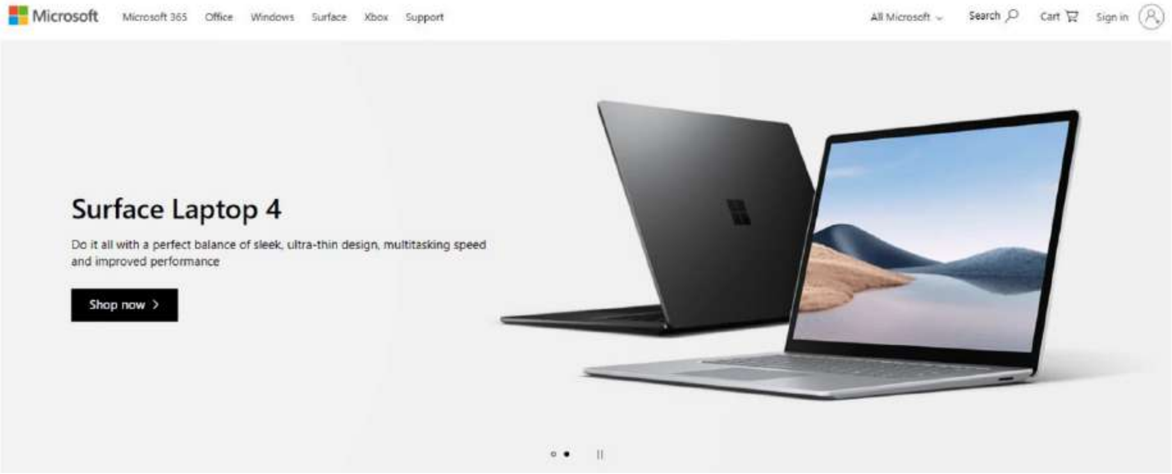
Project 2: Microsoft Clone (Responsive) in Tailwind CSS: Tailwind Tutorial #15

In this article, you will get to know how to create a Microsoft Clone with the help of Tailwind CSS. You can check out the previous Facebook clone project by clicking here. Unlike the previous one, This clone of Microsoft will be fully responsive and would require a deep understanding of Tailwind CSS. Thus, To know more about the workflow and the fundamentals of Tailwind, You can check out the Tailwind CSS course by clicking here. Let's get started-

Microsoft: Clone Reference

After performing the basic Setup of Tailwind CSS, You need to collect the reference image to help you out throughout the clone creation process. As per the title, The below sample of Microsoft Homepage will be in our consideration throughout the tutorial-



Surface Laptop 4

Do it all with a perfect balance of sleek, ultra-thin design, multitasking speed and improved performance

Figure 1.1: Reference: Microsoft Homepage

Mobile-First Approach

This project of Microsoft Homepage is going to be responsive, which means that the application can change the interface according to different devices.
As discussed earlier, while creating a responsive application with the help of Tailwind CSS, it is always preferred to begin designing it with the perspective of smaller devices. After that, One can use Tailwind CSS breakpoints to interchange the interface for different devices. Thus, You can follow the recommended method of initially creating the website for the phone and then moving to different dimensions with the help of breakpoints. Let's start creating each component of the website as discussed below-

Creating Navbar - Microsoft

The navbar of the original application has certain items which are displayed in an appropriate fashion on a mobile device. Firstly, One can create a flexbox navbar container and then begin with the designing and creation of other components of the Navbar as mentioned below-

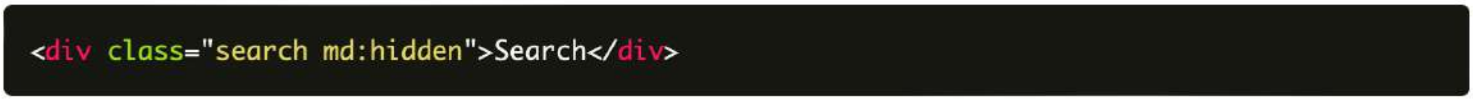
1. Creating Hamburger Menu

It is an icon that consists of three parallel horizontal lines. This button is generally placed at the top corner of the application and helps to preserve screen space by hiding navigational elements. Here's how one can create a hamburger button-

```
<div class="“navbar" flex justify-between items-center p-4”>
  <div class="“flex" justify-center items-center”>
    <div class="“" hamburger inline-block p-4 cursor-pointer”>
      <div class="“line" h-0.5 w-6 my-1 bg-black”></div>
      <div class="“line" h-0.5 w-6 my-1 bg-black”></div>
      <div class="“line" h-0.5 w-6 my-1 bg-black”></div>
    </div>
  </div>
</div>
```

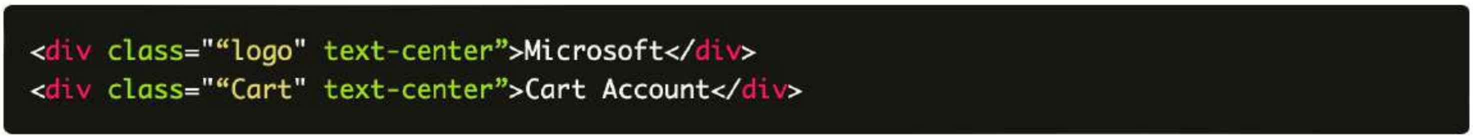
Explanation: Inside the navbar flex container, a new div Hamburger container with some interface enhancing Tailwind CSS properties have been created. This container contains three child containers having a black line of appropriate width, height and margin. Moreover, The entire Hamburger has been accommodated inside a dummy div flexbox to declare it as one element and not as a collection of three. Please note that the navigational elements will be added to the Hamburger menu in the later part of the tutorial.

2. Adding Search Icon: It’s really simple to add a search icon beside the Hamburger menu. To do so, One has to simply write the code of the search icon inside the dummy div container, which accommodates the hamburger. For ease, You can use the below piece of code to add the search icon in your application. As we are creating this search icon for the smaller devices, therefore we are using the hidden utility class to hide it at the md breakpoint-



```
<div class="search md:hidden">Search</div>
```

3. Adding Microsoft Logo, Cart, and Account: In the reference, The Microsoft logo is at the center and the cart icon is at the right corner of the navbar respectively. You can do something similar by using the below piece of code inside the navbar container. Please note that the image, utility classes, and responsiveness modifications will be added in the next section of the article.



```
<div class="“logo" text-center”>Microsoft</div>
<div class="“Cart" text-center”>Cart Account</div>
```

Result: Here’s the look of the format of Microsoft cloned Navbar at lower screen dimensions. In the below case, The text has been added in place of an icon to provide a better understanding of the concept.

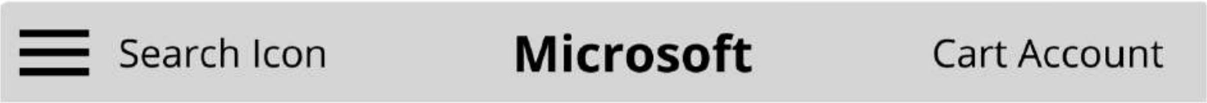
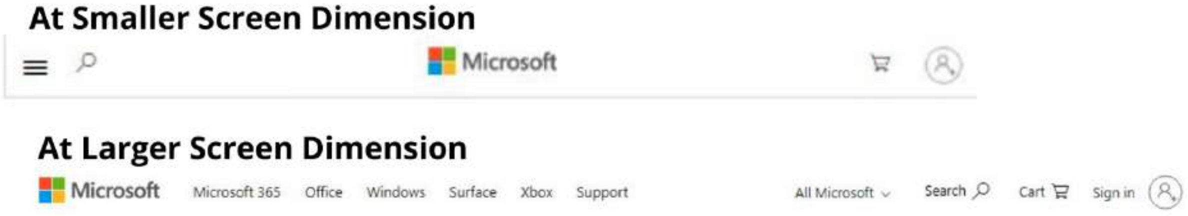


Figure 1.2 Microsoft Navbar Format

Using Tailwind Breakpoints- Responsiveness

As per the reference, On increasing the dimension of the Original application a sudden change in the design of the navbar occurs at a specific instance. Eventually, Tailwind CSS provides you with the Tailwind breakpoints which can be used to change the design after specific dimensions are achieved. Here's the look of both the original navbars at different dimensions-



At Smaller Screen Dimension

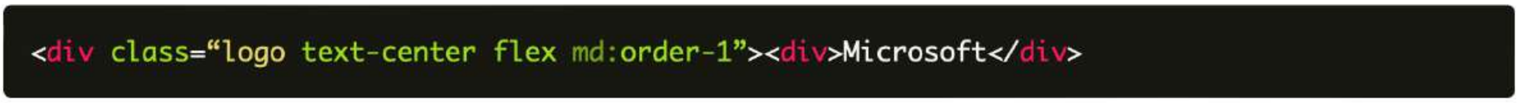
At Larger Screen Dimension

Figure 1.3: Navbar at Different Breakpoints

Explanation: As you can see, The Microsoft logo has moved from the center to the extreme left corner of the page. Moreover, The hamburger has disappeared after attaining larger dimensions. As a result, All the navigational elements are spread apart. Furthermore, The search icon has been moved to the right corner, just beside the cart icon. Let's make the above changes in the cloned Navbar at the 'md' breakpoint as discussed below-

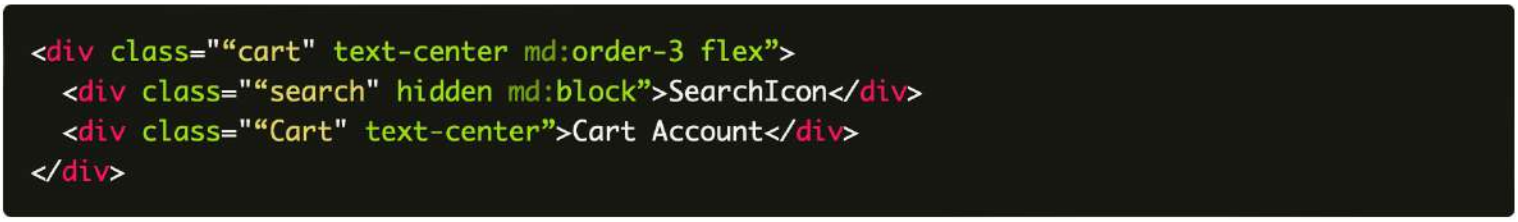
1. Hiding Hamburger: One can simply add the 'md:hidden' class of Tailwind in the hamburger container to make it disappear at the 'md' breakpoint.

2. 'Microsoft Logo' on the Left: After hiding the Hamburger, You have to shift the Microsoft logo to the left corner of the navbar at the 'md' breakpoint. To do so, You can simply use the 'Order' utility class of Tailwind in the logo container as shown below-



```
<div class="logo text-center flex md:order-1"><div>Microsoft</div>
```

3. 'Search Icon' at the Right: Currently, On hitting the MD breakpoint the existing search icon will acquire the center position due to the addition of md:order-2 class in the flex container. But, the search icon needs to be in the right corner of the navbar at the MD breakpoint. Therefore, you can hide the existing search icon at this breakpoint by adding the "md:hidden" class and then creating an alternative default hidden search icon that becomes visible on the right corner of the page at the 'MD' breakpoint as shown below-



```
<div class="“cart" text-center md:order-3 flex”>
  <div class="“search" hidden md:block”>SearchIcon</div>
  <div class="“Cart" text-center”>Cart Account</div>
</div>
```

Navigational Elements

In the original navbar, the navigational elements of the page are embedded in the navbar to provide a better user interface. Moreover, In small devices such as phones, the hamburger contains the list of navigational elements and at certain screen sizes, the elements get sequentially arranged in the Navbar.

Creating Navigational Elements

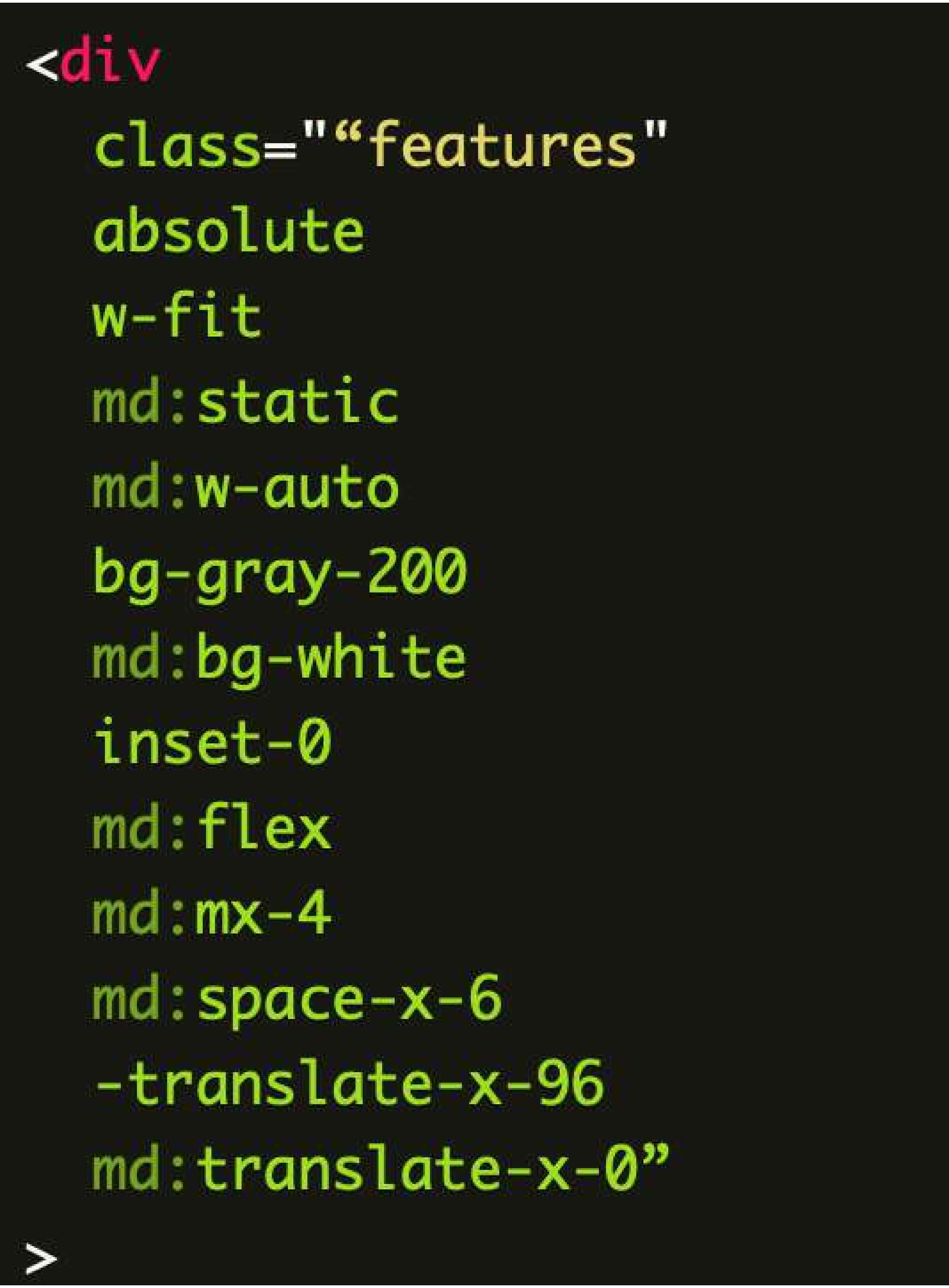
Perhaps, You can create a <div> ‘features’ box containing all the navigational elements and then add them to different locations depending on the screen dimension. You can use the below piece of code for your ease. Here, we have added the feature container inside the Microsoft Logo container as shown below-

```
<div class="“logo" text-center flex md:order-1”>
  <div>Microsoft</div>
  <div class="“features" absolute md:flex mx-4 space-x-2”>
    <div class="“fitem”">Microsoft 365</div>
    <div class="“fitem”">Office</div>
    <div class="“fitem”">Windows</div>
    <div class="“fitem”">Surface</div>
    <div class="“fitem”">Xbox</div>
    <div class="“fitem”">Support</div>
  </div>
</div>
```

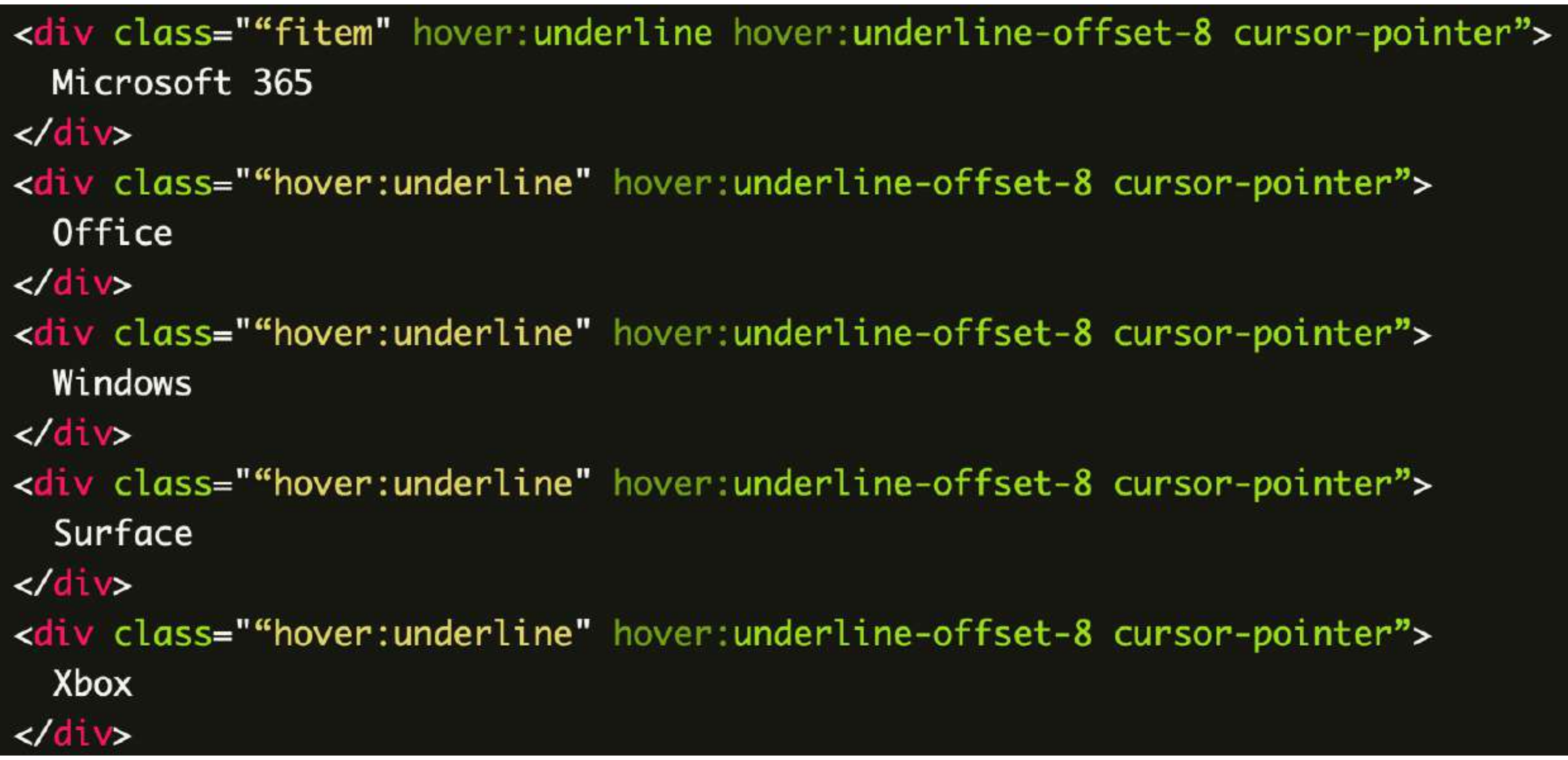
Hence, the features flexbox will be successfully aligned in the horizontal row with appropriate spacing between the elements due to the addition of the space-x class. Now, let's use the Tailwind Breakpoints to provide responsiveness to these elements as discussed in the next section.

Responsiveness- Navigational Elements

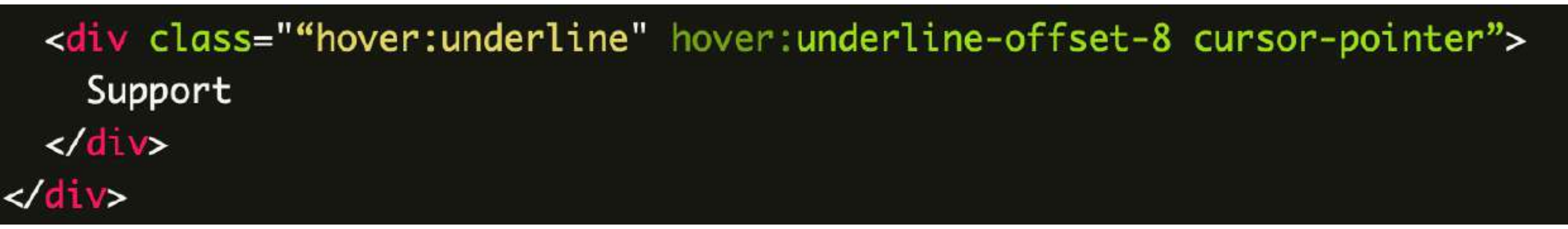
Perhaps, The above Navigational features aren't responsive, as a result, the navbar interface is ruined on small devices. Thus, Let's start fixing the features container by adding the elements in the Hamburger menu of small devices as shown below-



```
<div
  class="“features"
  absolute
  w-fit
  md:static
  md:w-auto
  bg-gray-200
  md:bg-white
  inset-0
  md:flex
  md:mx-4
  md:space-x-6
  -translate-x-96
  md:translate-x-0”
>
```



```
<div class="“fitem" hover:underline hover:underline-offset-8 cursor-pointer”>
  Microsoft 365
</div>
<div class="“hover:underline" hover:underline-offset-8 cursor-pointer”>
  Office
</div>
<div class="“hover:underline" hover:underline-offset-8 cursor-pointer”>
  Windows
</div>
<div class="“hover:underline" hover:underline-offset-8 cursor-pointer”>
  Surface
</div>
<div class="“hover:underline" hover:underline-offset-8 cursor-pointer”>
  Xbox
</div>
```



```
  <div class="“hover:underline" hover:underline-offset-8 cursor-pointer”>
    Support
  </div>
</div>
```

Explanation:

1. Assigning Position: Here, we have used the ‘absolute’ utility class to position the feature element outside the normal flow of the page, causing the neighboring elements to act as if the feature container doesn’t exist in the workflow. Moreover, At the ‘MD’ breakpoint, we have used the ‘static’ utility class to position the element according to the decorum of the document. Furthermore, One can use the inset, translate, space-x and other classes to provide an appropriate alignment to the container.

2. Setting Width: According to the need, You can set the width of the container by using any of the tailwind utility classes. For example, You can use the ‘w-fit’ class as the default width of the container as shown in the above example.

3. Adding Background: Initially, at the smaller screen dimension, the navigational elements are compromised in the Gray background color Hamburger menu. At the MD breakpoint, the background color needs to be turned ‘white’ as the elements will be aligning in the Navigation bar.

4. Enhancing Flex Items: One can improve the user interaction with the components by adding some of the utility classes such as ‘hover: underline’ and ‘cursor pointer’.

Eventually, One can use Javascript language to show the navigational elements after the Hamburger menu is clicked.

Finishing Touch to Navbar

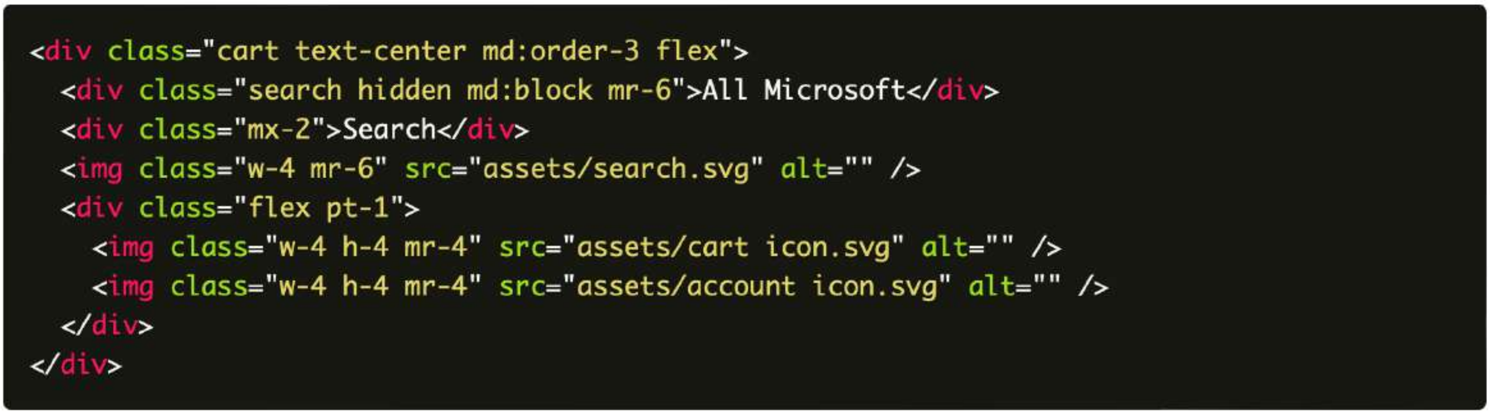
1. Adding Microsoft Logo: Finally, You can add the Microsoft logo png image by using the src tag. Moreover, You can assign a proper width to the image as shown below-

```
<div class="logo text-center flex md:order-1">
<div class="flex justify-start"><img class="w-[50%]" src="assets/mslogo.png" alt=""></div>
```

2. Enhancing Search Icon(Smaller device): This is the search icon that was created to be visible in the smaller screen dimension. As mentioned before, You can use the src tag to add the search icon logo. Moreover, You can assign the width and margin to the icon as shown below-

```
<div class="search md:hidden w-4 mr-6">Search</div>
```

3. Enhancing Search Icon(Larger Device) and Cart Icon: Similarly, You can assign a proper position and image to the search icon at md breakpoint as shown below. Furthermore, You can search for the ‘carts and account icon’ svg and add it with proper height, margin and width in your application as shown below-



```html
<div class="cart text-center md:order-3 flex">
  <div class="search hidden md:block mr-6">All Microsoft</div>
  <div class="mx-2">Search</div>
  <img class="w-4 mr-6" src="assets/search.svg" alt="" />
  <div class="flex pt-1">
    <img class="w-4 h-4 mr-4" src="assets/cart icon.svg" alt="" />
    <img class="w-4 h-4 mr-4" src="assets/account icon.svg" alt="" />
  </div>
</div>
```

Creating Slider Component

The next step is to create the carousel slider just below the navbar to enhance the appearance of the page. As per the observation of the reference image, the big flex container comprises two elements. The first one is a text box containing the tagline and a ‘shop now’ button. The latter component is an image of appropriate width and height. These two elements interchange the positions on reducing the screen size. You can start replicating the original Microsoft Carousel as discussed below-

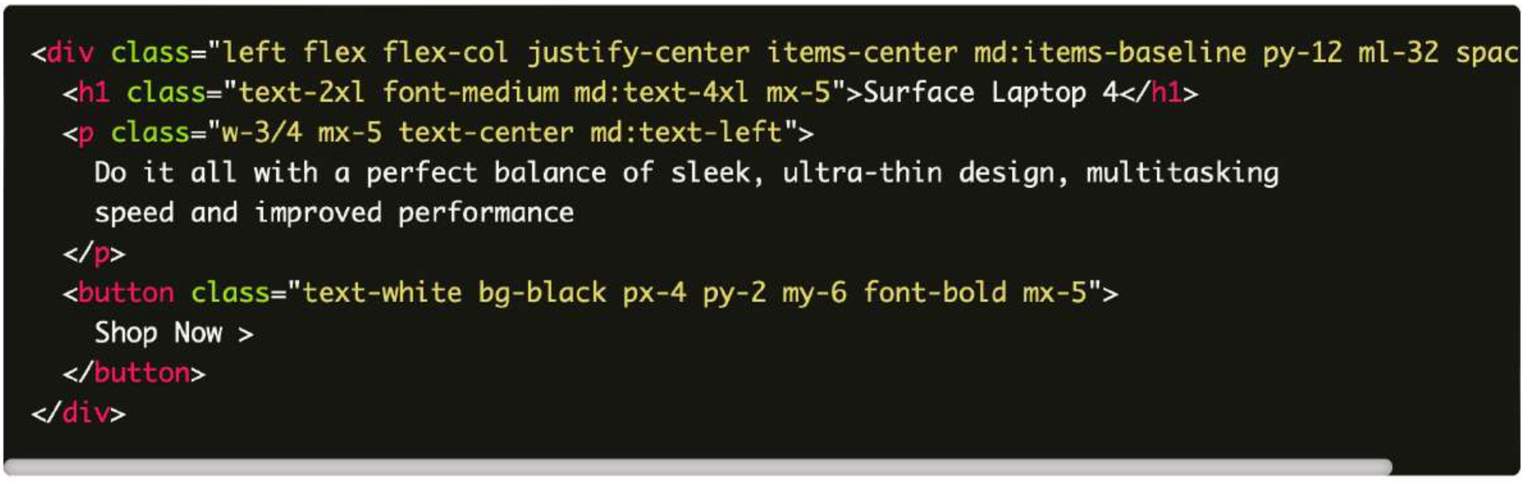
Structuring the Slider Component

First of all, we would download all the required icons from the official site to add to the application. Let’s begin structuring the carousel slider component of the clone as follows-

1. Creating the ‘Slider’ Container: Eventually, You have to create a parent flexbox container and add some enhancing properties like background color, column reverse, and flex row at particular instances as shown below-

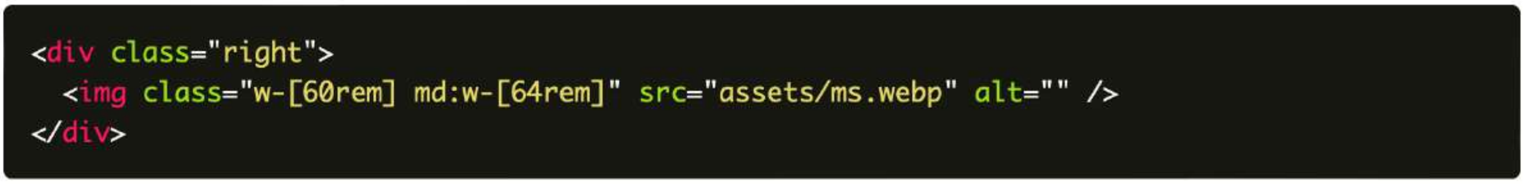
```
<div class=“slider flex-col-reverse md:flex-row bg-[#f2f2f2] ”> </div>
```

2. Creating the 'Left' Container: Inside, the parent container, You can add the left flexbox component. Eventually, You can add the heading and the tagline with the help of the <h1> and <p> tags respectively. Furthermore, you need to add a 'Shop Now' button with white text, black background, appropriate padding, margin, and bold text. Here's the code of the left flexbox container with suitable Tailwind utility classes.



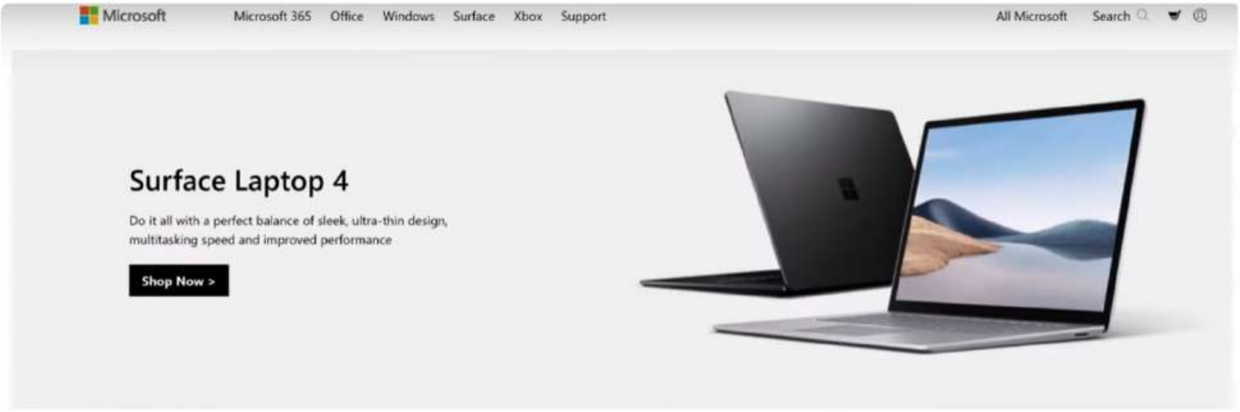
```
<div class="left flex flex-col justify-center items-center md:items-baseline py-12 ml-32 spac
  <h1 class="text-2xl font-medium md:text-4xl mx-5">Surface Laptop 4</h1>
  <p class="w-3/4 mx-5 text-center md:text-left">
    Do it all with a perfect balance of sleek, ultra-thin design, multitasking
    speed and improved performance
  </p>
  <button class="text-white bg-black px-4 py-2 my-6 font-bold mx-5">
    Shop Now >
  </button>
</div>
```

3. Creating the 'Right' Container: As discussed earlier, the right box will contain an image of suitable width for the application. Moreover, You have to use the md breakpoint to set a different width of the image for a larger application. Here's the code for adding the right container into your application-



```
<div class="right">
  <img class="w-[60rem] md:w-[64rem]" src="assets/ms.webp" alt="" />
</div>
```

Result: You can have a look at the amount of progress made till this instance of creating the clone-



Surface Laptop 4

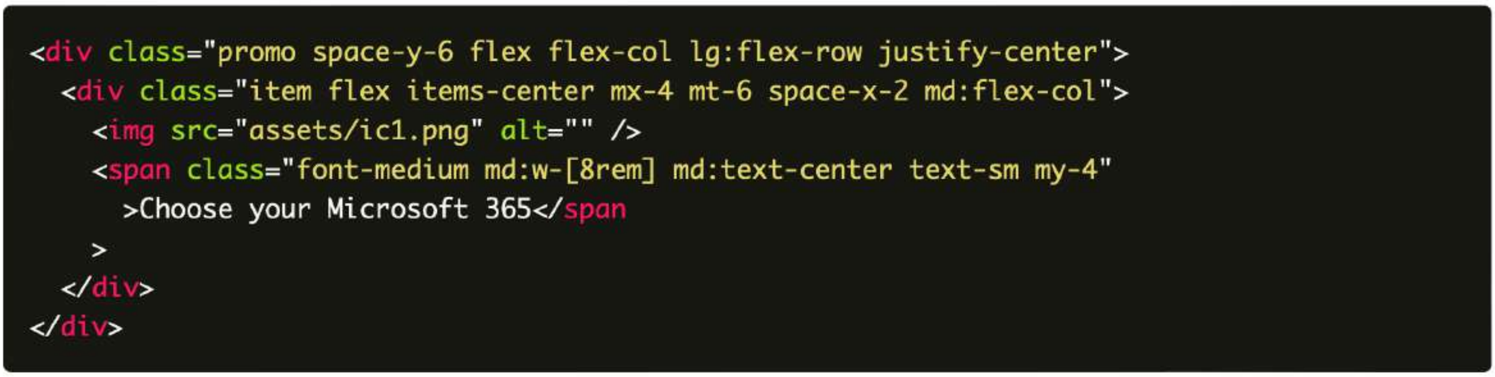
Do it all with a perfect balance of sleek, ultra-thin design, multitasking speed and improved performance

Figure 1.4: Created the Slider

Creating the 'Promo' Container

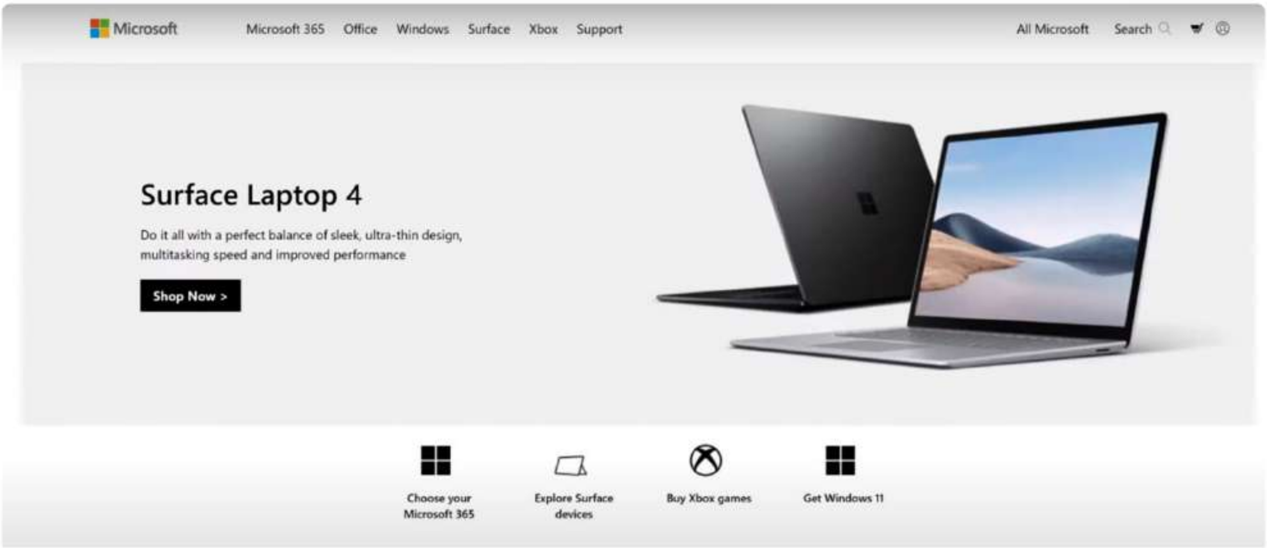
Now, The next step is to add the 'promo' container just beneath the carousel of the application. As per the reference, This component will show 2-4 featured items of the website with proper icons and a short tagline. Let's begin adding the children items in this third component as discussed below-

Creating the Items: This item will contain the logo image of Microsoft and span tagline text. One can enhance the image by adding suitable margins from the top and x-direction with some more utility classes. Moreover, You can use the font medium, text-sm and my-4 class to provide proper font size and margin to the span text respectively. Furthermore, You can use the MD breakpoint to assign appropriate utility classes to both components for larger screen dimensions. Here's the code of the first Item of the Promo-



```
<div class="promo space-y-6 flex flex-col lg:flex-row justify-center">
  <div class="item flex items-center mx-4 mt-6 space-x-2 md:flex-col">
    <img src="assets/ic1.png" alt="" />
    <span class="font-medium md:w-[8rem] md:text-center text-sm my-4"
      >Choose your Microsoft 365</span
    >
  </div>
</div>
```

Similarly, You can create and add other items in the promo section of the page. Here's the look of the application after the addition of four such suitable items. Hence, we have successfully cloned the Microsoft application to some extent.



Surface Laptop 4

Do it all with a perfect balance of sleek, ultra-thin design, multitasking speed and improved performance

Figure 1.5: Microsoft Clone- Result

I hope you get the basic idea of how to clone a website with the help of Tailwind CSS. In this article, we have cloned the upper portion of the Microsoft page. However, You can complete the rest of the application with your understanding of Tailwind CSS.

Thank you for being with me throughout the tutorial. See you all in the next tutorial. Till then Keep learning and Keep Coding.