Project3- Polaroid Image Gallery

Html

1. Build an app similar to this app: <https://codeandcoffee.000webhostapp.com/photoAlbum/>

2. Follow the below steps and pass all the tests. Give it your personal style.

3.It is recommended to use only HTML and CSS for this project. It will be best if you use plain CSS for this project. Do not use any of the JavaScript frameworks for this project as you will get chance to work on them later in this course. So let’s jump in!!.

4.The image gallery should start with a title in side the head tags with text of “Image Gallery”. Note this title will be visible in the address bar of the current tab.

5.Next in the body section i should see a h1 element that contains text like “Polaroid photo album”. It should be wrapped in between the header tags.

6.I should see a div element with a class of “wrapper” in it, which will contain all the images inside it.

7.If you don’t know what polaroid photo means, it means an image with some text below it. Inside the wrapper class take another div element with a class of “item”. Then i should see a class inside another div with “polaroid”. To put the image use the img tag with src= “<https://farm3.static.flickr.com/2023/2313208499_94cf139fed_m.jpg>”.

8.Next to give the image a caption use a class “caption” and wrap it around the text.

9.Repeat the above two steps 4 more times in a similar manner with any caption of your choice. Remember the caption should be related to the image you are putting.

Four other images are : <https://s3-us-west-2.amazonaws.com/s.cdpn.io/210284/LZkivxR.jpg>

<https://s3-us-west-2.amazonaws.com/s.cdpn.io/210284/hqcMtrF.jpg>

<https://s3-us-west-2.amazonaws.com/s.cdpn.io/210284/l867sBU.jpg>

<https://s3-us-west-2.amazonaws.com/s.cdpn.io/210284/7cQCk5I.jpg>

After this your html is page is all done.

CSS(Cascading style sheet)

1. First of all i should that the css file is linked with the html through the link tag which is placed in between the head section like this :<link rel="stylesheet" type="text/css" href="style.css">
2. At the top i should see an universal selector (\*) and give it a property of box-sizing with a value of “border-box”.
3. Next i should see the body selector with properties of “font-family” set to ‘Helvetica Neue’. Give it a background-image property with a value of url(bricks.jpg); and remember to keep the background-size property to 100%.
4. Target the header element selector and keep its border-style property to “inset”. Give it a border-width property of 5px and keep its height to “auto”.
5. Next to keep the h1 element selector in the center use the text-align property and set it to “center”. Give it a padding of 20px on top, 0px right, 50px bottom and 0px on left. To make the text look larger give it a font-size of 40px. Keep the letter-spacing property to 2px.
6. Next target the img selector and keep its max-height to 100%. Set the height property to auto as the element will automatically adjust its height to allow its content to be displayed correctly.
7. Now we will target the class selectors.
8. Target the .wrapper class selector and give it a width of 100% because “width: 100%” by default the img element displays an image in its original dimensions. By giving it width of 100% we said it to be as wide as it’s parent, not more than that. And use the text-align property to center the elements of wrapper class.
9. Next we will target the .polaroid class and give it a padding property and set it to “1 rem”. And remember to apple the box-shadow property and give it a value of “0 0.25rem 1rem rgba(0,0,0,0.2);”.
10. When you added the 2nd image, you must have seen that the next image is placed below the first, we want it to be beside it, not below it. To do that we will use the display: inline-block; which will place the image side by side. We will use the margin-top property and set it to 2rem. Next we will use the filter property. The **filter** [CSS](https://developer.mozilla.org/en-US/docs/Web/CSS) property applies graphical effects like blur or color shift to an element. Filters are commonly used to adjust the rendering of images, backgrounds, and borders. We will set it to filter: grayscale(100%);. Remember all this properties will be applied to the .item class.
11. Next we will target the .polaroid class inside the .item class and use a pseudo selector :before with it like this: .item .polaroid:before{}. We will use the position property and set its position to absolute. An element with position: absolute; is positioned relative to the nearest positioned ancestor. We will also use the z-index property and set it to -1. The z-index property specifies the stack order of an element. We will add the transition property which allows us to change the property very smoothly over a given duration. Syntax would be transition: all 0.35s;
12. Now you have to use the :nth-of-type(*an + b*) selector which matches every element that is the *n*th child, of a particular type, of its parent. We will use this selector with the .item class. *a* represents a cycle size, *n* is a counter (starts at 0), and *b* is an offset value. Syntax would be .item:nth-of-type(4n+1) {}. We will use the transform property with values scale(0.8, 0.8) and rotate(5deg). The rotate() method rotates an element clockwise or counter-clockwise according to a given degree and the scale() method increases or decreases the size of an element (according to the parameters given for the width and height). And the transition property we used before with same values.
13. Next we will target the .item:nth-of-type(4n+1) .polaroid:before {}. The class .polaroid:before means, The ::before selector inserts something before the content of each selected element(s). The property used in this are transform property set to rotate(-6deg), height set to 20%, width: 47%, bottom: 30px, left: 12px,and the box-shadow property to box-shadow: 0 2.1rem 2rem rgba(0,0,0,0.4);
14. In the next 3 steps we will follow the above 2 process with little changes like (4n + 2), (4n + 3) and (4n+4). In the .item:nth-of-type(4n+2){} the transform property would be having the value transform: scale(0.8, 0.8) rotate(-5deg) and the same transition value as in step 12.
15. Then we will target the .item:nth-of-type(4n+2) .polaroid:before{}. It will have all the same property values except the transform property would have the value set to “rotate(-6deg)” and left property set to 12px. For the next two steps try them on their own for (4n +3) and (4n+4).
16. Next we will use the pseudo class of hover on the .item class. Syntax would be .item:hover{} . The :hover selector is used to select elements when you mouse over them. We will keep the filter property set to “none” in this case and the transform property will be set to transform: scale(1, 1) rotate(0deg) !important. When an important rule is used on a style declaration, this declaration overrides any other declarations. And the transition property is set to transition: all 0.35s.
17. In the last step we will use the class .item:hover .polaroid:before{}. We will use the absolute position property and set the z-index to -1. Set the transform property to “rotate(0deg)”. The height and width should be set to 90%. The box-shadow is set to box-shadow: 0 1rem 3rem rgba(0,0,0,0.2) and at last set its transition property to transition: all 0.35s.
18. Once you are done with the project , submit the url to your project on the following link.

19. If you are stuck at any part of the project remember to ask me on this link.

20. The image used in the project is for reference use feel free to use any poster image of your choice.