Name: Nalinee Singh
Div: F
Roll no: 35

Practical –2

Task: Develop a webpage with a grid layout to display a collection of images
Requirements:
1. Apply CSS to style the gallery, including image borders, margins, and spacing
2. Implement hover effects to highlight images on mouse over.
3 Make the gallery responsive and ensure it looks good on both desktop and mobile devices
Aim:
To create a visually appealing photo gallery webpage using HTML and CSS, implementing a grid layout that includes styled images, hover effects, and responsive design.
Objective:
1. Develop a grid layout to display images in an organized manner.
2. Apply CSS styles for image borders, margins, and spacing.
3. Implement hover effects to enhance user interaction.
4. Ensure the gallery is responsive for optimal viewing on both desktop and mobile devices.
Software:
● Visual Studio Code (VSCode): A code editor used for writing HTML and CSS.
● Web Browser: For testing and viewing the webpage.

Theory:

A photo gallery is a collection of images presented in a structured format. The grid layout is commonly used for galleries, allowing multiple images to be displayed in rows and columns. CSS plays a critical role in styling the gallery, including defining the appearance of images, their spacing, and interactivity through hover effects. Responsive design ensures that the gallery adjusts to different screen sizes, enhancing usability on mobile devices.

1. Develop a Grid Layout to Display Images
A grid layout is an effective way to organize content in a structured manner, especially for visual elements like images. CSS Grid Layout allows developers to create complex responsive layouts using rows and columns, making it ideal for galleries.

Key Concepts:

● Grid Container: The parent element that contains grid items, serving as the structure for the layout.
● Grid Items: The child elements (in this case, images) within the grid container.
● Grid-template-rows
The grid-template-rows property specifies the heights of the rows in a grid. Here are a few examples:
Equal Row Heights:

grid-template-rows: auto auto;
Fixed Heights:
grid-template-rows: 50px 100px;

Flexible Row Heights:
grid-template-rows: repeat(3, 1fr);
This creates three rows of equal height, each taking up an equal fraction of the available space.
● Grid Template Columns: This defines the number of columns and their width. For example, using grid-template-columns: repeat(auto-fill, minmax(200px, 1fr)); allows the grid to adjust the number of columns based on the available width, creating a flexible layout.
Examples:
1. grid-template-columns: auto auto;
2. grid-template-columns: auto auto auto;
3. grid-template-columns: 50px 100px;
● Gap: The space between grid items, which visually separates images and enhances the overall appearance.

2. Apply CSS Styles for Image Borders, Margins, and Spacing
CSS provides a wide range of properties to style images effectively, enhancing both functionality and aesthetics.
Key Concepts:
● Borders: Adding borders to images can create a framed effect. Using properties like border and border-radius can soften the edges and improve visual appeal.
For example, border: 5px solid #ddd; gives a subtle outline to the images.
● Margins: Margins are the space outside
an element. They help to separate images from other content and each other. By adjusting margin values, we can create uniform spacing around images, making the layout look clean and organized.
● Padding: While padding applies space within the border of an image, it is more commonly used for container elements.
By applying these styles, we enhance the user experience by ensuring that images are visually
appealing and clearly distinguished from one another and the surrounding content.

3. Implement Hover Effects to Enhance User Interaction
Hover effects are a crucial aspect of web design, providing visual feedback to users and enhancing interactivity.
Key Concepts:
● Transition: The transition property allows for smooth changes between different states of an element. For example, transition: transform 0.3s, opacity 0.3s; enables images to scale and fade smoothly when hovered over, creating a visually appealing effect.
● Transform: This property alters the scale or position of an element. By using transform: scale(1.05);, the image slightly enlarges on hover, drawing attention and creating a sense of interaction.
● Opacity: Adjusting the opacity can create a layered effect, adding visual depth. For

instance, an opacity change on hover, like opacity: 0.9;, signals to users that the image is being interacted with.

4. Responsiveness in Web Design
Responsive design is a crucial aspect of modern web development, ensuring that websites look and function well across a wide range of devices, from large desktop monitors to small mobile screens. A responsive website adjusts layout, images, and interactions to accommodate different screen sizes.
To create a responsive website, add the following <meta> tag to all your web pages:
<meta name="viewport" content="width=device-width,
initial-scale=1.0">

3. If the CSS width property is set to 100%, the image will be responsive and scale up and down:
<img src="img_girl.jpg" style="max-width:100%;height:auto;">

CSS STYLE
```
body {
font-family: Arial, sans-serif;
background-color: #f7f7f7;
padding: 20px;
}
```

4.
```
/* Grid Container */
.gallery {
display: grid;
grid-template-columns: repeat(auto-fill, minmax(200px, 1fr)); /* Flexible column layout */
gap: 15px; /* Gap between grid items */
margin-top: 20px;
}
/* Grid Items (images) */
.gallery img {
width: 100%;
height: auto;
border-radius: 8px;
box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);
transition: transform 0.3s ease, opacity 0.3s ease;
}
/* Hover Effects */
.gallery img:hover {
transform: scale(1.05); /* Slight zoom on hover */
opacity: 0.9; /* Reduce opacity for interaction feedback */
}
}
```

SOURCE CODE:

HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0" />
   <title>Photogallery</title>
   <link rel="stylesheet" href="style.css">
</head>
<body>
   <h1>Photo Gallery</h1>
   <div class="pg1">
       <a href="/blackhole.jpg">
           <img src="/blackhole.jpg" alt="">
       </a>
       <div class="desc">Blackhole image</div>
   </div>
   <div class="pg1">
       <a href="/Milky_Way_Galaxy.jpg">
           <img src="/Milky_Way_Galaxy.jpg" alt="">
       </a>
       <div class="desc">Milkyway Galaxy image</div>
   </div>
   <div class="pg1">
   <a href="/neutron_star.jpg">
       <img src="/neutron_star.jpg" alt="">
   </a>
   <div class="desc">Neutron star image</div>
</div>
</body>
</html>
```

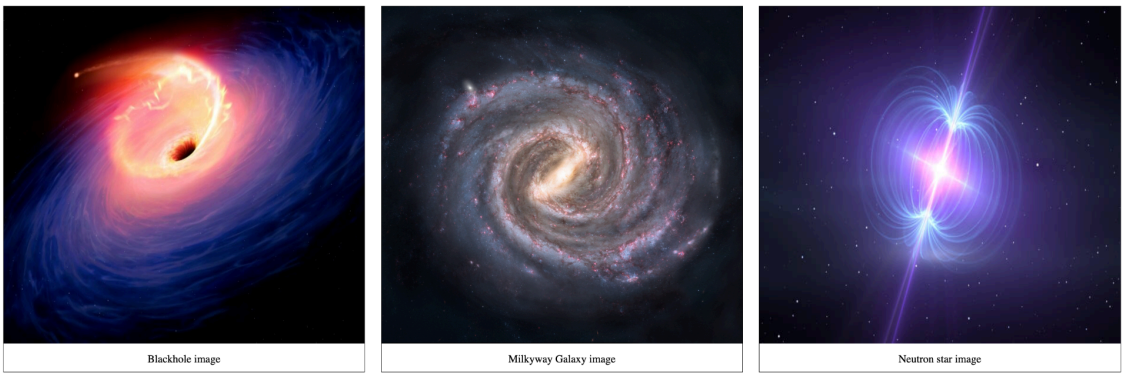
CSS

```
.pg1 {
   display: inline-block;
   width: 30%;
   border: 1px solid black;
   margin: 10px;
   }
   .pg1 img{
       width: 100%;
       height: 500px;
   }
   h1{
```

```
        text-align: center;
        font-family:Georgia, 'Times New Roman', Times, serif;
    }
    .pg1 .desc{
        padding: 10px;
        text-align: center;
    }
    .pg1:hover {
        transform: scale(1.05);
        box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2);
    }
```

OUTPUT:

Photo Gallery



Blackhole image | Milkyway Galaxy image | Neutron star image

Conclusion:

The project successfully created a visually appealing photo gallery using HTML and CSS, featuring a grid layout, hover effects, and responsive design. The use of Visual Studio Code 5

References:

- http://www.w3schools.com/css/css_grid.asp - W3Schools - CSS Grid Layout
- http://www.w3schools.com/css/css3_flexbox.asp - W3Schools - CSS Flexbox
- http://www.w3schools.com/css/css3_transitions.asp - W3Schools - CSS Hover Effects
- https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Grid_Layout - MDN Web Docs - Using CSS Grid

- US/docs/Learn/CSS/Building_blocks/Responsive_design - MDN Web Docs - Responsive Design