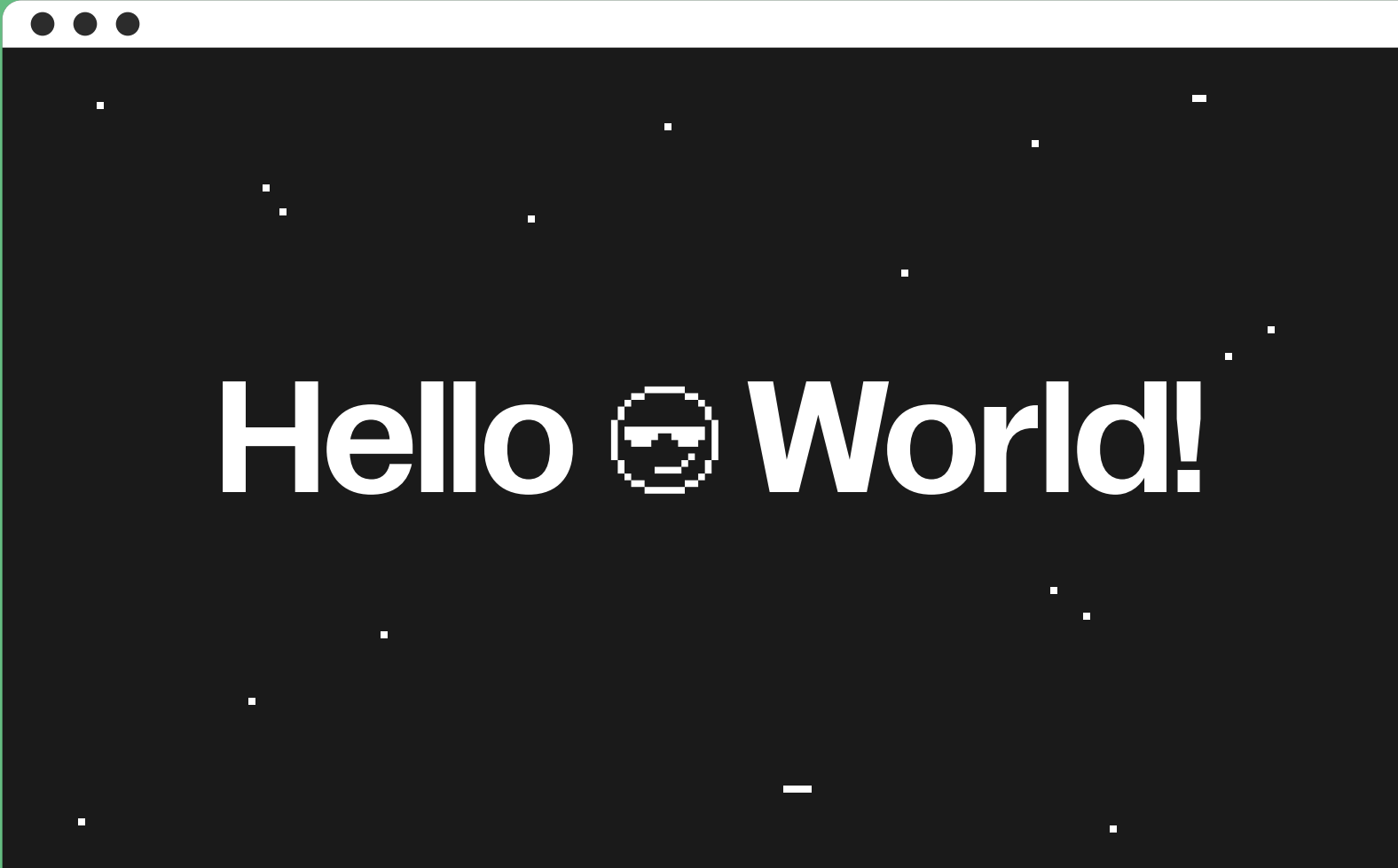


Your first independent project

Basic HTML and CSS

Project brief



Project structure and starter code	2
Final result	3
Header	4
Adding a background image	5
The main page heading	6
Cards	7
Adding markup for the cards	9
Styling the cards	10
Footer	12

Project structure and starter code

To start the project, follow the directions on the platform project page to download the starting code archive, extract the files, and open the folder in VSCode. If you open the `index.html` and `style.css` files you'll notice they each already contain some code.

Here's a breakdown of the code that's already been written for you:

- The HTML file already contains the basic structure for your web page and includes some comments to show you where to write the rest of your code.
- In the `fonts` folder, we added the font "Poppins" that you'll use for headings in your project. We did it for you this time so you can start using the font immediately, but in the first sprint of the main program, you'll learn how to add custom fonts to your own website.
- Recall that the browser automatically applies styles to some HTML elements, such as the default margins that are often present. At the top of `style.css`, we've added a CSS rule that overrides this behavior by resetting the margins (that is, by setting them to `0`).

- We've also applied the `box-sizing: border-box;` declaration to all elements using the `*` selector. Recall that the `border-box` value means that the values of the `border` and `padding` properties are included in the `width` and `height` values specified for an element.
- Note that in the HTML file, the `<body>` element has the `page` class. In the CSS file, we've create a rule with the `.page` selector, which applies a default font and background color for the page.

There's more code than this in the starter kit, but we'll explain the rest in more detail in the instructions for each section.

SOFTWARE ENGINEER

MEGAN O'SULLIVAN



Music

I'm a singer in an all-female power metal band called "Great Vengeance." We rehearse in my garage and play at local shows here in Oregon every couple of months.



Friends

I love hanging out with my girls after rehearsals, sipping on yerba maté, and watching wrestling. I hope we go on a big tour one day.



Family

My partner Sean and I are proud parents of two. Our older one, Lola, is a tween and baby Payton is one and a half. They're a lot of fun, but sometimes they can be vvvquite a handful.



Software Engineering

I'm super excited to learn how to program. I've been a full-time mom for a while and I'm ready to get back to work and start a new career.

© Megan O'Sullivan 2023

Header

Let's start building the project with the header. If you look in the HTML file, you'll notice that there's already a `<header>` element with the `header` class. If you look in the CSS file, you'll see that we've already created a rule for this class.

Right now, it just contains the `height: 660px;` declaration. You'll add some more styles soon.



Adding a background image

Now it's time for you to start writing some code. Start off by adding a background image to the header. Here's what you need to do:

- Add a picture of your choice to the `images` folder.
- Then, use the `background-image` property to specify a path to it. To specify the path to your image, the value of the `background-image` property should look something like this: `url(./images/name-of-file.jpg)`.
- Use `background-size`, `background-position`, and/or other background image related properties to set the image's size and position.
- Open the `index.html` file in your browser to make sure it looks good.

You may have noticed that the CSS file contains a rule for the `.overlay` selector. This superimposes a semi-transparent layer on top of the background image using the `background-color` property.



Pro tip

To choose photos, you can use free stock photo resources like [Unsplash](#) or [Pexels](#)

The main page heading

Next, we'll work on the main page heading. Open the `index.html` file and look for the comment that says `<!-- add the page title here -->`. Replace this comment with an `<h1>` element. Give it a descriptive class name and write your name between the opening and closing tags. Refresh the page in the browser. Your name should appear at the bottom of the page.

Next, you need to add some styles to your heading. Create a new CSS rule in the `style.css` file using the class name you created in the selector. Add the following declarations to the rule:

- `font-size: 140px;`
- `font-weight: 500;`
- `line-height: 132px;`
- `letter-spacing: -5.6px;`
- `color: #ffe247;`
- `text-transform: uppercase;`
- `margin: 0 auto 20px` — this will remove the element's default top border, set its bottom border to `20px`, and center it horizontally.
- `max-width: 1400px;` — this will limit the width of the text element.
- `text-align: center;` — this will make sure that if there's too much text to fit on one line, then the text will still appear centered.

After making these changes, check your webpage in the browser to make sure it looks correct.



Pro tip

You may be wondering why the text elements in the header are appearing the way they are, with so much space between them. This is because of three properties we added to the `.overlay` element:

```
display: flex;
flex-direction: column;
justify-content: space-between;
align-items: center;
```

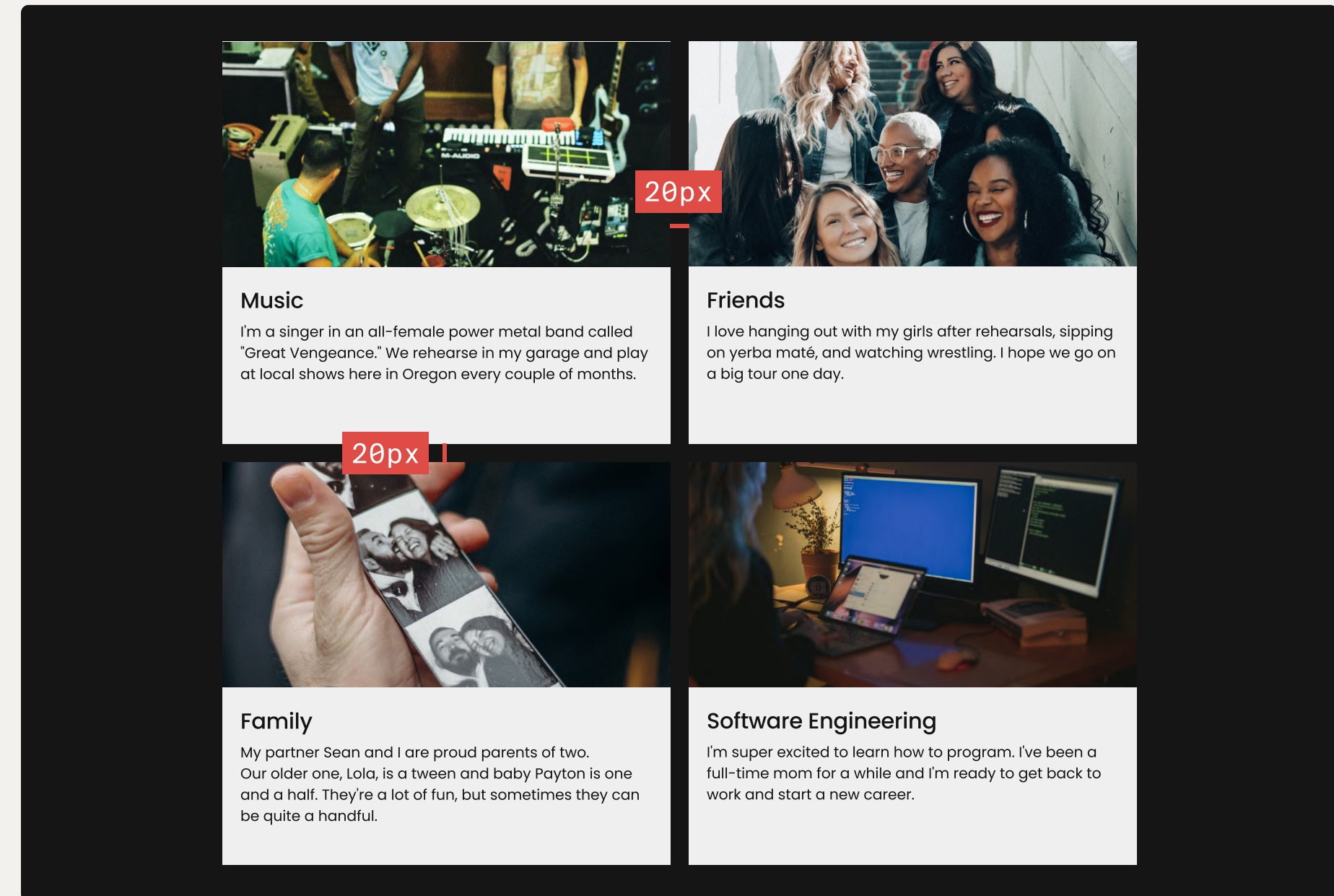
Don't worry about how this works now. You'll learn all about flexbox in the next sprint.

Cards

7

After the header, it's time to move on to the page's main content. This is where you'll build the cards that show interesting facts about yourself. There should be four cards in the main content area, and they should be centered horizontally and wrapped onto two lines. To make this happen, we've already added three CSS declarations to the rule with the `.content` selector:

- `display: flex;`
you're already a little familiar with this one. It provides us with a wide range of options for arranging our elements. You'll learn all about these options in the full program, but we'll use the following two options for this project.
- `flex-wrap: wrap;`
when this declaration is used, items within a container will wrap onto a new line if there isn't enough room for all of them with their specified dimensions. Otherwise, by default, `display: flex` would cause the elements to squeeze onto one line.
- `justify-content: space-between;`
this declaration evenly distributes the items within its container. In our case, the container for the cards will be the `<main>` element with the `content` class.
- We've also added `gap: 20px;`, which creates space between elements inside a flex container.




Before creating the cards, add the following declarations to the CSS rule with the `.content` selector to define the dimensions of the container:

- `max-width: 1000px;`
- `margin: 40px auto;`

Also, add the `background-color: #151515;` declaration to the `.page` selector.


1000px

40px




Music

I'm a singer in an all-female power metal band called "Great Vengeance." We rehearse in my garage and play at local shows here in Oregon every couple of months.



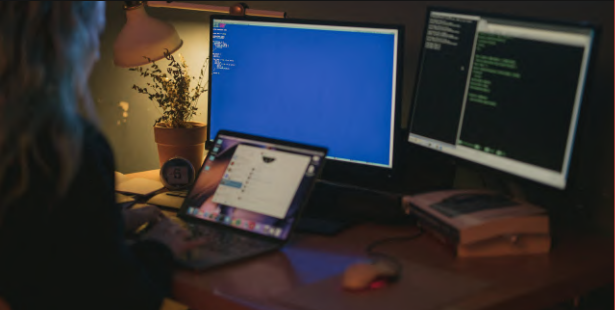
Friends

I love hanging out with my girls after rehearsals, sipping on yerba maté, and watching wrestling. I hope we go on a big tour one day.



Family

My partner Sean and I are proud parents of two. Our older one, Lola, is a tween and baby Payton is one and a half. They're a lot of fun, but sometimes they can be quite a handful.



Software Engineering

I'm super excited to learn how to program. I've been a full-time mom for a while and I'm ready to get back to work and start a new career.

40px

Adding markup for the cards

Let's get back to the HTML file. Note that every card on the design has the same structure. It contains an image, title, and text. Inside the `<main>` element, create a `<div>` element with the `card` class. This will be the parent element for an individual card's content. Inside this `<div>` element, create an `` element. Give it a descriptive class (such as `card__image`) and the following attributes:

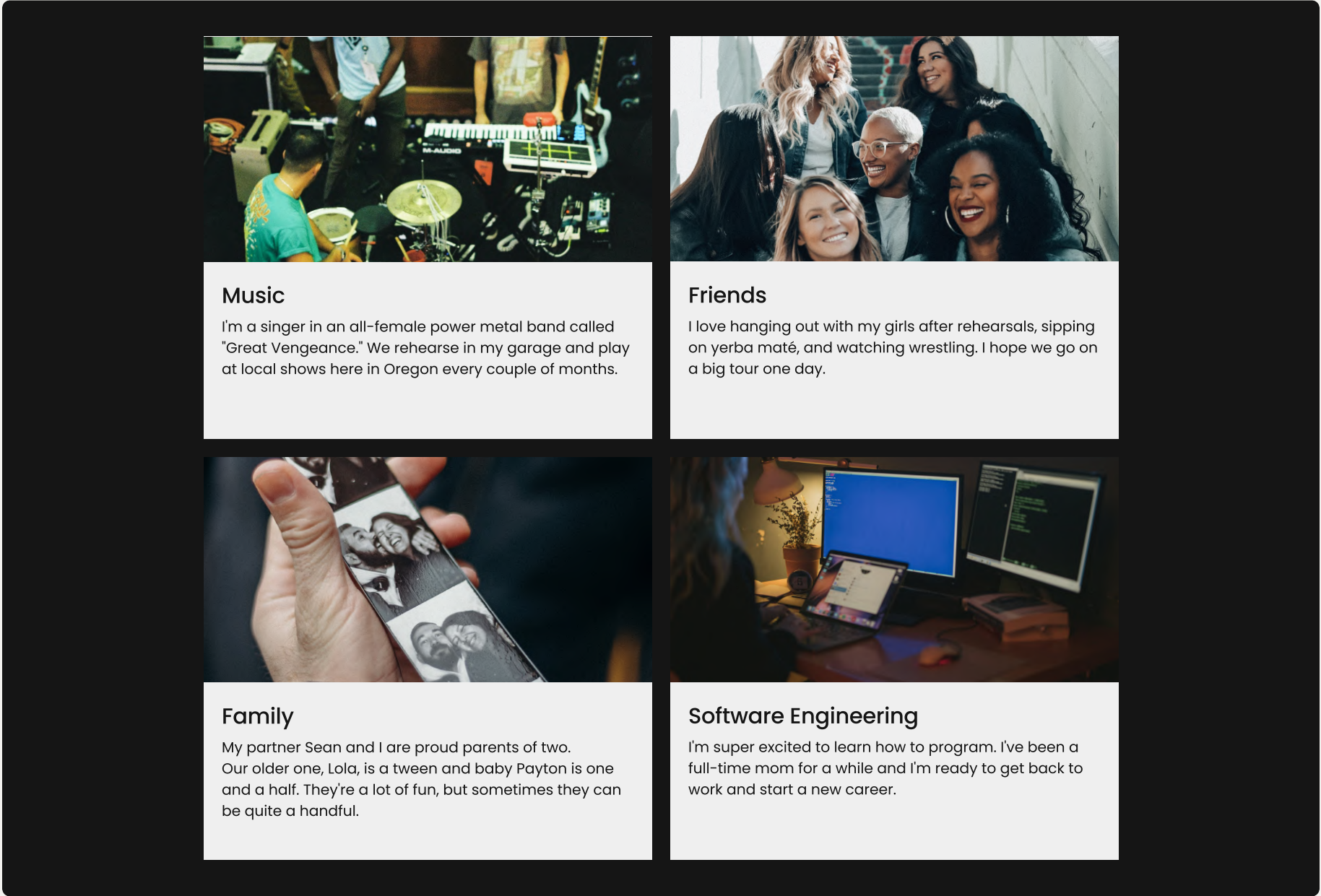
- `src` to specify a path to the image
- `alt` to add an alternative text for the image

Beneath the `` element, add another `<div>`. Give it the class `.card__content`.

Inside the `card__content` element, add an `<h2>` element for the card title and a `<p>` element for its description. Assign class names to them as well.

Now you have the basic structure for one card. You need four cards, so copy the `<div>` with the `card` class and its contents, and paste it three more times below.

Now you can start adding some content to the cards. Add the images you want to the `images` folder, then link to them via the `src` attribute of each `` element. Fill the text elements with information about yourself, and add alternative texts to the images via their `alt` attribute.



Styling the cards

Next, let's style our cards. Add a CSS rule with the `.card` selector with the following declarations:

- `width: 490px;`
- `min-height: 440px;`
- `background: #ffffff;`

Next, add a rule for the card image with the following declarations:

- `width: 100%;`
- `display: block;`
- `object-fit: cover;` — this declaration makes the element fit within the fixed content box while preserving its aspect ratio.
- `height: 246px;`— this declaration sets a fixed height, as your own images may vary in size.

Add a CSS rule for the card title (i.e., the `<h2>` element). Don't forget to specify styles for fonts and the bottom margin:

- `font-size: 24px;`
- `line-height: 1.5;`
- `font-weight: 500;`
- `letter-spacing: -0.16px;`
- `color: #151515;`
- `margin-bottom: 10px;`



Styling the cards

Give the element with the `card__content` class the `padding: 20px 20px 0;` declaration. This will create the whitespace surrounding the text content, except for the whitespace on the bottom of the card, which will be created automatically based on the card element's height.

The CSS rule for the card description should contain the following styles:

- `font-size: 16px;`
- `line-height: 1.5;`
- `font-weight: 400;`
- `color: #151515;`



Music

I'm a singer in an all-female power metal band called "Great Vengeance." We rehearse in my garage and play at local shows here in Oregon every couple of months.



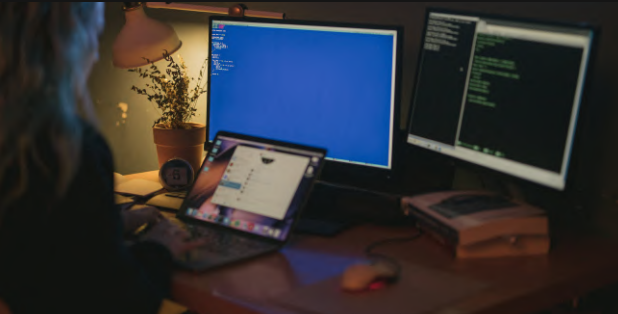
Friends

I love hanging out with my girls after rehearsals, sipping on yerba maté, and watching wrestling. I hope we go on a big tour one day.



Family

My partner Sean and I are proud parents of two. Our older one, Lola, is a tween and baby Payton is one and a half. They're a lot of fun, but sometimes they can be quite a handful.



Software Engineering

I'm super excited to learn how to program. I've been a full-time mom for a while and I'm ready to get back to work and start a new career.

Footer

Let's move to the footer. Set the `padding` according to the design:

- `padding: 20px 0 40px;`

Add your signature inside the footer using a `<p>` tag. Give the `<p>` tag a class and the following styles:

- `font-size: 16px;`
- `line-height: 1.5;`
- `font-weight: 400;`
- `color: #efefef;`
- `text-align: center;`



Congratulations on completing your project!
Now you can return to the platform to submit it for code review.

