px`.

2. Now, take a look at Pic. 2. Notice how the header, main, and footer sections form the upper-level structure of the webpage. Create these semantic elements according to Pic. 2 inside the `page` block. Assign classes to them according to Pic. 2.



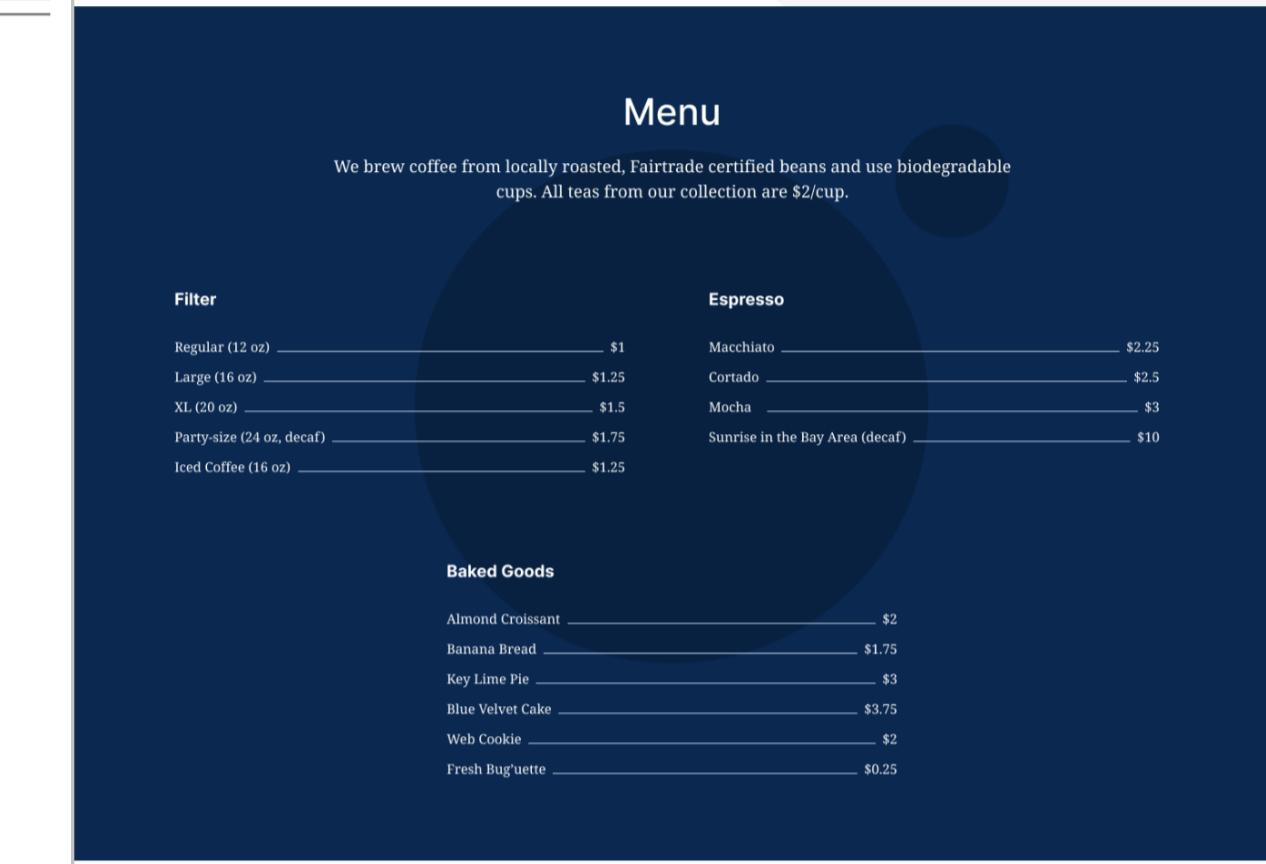
And with that, we're done with all the prep work. Now, you'll finish your webpage section by section. This is the regular process for marking up a webpage, though there is no strict rule.

Site header:  
`header` block

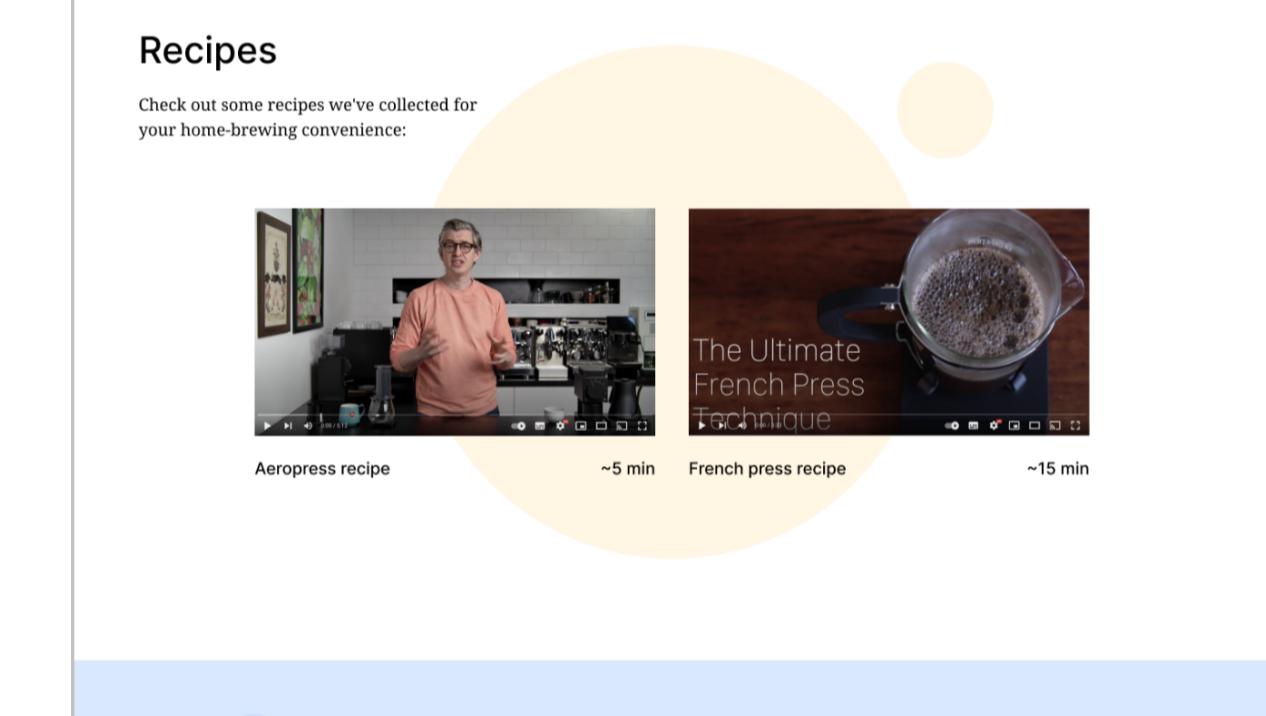


Main page content

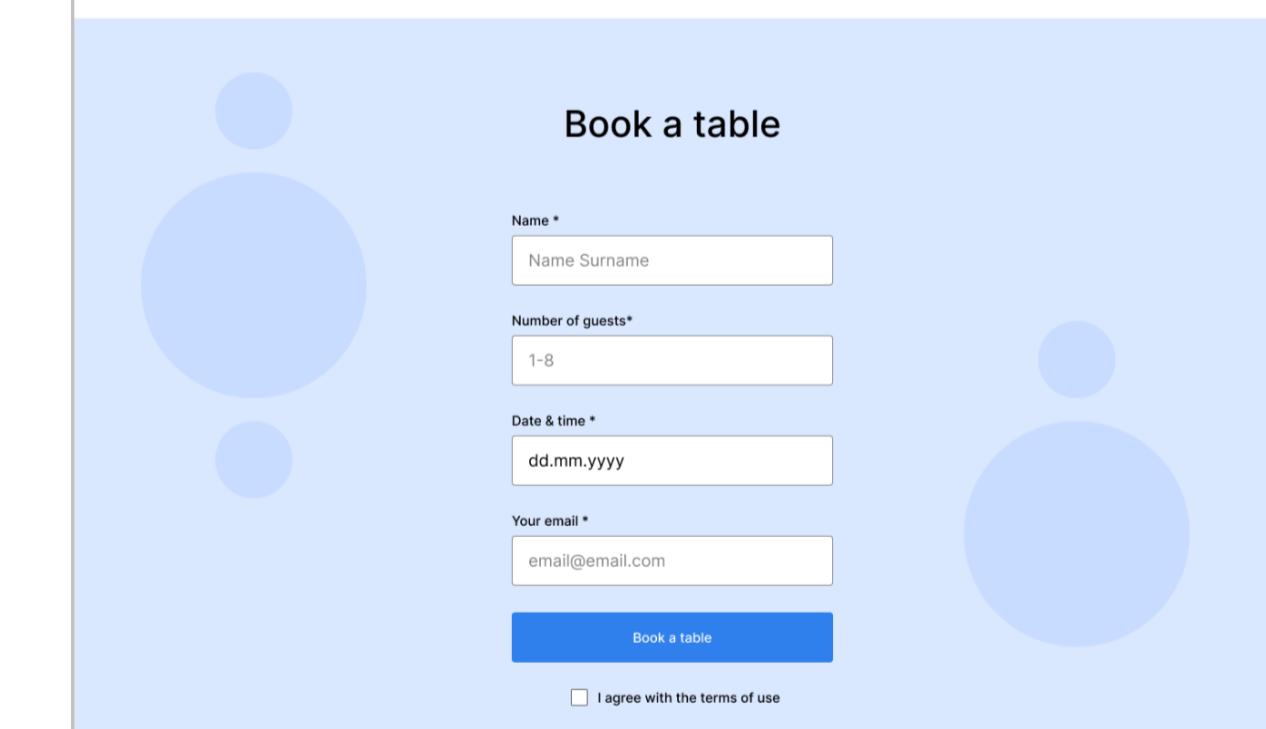
Site footer:  
`footer` block



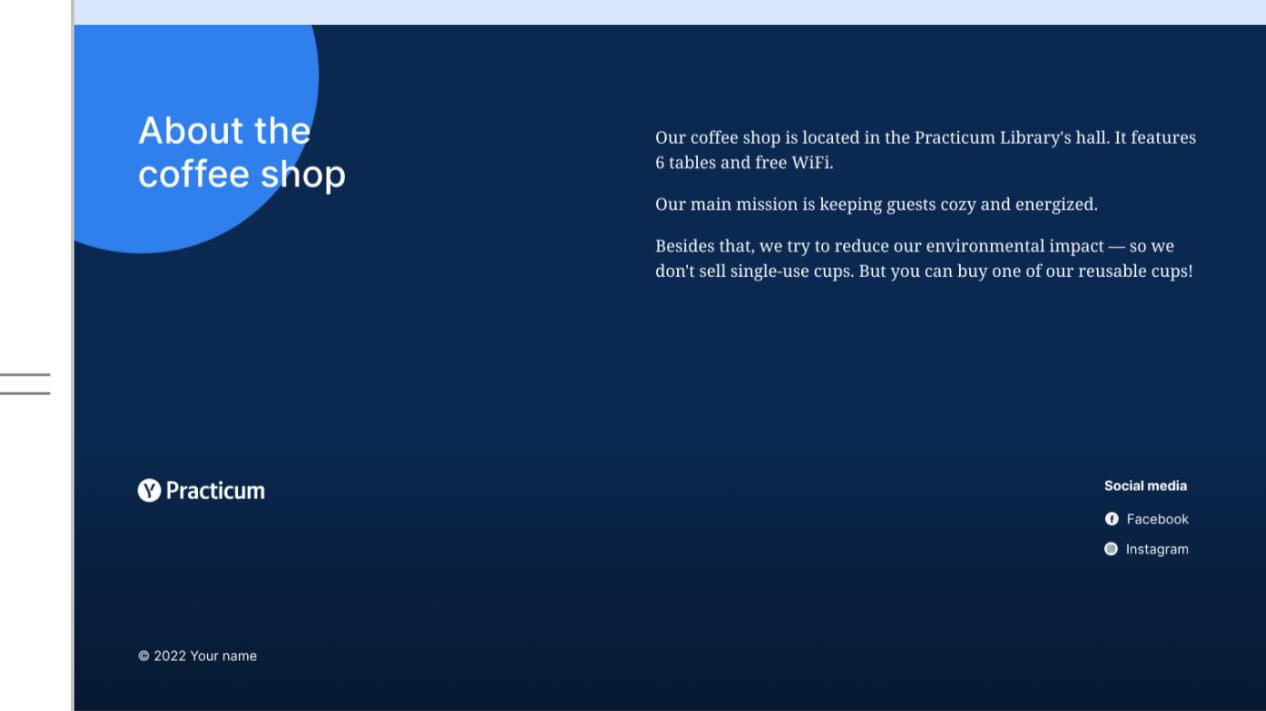
Section:  
`menu` block



Section:  
`recipes` block



Section:  
`reservation` block



Section:  
`about` block

Pic. 2

# Stage 2. Step-by-step implementation

Take the page content from the [texts.md](#) file that was provided in the archive. When you don't need it anymore, delete it from your repository — it's not needed for the finished project.

You'll work through each of the sections on the page, one by one.

**!** **Pro tip:**

Many developers prefer to create the entire HTML structure first and then define styles for the entire page. The alternative is finishing both the HTML and the CSS for each section as you work through them. Find out which way works best for you and don't be afraid to experiment!

For each individual section, we'll adhere to the following structure: Required elements and class names → sizes and spacings → typography.

The "About the coffee shop" section will also include instructions on animation.

All the elements and class names shown in the brief are required, but you are free to create additional elements, such as wrappers, if you think you need them. Just ensure that you adhere to the BEM naming principles.

# 1. Header



## Required elements and class names

**Background:** Set the background color to `#FFFFFF`. For the background image, provide the URL to the [background-header.svg](#) file. The background doesn't repeat and its position is centered. Set the size value to `cover`.

1. block: **logo**. Find the corresponding image in the [images](#) folder.

2-5. block: **nav**, element: **link**. The instructions for writing class names will follow this format throughout the brief. In this case, it means that your class name should be written as follows: `nav__link`.

Create a hover state for links so that they smoothly transition to `#2F80ED`. You can specify any values you'd like for transition properties, such as duration or timing function.

Remember to use the **nav** semantic element. You can build the navigation bar in a similar way to what you did in Project 1.

6. block: **header**, element: **title**. Since this is the title of our page, use an **<h1>** heading. Also, keep the text on separate lines, as shown in the design. It may be a small detail, but small details can have a big impact on the look of the site.

7. block: **header**, element: **span-accent**. This is for the blue text in italics that says "Specialty coffee". This text should be wrapped in a set of **<span>** tags. The element is an inline tag often used to apply separate styles to part of a paragraph in order to make it stand out.

8. block: **header**, element: **description**.

9. block: **header**, element: **image**.

10-13. block: **header**, element: **paragraph**. Think about how you can group the three paragraphs as shown in the design. Hint: your knowledge of flexbox and a few divs can help.

### ! Pro tip:

How can you break lines of text into multiple lines and control the exact placement of those breaks? You could use `<br>` but this may cause accessibility issues due to the way `<br>` is handled by screen readers.

One trick is to wrap the desired lines in **<span>** elements with `display: block;` applied. Another option could be to use a non-breaking space: a special space character that can't be broken by a new line. Simply insert `&nbsp;` in your HTML between the words that should stick together.



## Sizes and spacings

Add all the sizes according to the picture above.

**1** Menu Recipes Book a table Contacts

**2** Specialty coffee in the Practicum Library

**3** Welcome to our study zone! We've got the coffee to energize you and the cozy vibes to help you get creative.

**4** Hours:  
Monday – Friday (10:00 – 19:00)  
Saturday – Sunday (11:00 – 18:00)

**5** 200 Success Avenue, The Town of Practicum

```
1 font-family: Inter;
font-style: normal;
font-weight: 500;
font-size: 16px;
line-height: 20px;
```

```
2 font-family: Noto Serif;
font-style: italic;
font-weight: normal;
font-size: 60px;
line-height: 72px;

color: #2F80ED;
```

```
3 font-family: Inter;
font-style: normal;
font-weight: 500;
font-size: 60px;
line-height: 72px;
```

```
4 font-family: Noto Serif;
font-style: normal;
font-weight: normal;
font-size: 20px;
line-height: 30px;
```

```
5 font-family: Inter;
font-style: normal;
font-weight: 500;
font-size: 16px;
line-height: 20px;
```

## Typography



During the next sprint, you'll start working in Figma, which is design software that also prepares some of the CSS styles for developers. However, only a few of those styles can be safely copied into our CSS files. Font styles can be safely copied, but this should still be done carefully. We'll practice doing this in the Typography portion of the brief.

The numbered code snippets correspond to their respective numbered text elements. We don't need to duplicate the styles to multiple CSS selectors — take a moment to consider the reason why.

First, remember the styles we assigned to the [page](#) block? They are inherited by all of its child elements, meaning we don't need to assign the same styles again.

Second, you don't need to write values that are the same as the defaults, e.g. `font-style: normal;` and `font-weight: normal;` (which is equivalent to `font-weight: 400;`).

With all this in mind, write the necessary styles for the necessary text elements in CSS.



While this task may seem as simple as copy-pasting, it's still part of the work of a web developer. By doing this task, you're completing a certain portion of your project and ensuring that the webpage looks great for your users. Moreover, having touched on these routine tasks, it'll be easier for you to understand why these new features were added to CSS and why developers came up with automation tools and some other useful technologies. We will learn some of them later on in the program.

## 2. Section: Menu



### Required elements and class names

**Background:** Set the background color to `#0a2750`. For the background image, provide the URL to [background-menu.svg](#). The background doesn't repeat and its position is centered. Set the size value to `cover`.

1. block: `menu`, element: `title`.
2. block: `menu`, element: `subtitle`.

3. block: `menu`, element: `cards`. Notice how the third card is centered. It may look difficult at first, but this is where flexbox shows its true potential.

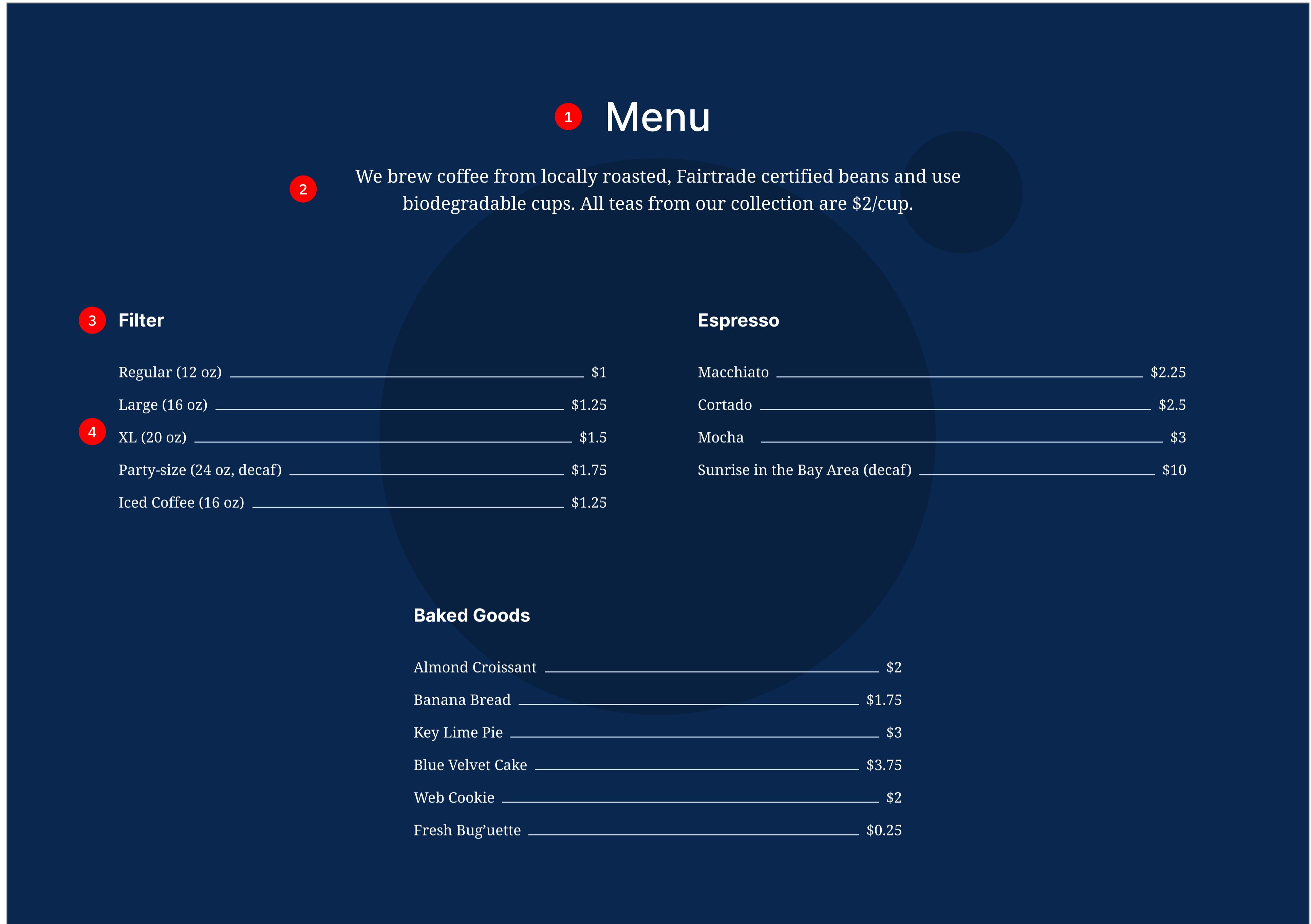
4. block: `card`.
5. block: `card`, element: `title`.
6. block: `card`, element: `list-item`.
6. block: `card`, element: `list-divider`.



## Sizes and spacings

Add all the sizes according to the picture above.

All blocks should be centered inside the section.



**1** `font-family: Inter;  
font-style: normal;  
font-weight: 500;  
font-size: 44px;  
line-height: 52px;`

**2** `font-family: Noto Serif;  
font-style: normal;  
font-weight: normal;  
font-size: 20px;  
line-height: 30px;`

**3** `font-family: Inter;  
font-style: normal;  
font-weight: bold;  
font-size: 20px;  
line-height: 30px;`

**4** `font-family: Noto Serif;  
font-style: normal;  
font-weight: normal;  
font-size: 16px;  
line-height: 20px;`

## Typography

The numbered code snippets correspond to their respective numbered text elements.

### 3. Section: Recipes

**1** Recipes

**2** Check out some recipes we've collected for your home-brewing convenience:

**3**

**4**

**5**

Aeropress recipe      ~5 min      French press recipe      ~15 min

#### Required elements and class names

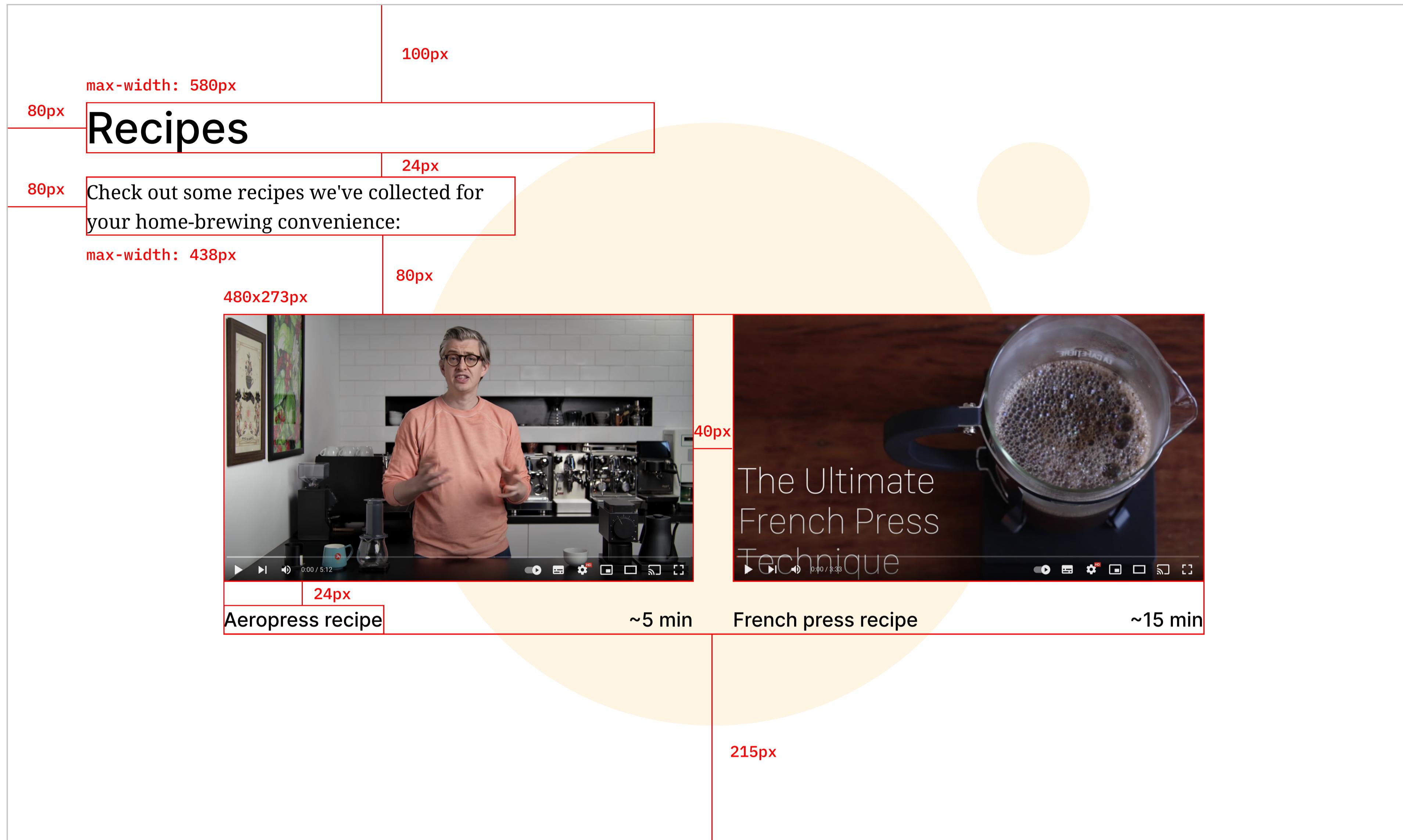
**Background:** Set the background color to `#FFFFFF`. For the background image, provide the URL to [background-recipes.svg](#). The background doesn't repeat and its position is centered. Set the size value to `cover`.

Use the `<iframe>` tag to add videos. You can use these links or choose your own (make sure to change the description and time accordingly):

[https://www.youtube.com/watch?v=j6VIT\\_jUVPc](https://www.youtube.com/watch?v=j6VIT_jUVPc)

<https://www.youtube.com/watch?v=st571DYYTR8>

1. block: `recipes`, element: `title`.
2. block: `recipes`, element: `subtitle`.
3. block: `recipes`, element: `videos`.
4. block: `recipes`, element: `iframe`.
5. block: `recipes`, element: `video-caption`.



## Sizes and spacings

Add all the sizes according to the picture above.

The `videos` element should be centered inside the section.

# 1 Recipes

Check out some recipes we've collected for your home-brewing convenience:



3 Aeropress recipe ~5 min



French press recipe ~15 min

1  
font-family: Inter;  
font-style: normal;  
font-weight: 500;  
font-size: 44px;  
line-height: 52px;

2  
font-family: Noto Serif;  
font-style: normal;  
font-weight: normal;  
font-size: 20px;  
line-height: 30px;

3  
font-family: Inter;  
font-style: normal;  
font-weight: 500;  
font-size: 20px;  
line-height: 30px;

## Typography

The numbered code snippets correspond to their respective numbered text elements.

## 4. Section: Reservation

**1 Book a table**

**2**

**3**

**4**

**5**

**6**

**7**

### Required elements and class names

**Background:** Set the background color to `#D9E7FF`. For the background image, provide the URL to [background-reservation.svg](#). The background doesn't repeat and its position is centered. Set the size value to `cover`.

Use inputs of the appropriate type: text, number, date, email, checkbox. Specify placeholder texts as shown in the design. All fields, including the checkbox, should be required.

1. block: `reservation`, element: `title`.

2. block: `reservation`, element: `form`. Also, mix it with the `form` block. For `reservation__form`, define the width and margins. Remember, we can only specify these properties for child elements according to BEM. The `form` block and its elements will contain all the styles that apply to the form, thus making it reusable.
3. block: `form`, element: `fieldset`.
4. block: `form`, element: `label`.

Checkbox default state ↓

I agree with the terms of use

Checkbox checked state ↓

I agree with the terms of use

5. block: `form`, element: `input`. Make a `border-radius` of `4px`.

Notice that you can't easily control CSS styles for the date input because each browser defines its own UI for it. That's why it's common practice to leave the standard styles. Accordingly, you should keep the standard styles for this project as well.

In general, not everything from a design can be easily implemented, and in some situations, without using JavaScript workarounds, it's virtually impossible. In such a case, the time-cost ratio may not be reasonable for the business, and the eventual solution may not be readily scalable. Accordingly, when dealing with something like this, it's good practice to communicate with the designer and find the most appropriate solution together.

6. block: `form`, element: `button`. Use the appropriate HTML tag for the button. Make a `border-radius` of `4px`. Create a hover state, so that the opacity changes from 100% to 70%. Add smooth transition for the `opacity` property.

7. block: `form`, element: `checkbox`. Make a `border-radius` of `2px`. Ensure that the text is on the same level as the checkbox.

You can keep the standard checked styles for it. However, in this case, remember to remove the `check.svg` from the images folder.

If you want to challenge yourself, you can also style it as follows: set the background color to `#2F80ED`, and a background image of `check.svg`.

Checkbox default state ↓

I agree with the terms of use

Checkbox checked state ↓

I agree with the terms of use

100px

80px

8px

32px

12px

32px

120px

8x8px

**Book a table**

Name \*

Name Surname `max-width: 385px;`

Number of guests \*

1-8

Date & time \*

dd.mm.yyyy

Your email \*

email@email.com

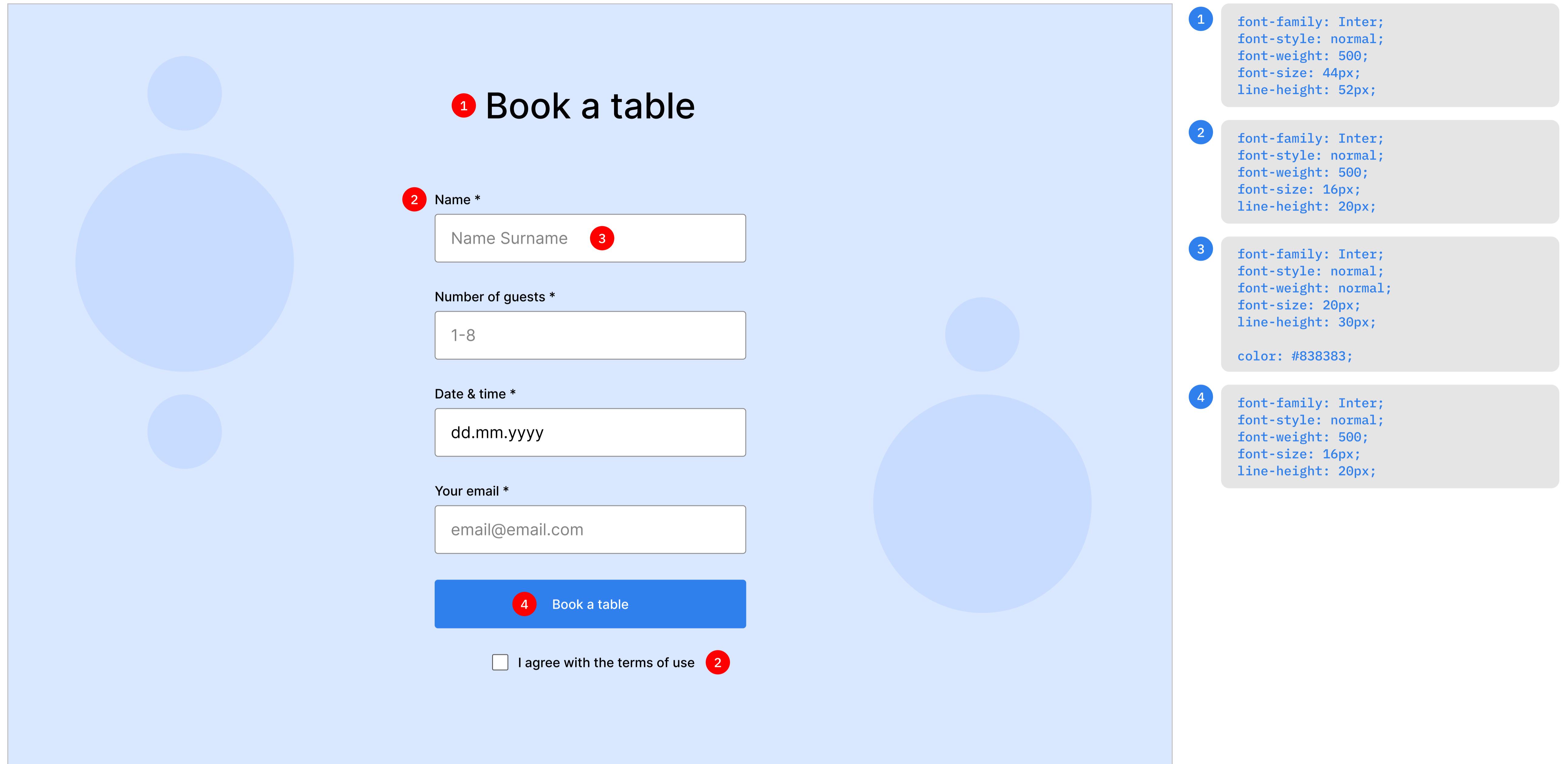
I agree with the terms of use

Book a table

## Sizes and spacings

Add all the sizes according to the picture above.

The blocks should be centered inside the section.



# 1 Book a table

2 Name \*

Name Surname 3

Number of guests \*

1-8

Date & time \*

dd.mm.yyyy

Your email \*

email@email.com

4 Book a table

I agree with the terms of use 2

1

```
font-family: Inter;
font-style: normal;
font-weight: 500;
font-size: 44px;
line-height: 52px;
```

2

```
font-family: Inter;
font-style: normal;
font-weight: 500;
font-size: 16px;
line-height: 20px;
```

3

```
font-family: Inter;
font-style: normal;
font-weight: normal;
font-size: 20px;
line-height: 30px;
color: #838383;
```

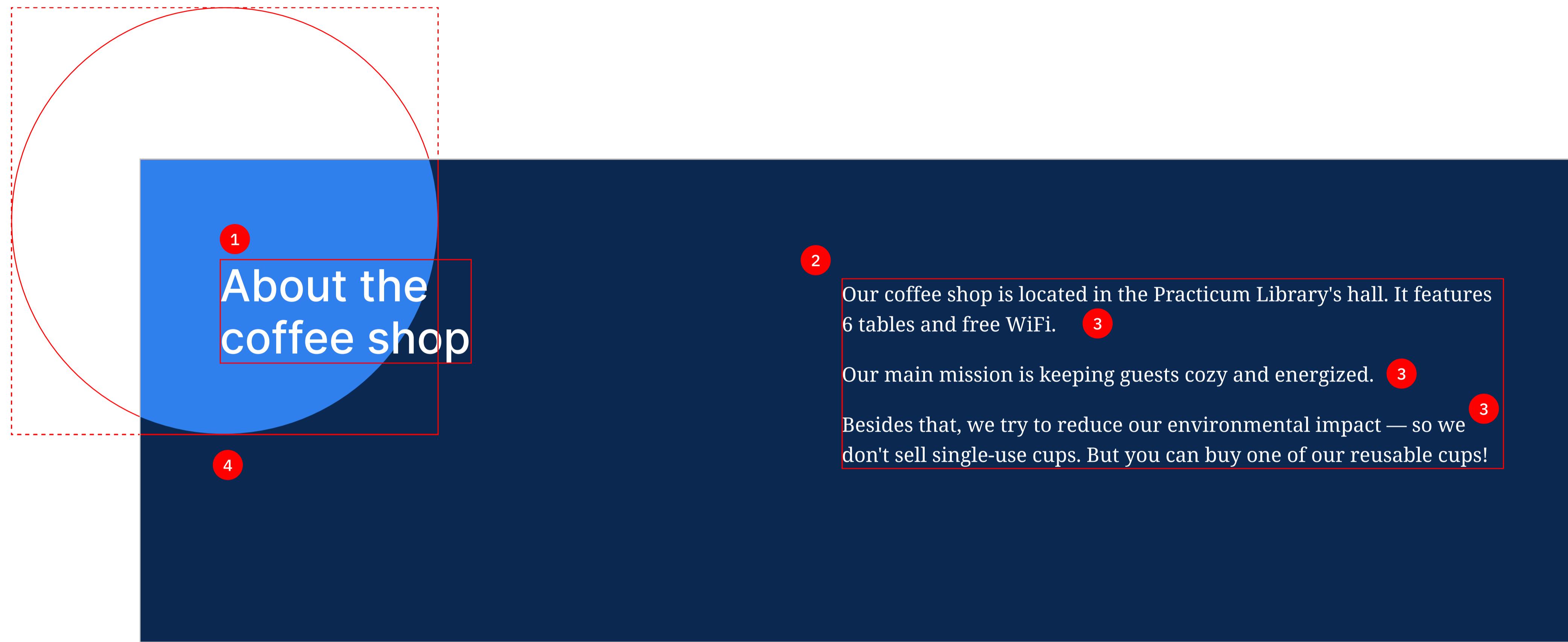
4

```
font-family: Inter;
font-style: normal;
font-weight: 500;
font-size: 16px;
line-height: 20px;
```

## Typography

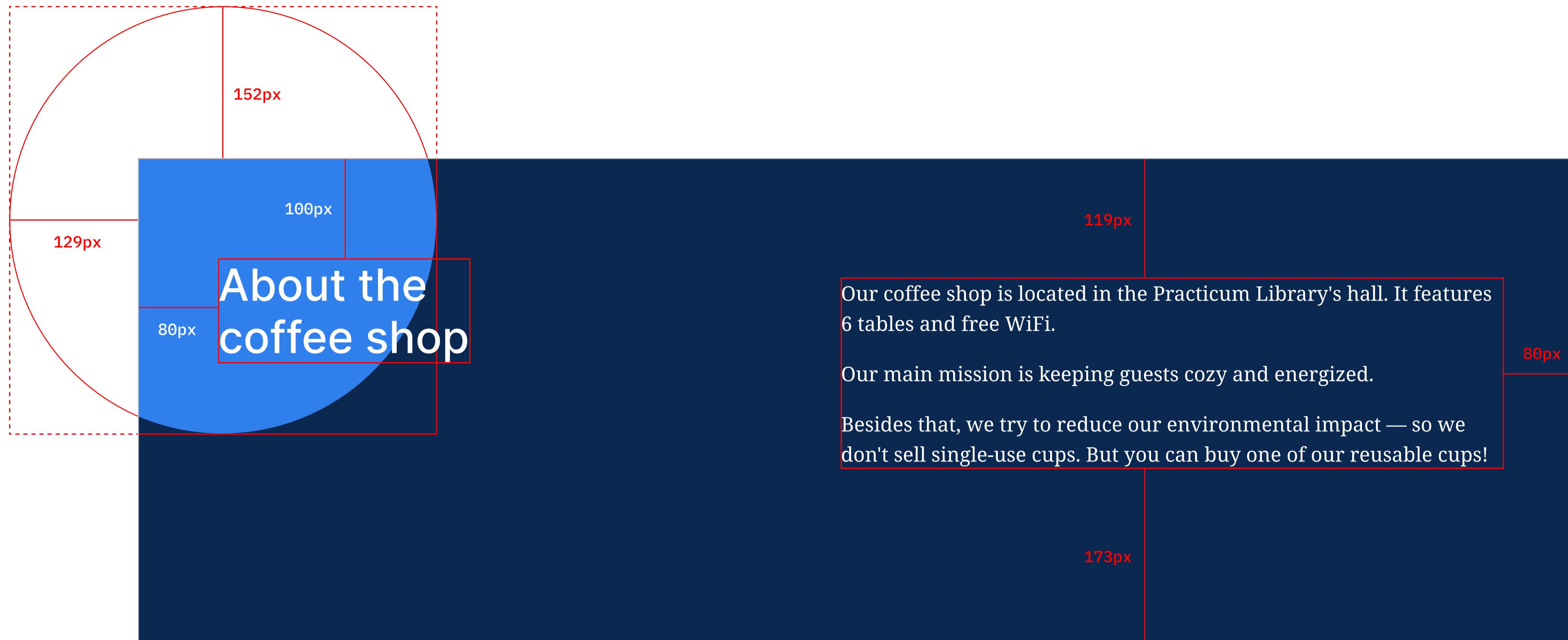
The numbered code snippets correspond to their respective numbered text elements. Notice that the color of placeholder text is different.

## 5. Section: About the coffee shop



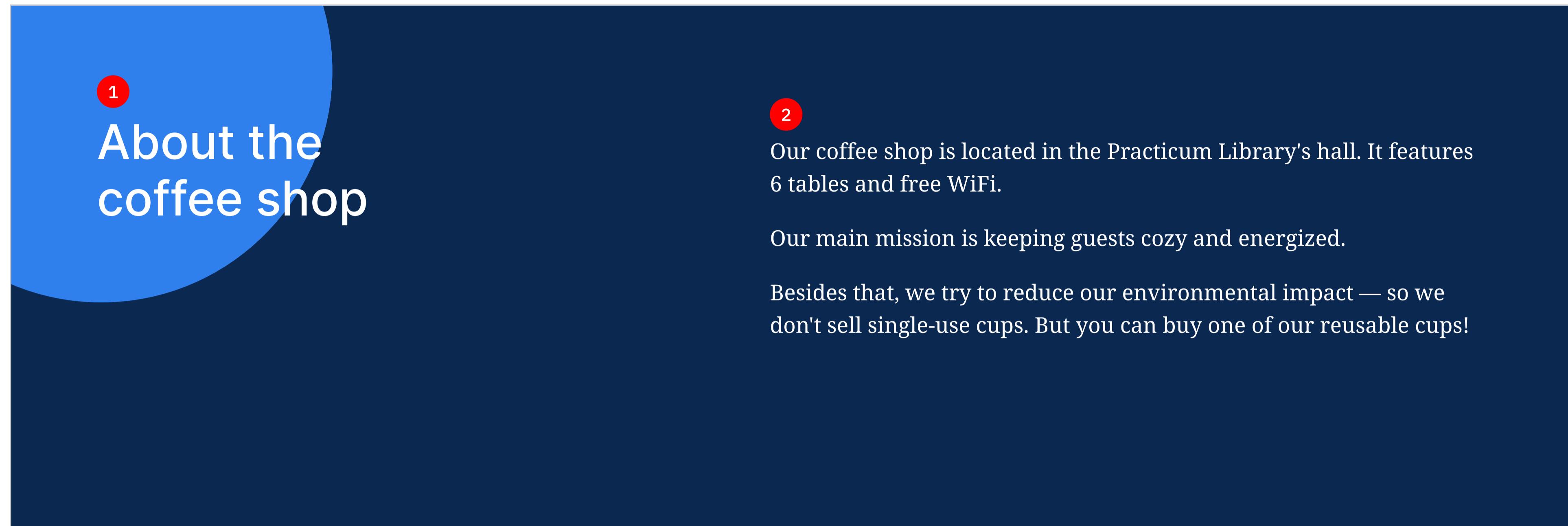
### Required elements and class names

1. block: `about`, element: `title`.
2. block: `about`, element: `content`.
3. block: `about`, element: `paragraph`.
4. block: `about`, element: `circle`.



## Sizes and spacings

Add all the sizes according to the picture above.

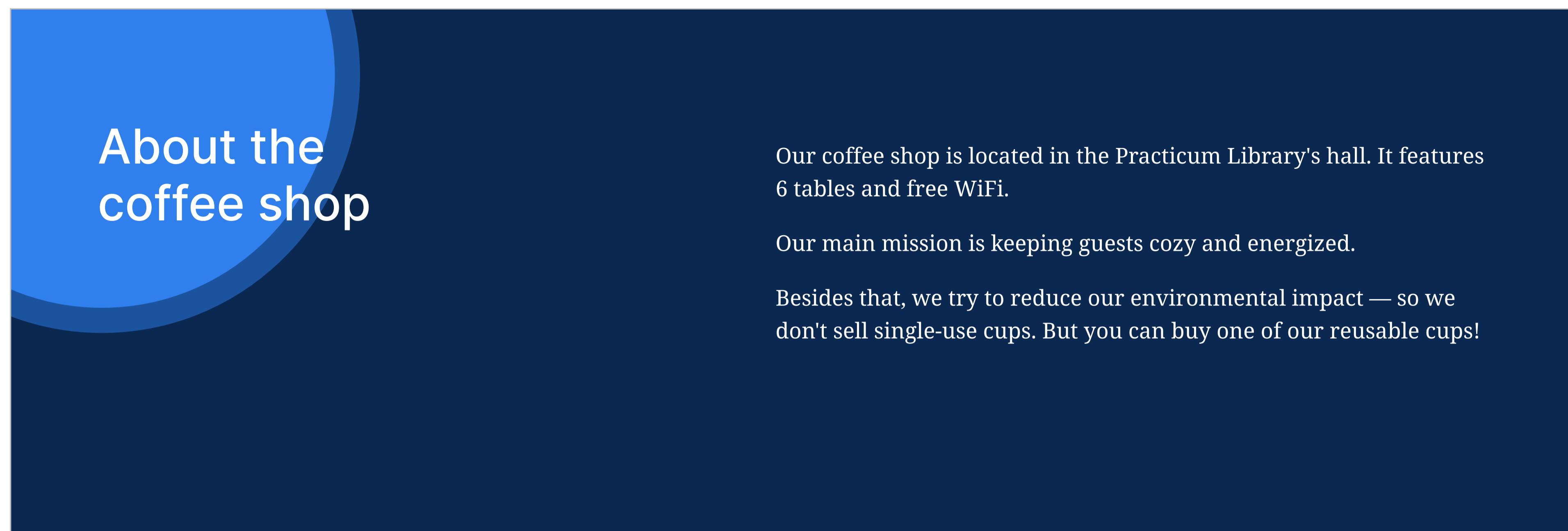


1  
font-family: Inter;  
font-style: normal;  
font-weight: 500;  
font-size: 44px;  
line-height: 52px;

2  
font-family: Noto Serif;  
font-style: normal;  
font-weight: normal;  
font-size: 20px;  
line-height: 30px;

## Typography

The numbered code snippets correspond to their respective numbered text elements.



## About the coffee shop

Our coffee shop is located in the Practicum Library's hall. It features 6 tables and free WiFi.

Our main mission is keeping guests cozy and energized.

Besides that, we try to reduce our environmental impact — so we don't sell single-use cups. But you can buy one of our reusable cups!

## Animation

The blue circle will be animated to pulsate like a heart. You can see the animated version in the project description on the platform.

We need our current `circle` element to remain static, so for the animation, we'll add a new one. Duplicate the `<div>` element with the static circle with its class, then add a BEM modifier to the duplicated one:

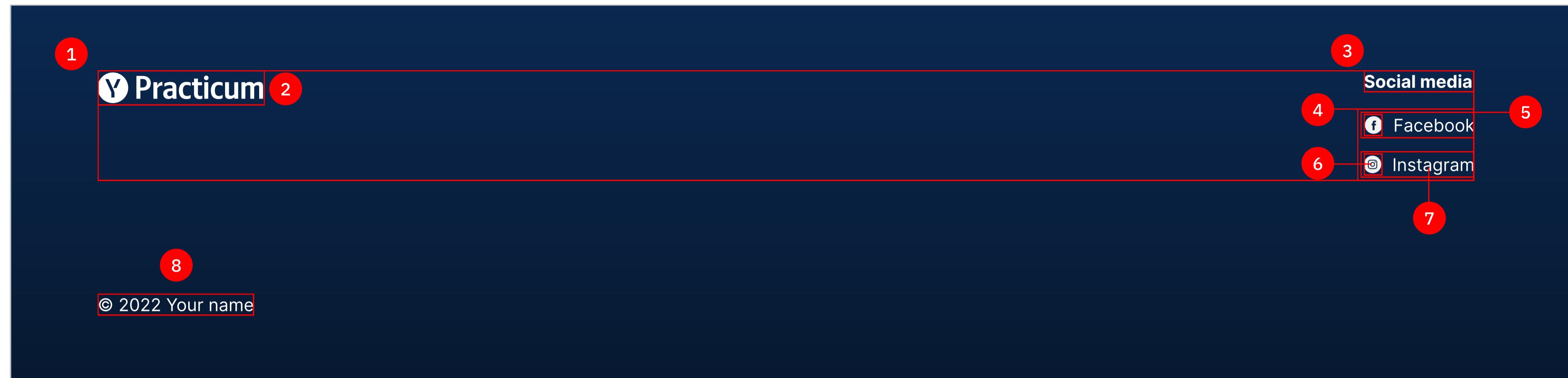
[about\\_\\_circle\\_animation\\_blurred](#).

Declare the animation with the `@keyframes` at-rule. You should give it a meaningful name, such as `pulsate`. The animation should have three frames:

1. At the start of the animation, make the animated circle invisible using the `opacity` property.
2. During the middle stage (`50%`), increase the opacity to `0.6`.
3. When the animation is complete, apply the `transform` property to scale the circle. In the scale function, specify any value, to your liking, larger than `1` but not larger than `1.5`. Make the circle invisible again.

Implement the `animation` to the circle by adding the `animation` property to the `circle` modifier. Specify the animation's duration (`1.3s`), the timing function should be `ease-in-out`. There should be no delay, and the animation should be infinite.

## 6. Footer



### Required elements and class names

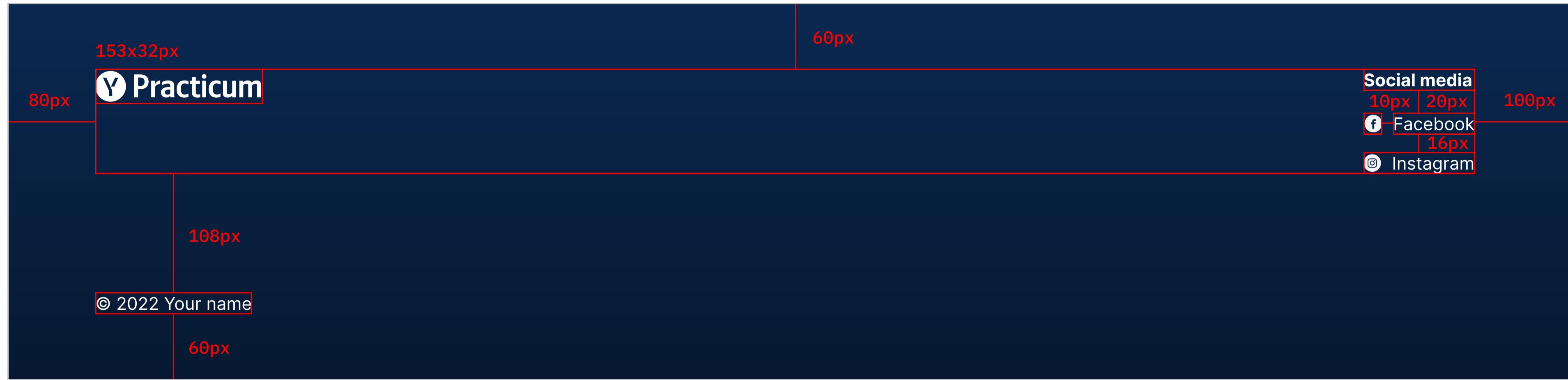
**Background:** notice the fancy gradient. Figma provides you with ready-made code for gradients, so let's use the code for this one:

```
background: linear-gradient(180deg, #0A2850 0%, #071931 100%);
```

1. block: **footer**, element: **content**.
2. block: **footer**, element: **logo**.
3. block: **footer**, element: **social-heading**.
4. block: **footer**, element: **list**.
5. block: **footer**, element: **list-item**.
6. block: **footer**, element: **social-icon**.
7. block: **footer**, element: **social-link**.

Create a hover state for links so they smoothly transition into a background color of **#C1C1C1**. You can specify any transition properties you'd like, such as duration or timing function.

8. block: **footer**, element: **copyright**. Remember to write your name here. 😊



## Sizes and spacings

Add all the sizes according to the picture above.



## Typography

The numbered code snippets correspond to their respective numbered text elements.

1  
font-family: Inter;  
font-style: normal;  
font-weight: bold;  
font-size: 16px;  
line-height: 20px;

2  
font-family: Inter;  
font-style: normal;  
font-weight: normal;  
font-size: 16px;  
line-height: 20px;

3  
font-family: Inter;  
font-style: normal;  
font-weight: normal;  
font-size: 16px;  
line-height: 20px;

# Stage 3. Self-review

You are nearing the project finish line, and you've already completed the hardest part. Still, there is one last crucial task before you complete your project: conducting the self-review. To help, we've prepared a checklist:

- Compare each section of your webpage against the brief (open your project side-by-side with the brief and go through each point. Consider jotting down notes for yourself with things that you want to remember).
- Check the validity of your code using a [validator](#).
- Check your project against all the checklist items (you'll find this in the project description on the platform).
- Make sure to remove any redundant comment lines. Keep the comments that help you better understand the project structure and facilitate your understanding of complex features.

And with that, you've finally finished all the tasks for your second project at Practicum. Awesome job!