Interactive Photo Gallery

Documentation

Introduction

This project is an interactive photo gallery built with HTML, CSS, and JavaScript. The gallery is designed to be responsive, accurately match provided Figma designs, and include hover interactions to display additional details about each photo. The project is also compatible on different browser and follows best practices for code quality.

- ➢ Features: Responsive Design: Adapts to different screen sizes and devices using CSS media.
- ➢ Queries: Hover Interaction Displays additional details about each photo when a user hovers over it.
- ➢ Cross-browser Compatibility: Ensures consistent performance across modern web browsers, including Chrome, Firefox and Edge.
- ➢ Code Quality: Clean, readable, and well-structured code.

Project Structure

The project consists of the following files:

- ✓ `photo_gralley.html`
- ✓ `styles.css`
- ✓ `script.js`

Photo_gralley.html

The `index.html` file contains the basic HTML structure for the photo gallery.

```
<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Interactive Photo Gallery</title>

<link rel="stylesheet" href="styles.css">

</head>

<body>

  <div class="gallery">

    <div class="photo" data-details="image Details">
```

```
            <img src="image.jpg" alt="image">
        </div>
        <div class="photo" data-details="function Details">
            <img src="function.png" alt="function">
        </div>
        <div class="photo" data-details="picture Details">
            <img src="picture.jpg" alt="Picture">
        </div>
        <div class="photo" data-details="photo Details">
            <img src="photo.jpg" alt="photo">
        </div>
        <!-- Add more photos as needed -->
    </div>
    <script src="script.js"></script>
</body>
</html>
```

- **styles.css**

This   file contains the CSS styles for the gallery, including responsiveness and hover interactions.

```
body {
    font-family: Arial, sans-serif;
    margin: 0;
    padding: 0;
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
    background-color: #f0f0f0;
}
```

```
.gallery {

   display: flex;

   flex-wrap: wrap;

   gap: 16px;

   padding: 20px;

   max-width: 1200px;

   width: 100%;

   box-sizing: border-box;

}


.photo {

position: relative;

overflow: hidden;

flex: 1 1 calc(33.333% - 32px);

max-width: calc(33.333% - 32px);

box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);

cursor: pointer;

}

.photo img {

width: 100%;

height: auto;

display: block;

transition: transform 0.3s ease;

}

.photo:hover img {

transform: scale(1.1);

}

.photo::after {

content: attr(data-details);
position: absolute;
bottom: 0;
left: 0;
right: 0;
background: rgba(0, 0, 0, 0.7);
color: white;
padding: 10px;
font-size: 14px;
transform: translateY(100%);
transition: transform 0.3s ease;
text-align: center;
}
.photo:hover::after {
transform: translateY(0);
}
@media (max-width: 768px) {
.photo {
flex: 1 1 calc(50% - 16px);
max-width: calc(50% - 16px);
}
}

@media (max-width: 480px) {
  .photo {
    flex: 1 1 100%;
    max-width: 100%;
  }
}
```

script.js

script.js

- The `script.js` file contains the JavaScript code for handling additional hover interactions if needed. The current implementation primarily uses CSS for hover effects.

```
document.addEventListener('DOMContentLoaded', () => {
   const photos = document.querySelectorAll('.photo');
 photos.forEach(photo => {
       photo.addEventListener('mouseover', () => {
          const details = photo.getAttribute('data-details');
          // Additional hover logic can go here
       });
 photo.addEventListener('mouseout', () => {
          // Logic to revert any changes on mouse out can go here
       });
   });
});
```

Usage

1. Clone the repository.

2. Open `index.html` in your web browser.

3. Add your own photos and details by editing the HTML file.

Compatibility

This project has been tested on the latest versions of Chrome, Firefox and Edge.

I used VS Code (visual studio as my preferred editor)

Designer: Mr NIYOMUKIZA EMMANUEL