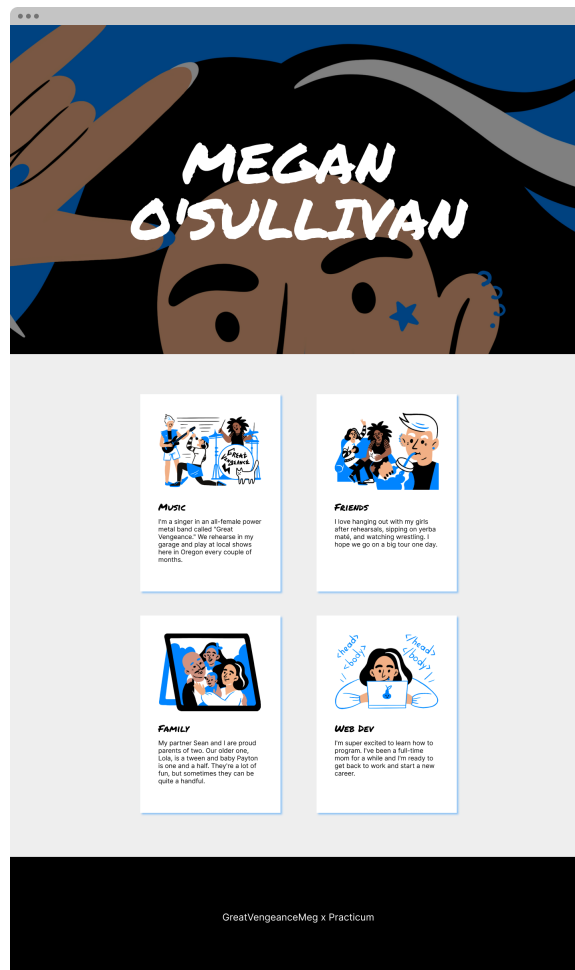




"Sobre Mim" — Resumo do Projeto



Estrutura do projeto e código inicial

Para começar, baixe o kit inicial na página "Seu primeiro projeto independente". Depois de baixar o arquivo, você deve ter uma pasta chamada "Sobre Mim Código Inicial" no seu computador. Se você abrir ela, verá que essa pasta contém quatro itens:

- Uma pasta `fonts/` contendo arquivos com a fonte customizada `Permanent Marker`. Você não precisa fazer nada com essa pasta. Esse é o local onde aquela fonte

legal do projeto de exemplo está armazenada. Isso significa que você poderá usar a mesma fonte para o seu projeto.

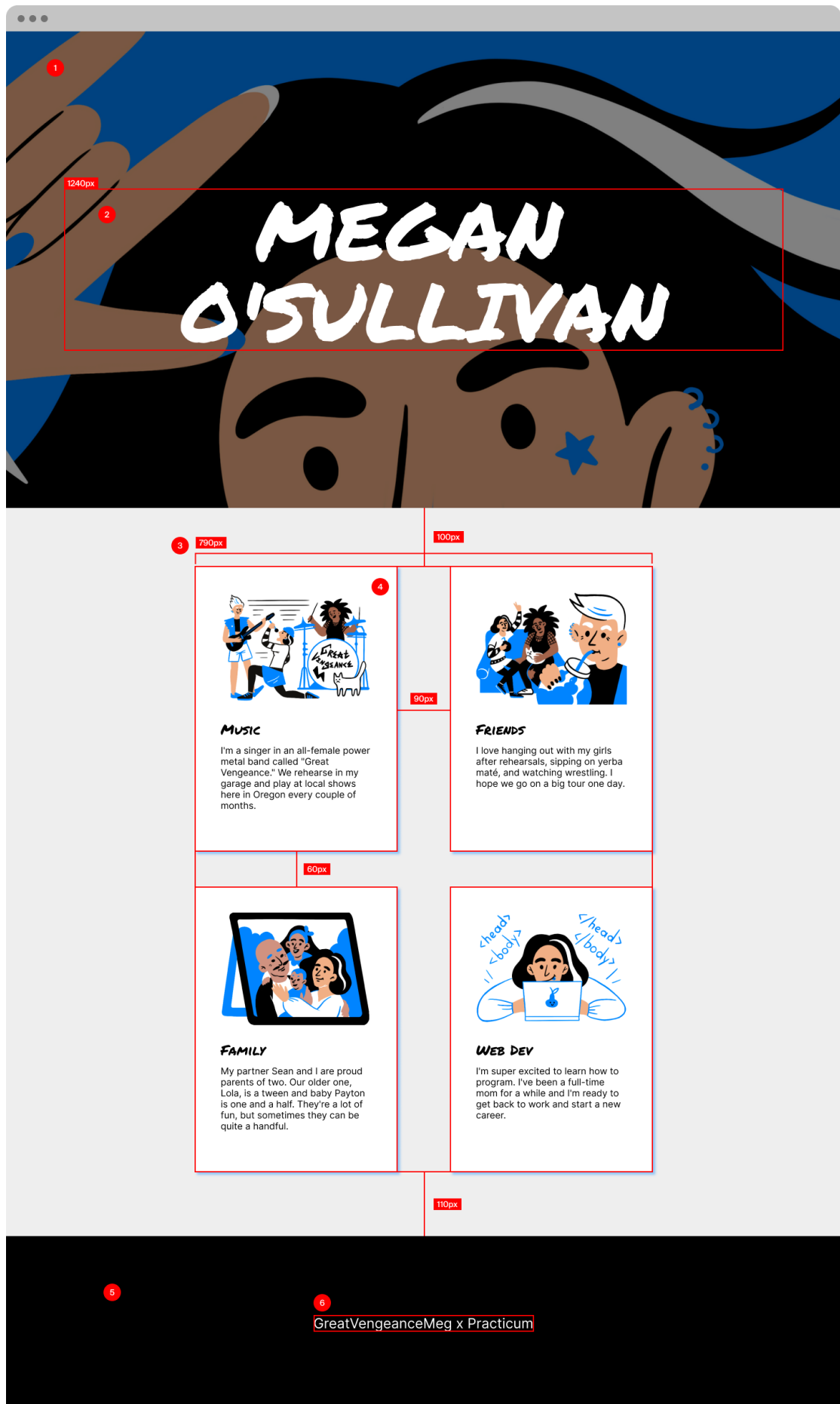
- Uma pasta `images/` para armazenar imagens que serão usadas no projeto. Para exibir imagens nos quatro cartões de sua página web, primeiro você precisará adicionar essas imagens a esta pasta para poder acessá-las.
- Os dois últimos itens são os arquivos `index.html` e `style.css`.

Abra o `index.html` e `style.css` no VSCode. Você perceberá que cada um deles já contém algum código.

Aqui está uma explicação do código que já foi escrito para você:

- O arquivo HTML já contém a estrutura básica de sua página web e inclui alguns comentários para mostrar onde escrever o resto do código.
- No arquivo CSS, a fonte personalizada já foi carregada para você usando a regra `@font-face`. A propriedade `font-family` é usada para especificar um nome de fonte para o projeto. A propriedade `src` contém links para os dois formatos de arquivo.
- Lembre-se de que o navegador aplica automaticamente estilos a alguns elementos HTML. Após a regra `@font-face`, adicionamos uma regra CSS que substitui esse comportamento, redefinindo as margens, tamanho da fonte e peso para o elemento `<body>`, títulos, parágrafos e links.
- Também aplicamos a declaração `box-sizing: border-box;` a todos os elementos usando o seletor `*`. Lembre-se de que o valor `border-box` significa que os valores das propriedades `border` e `padding` são incluídos nos valores `width` e `height` especificados para um elemento.
- Observe que no arquivo HTML, o elemento `<body>` tem a classe `page`. No arquivo CSS, criamos uma regra com o seletor `.page`, que aplica uma fonte padrão e uma cor de fundo para a página.

Há mais código do que isso no kit inicial, mas explicaremos o resto com mais detalhes nas instruções de cada seção.



Cabeçalho



(1)

Vamos começar a construir o projeto com o cabeçalho, ou header. Se você olhar no arquivo HTML, notará que já existe um elemento `<header>` com a classe `header`. Se você olhar no arquivo CSS, verá que já criamos uma regra para essa classe.

No momento, contém a declaração `height: 100vh;`. Isso garante que a altura da página web sempre será igual à altura da tela do usuário, independentemente do tamanho da tela.

Agora é hora de você começar a programar. Comece adicionando uma imagem de plano de fundo ao cabeçalho. Aqui está o que você precisa fazer:

- Adicione uma imagem de sua escolha à pasta `images/`.
- Em seguida, use a propriedade `background-image` para especificar um caminho para esta. Para especificar o caminho para sua imagem, o valor da propriedade

background-image deve ser semelhante a: `url(../images/name-of-file.jpg)`.

- Use as propriedades `background-size` e `background-position` para definir o tamanho e a posição da imagem.
- Abra o arquivo `index.html` em seu navegador para ter certeza de que está bom.

Você deve ter notado que o arquivo CSS contém uma regra para o seletor `.overlay`. Isso sobrepõe uma camada semitransparente na parte superior da imagem de plano de fundo usando a propriedade `background-color`.

(2)

A seguir, trabalharemos no título da página principal. Abra o arquivo `index.html` e procure o comentário que diz `<!-- add the page title here -->`. Substitua esse comentário por um elemento `<h1>`. Dê a ele um nome de classe descritivo e escreva seu nome entre as marcas de abertura e de fechamento. Atualize a página no navegador. Seu nome deve aparecer no canto superior esquerdo da página.

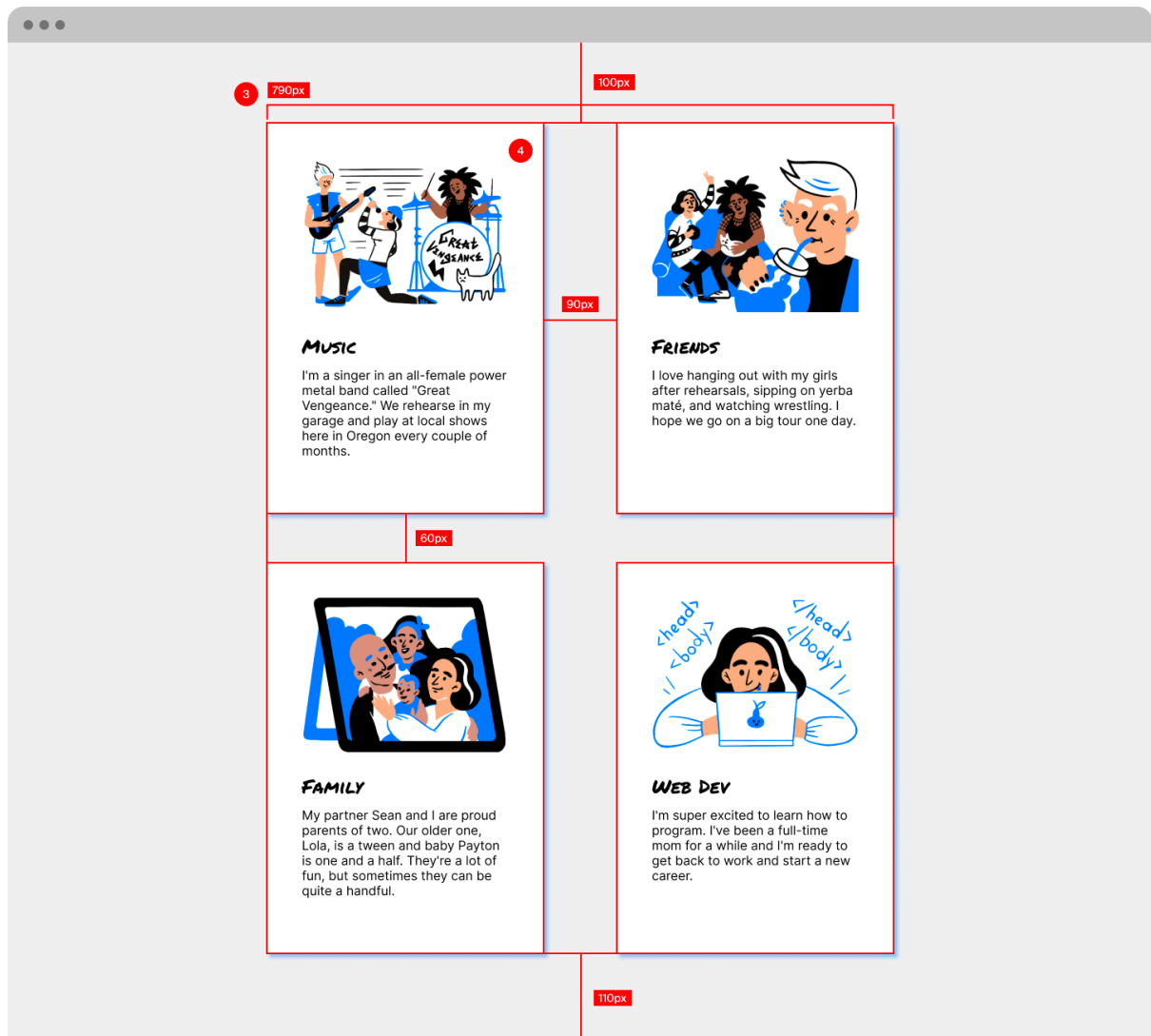
Em seguida, você precisa adicionar alguns estilos ao título. Crie uma nova regra CSS no arquivo `index.css` usando o nome da classe que você criou como seletor. Adicione as seguintes declarações à regra:

- `font-family: "Permanent Marker", cursive;`
- `font-size: 140px;`
- `line-height: 140px;`
- `color: white;`
- `text-transform: uppercase;`
- `max-width: 1240px;` — para limitar a largura
- `margin: auto;` — isso centralizará o título da página horizontal e verticalmente. Isto funciona para centralização vertical neste caso porque seu elemento pai com a classe `overlay` tem a declaração mágica `display: flex;` aplicada a ele.

Para garantir que o texto do título esteja sempre alinhado no centro da página, adicione a declaração `text-align: center;` à regra CSS para o cabeçalho.

Precisamos disso porque enquanto a declaração `margin: auto;` aplicada ao título garante que o elemento `<h1>` em si seja centralizado, seu conteúdo, ou seja, o texto, é alinhado à esquerda das bordas do elemento por padrão.

Cartões



(3)

Após o cabeçalho, é hora de passar para a seção de conteúdo. É aqui que você construirá os cartões que mostram fatos interessantes sobre você. Existem quatro cartões que devem ser centralizados horizontalmente e agrupados em duas linhas. Para que isso aconteça, já adicionamos três declarações CSS à regra com o seletor

`.content`:

`display: flex;` — você já está familiarizado com este. O valor `flex` nos permite centralizar nossos elementos. Ele também nos fornece uma ampla gama de opções para organizar nossos elementos. Você aprenderá mais sobre isso no programa completo, mas usaremos duas opções para este projeto:

- `flex-wrap: wrap;` — quando esta declaração é usada, os itens dentro de um contêiner serão agrupados em uma nova linha se não houver espaço suficiente para todos eles com suas dimensões especificadas. Caso contrário, por padrão, o valor `flex` faria com que os elementos se comprimissem em uma linha.
- `justify-content: space-between;` — esta declaração distribui uniformemente os itens dentro de seu contêiner. Em nosso caso, o contêiner para os cartões será o elemento `<section>` com a classe `content`.

Antes de criar os cartões, adicione as seguintes declarações à regra CSS com o seletor `.content` para definir as dimensões do contêiner:

- `max-width: 790px;`
- `margin: 100px auto 50px;`

(4)

Vamos voltar ao arquivo HTML. Observe que cada cartão no design tem a mesma estrutura. Ele contém uma imagem, título e texto. Dentro do elemento `<section>`, crie um elemento `<div>` com a classe `card`. Este será um elemento pai para o conteúdo do cartão. Dentro do elemento `<div>`, crie um elemento `` com os seguintes atributos:

- `class` para adicionar um nome do seletor correspondente. Lembre-se de que deve ser descritivo (por exemplo, `card__image`)
- `src` para especificar um caminho para a imagem
- `alt` para adicionar um texto alternativo para a imagem

Abaixo do elemento ``, adicione um elemento `<h2>` para o título do cartão e um elemento `<p>` para sua descrição. Atribua nomes de classe a eles também.

Agora você tem a estrutura básica de um cartão. Você precisa de quatro cartões, então copie o elemento `<div>` e cole-o mais três vezes abaixo.

Agora você pode começar a adicionar algum conteúdo aos cartões. Adicione as imagens que deseja à pasta `images/` e, a seguir, vincule a elas por meio do atributo `src` de cada elemento ``. Preencha os elementos de texto com informações sobre você.

A seguir, vamos definir o estilo de nossos cartões. Adicione uma regra CSS com o seletor `.card` com as seguintes declarações:

- `width: 350px;`

- `margin-bottom: 60px;`
- `padding: 45px 45px 70px;`
- `background: #ffffff;`
- `box-shadow: 4px 4px 5px rgba(0, 132, 255, 0.4);` — para adicionar um efeito de sombra

Adicione uma regra CSS para o título do cartão (ou seja, o elemento `<h2>`). Não se esqueça de especificar estilos para as fontes e a margem inferior:

- `font-family: "Permanent Marker", cursive;`
- `font-size: 24px;`
- `margin-bottom: 10px;`

A regra CSS para a descrição do cartão deve conter apenas a propriedade `font-size` com um valor de `16px`.

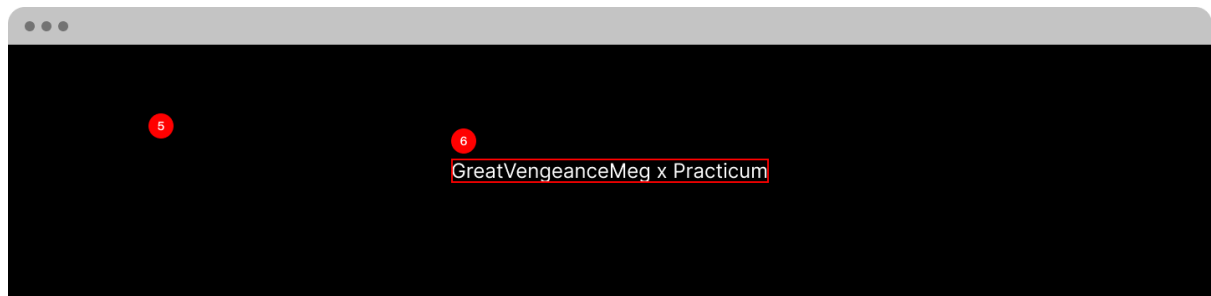
Por fim, adicione uma regra para a imagem do cartão com as seguintes declarações:

- `width: 100%;`
- `margin-bottom: 25px;`

Se suas imagens têm tamanhos diferentes e você percebe que alguns dos cartões têm alturas diferentes, você pode lidar com isso atribuindo as seguintes declarações à regra de classe `.card`:

- `height: 200px;` — para adicionar uma altura fixa
- `object-fit: contain;` — esta declaração faz com que o elemento caiba dentro da caixa de conteúdo fixo, enquanto preserva sua proporção

Rodapé



(5)

Vamos passar para o rodapé, ou footer. Uma declaração `display: flex;` já foi adicionada à regra CSS correspondente para centralizar o conteúdo do footer. Defina a altura e a cor de fundo de acordo com o design:

- `min-height: 300px;`
- `background-color: #000000;`

(6)

Adicione sua assinatura dentro do rodapé usando a tag `<p>` e estilize-a definindo o tamanho da fonte para `24px`, a cor para `#ffffff`, e centralize-a com a declaração `margin: auto;`.