

Image Gallery with Lightbox

Let's break down this assignment into clear steps so you can understand and learn the core concepts while building it.

Steps to Build the Image Gallery with Lightbox Effect

1. Project Setup

Create a folder for your project and inside it, create the following files:

- `index.html` → Structure of the gallery
- `style.css` → Styling with CSS Grid & Flexbox
- `script.js` → JavaScript for modal functionality

2. HTML: Structuring the Gallery

We'll create a simple grid layout where clicking an image will open a larger version in a lightbox.

Key HTML Elements

- A container `<div class="gallery">` for the images.
- A modal (`<div class="modal">`) that appears when an image is clicked.
- Inside the modal, we'll have:
 - The large image
 - A close button

```
html

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Image Gallery with Lightbox</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <div class="gallery">
    
    
    
    
    
    
  </div>

  <div class="modal">
    <span class="close">&times;</span>
    <img class="modal-content" id="modal-img">
  </div>

  <script src="script.js"></script>
</body>
</html>
```

3. CSS: Styling the Gallery and Modal

We'll use **CSS Grid** to display images in a clean grid layout and **Flexbox** for the modal.

```
css

/* Reset some default styles */
* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
}

/* Gallery Styling */
.gallery {
  display: grid;
  grid-template-columns: repeat(auto-fit, minmax(200px, 1fr));
  gap: 10px;
  padding: 20px;
}

.gallery img {
  width: 100%;
  height: auto;
  border-radius: 8px;
  cursor: pointer;
  transition: transform 0.3s ease;
}

.gallery img:hover {
  transform: scale(1.05);
}

/* Modal Styling */
.modal {
  display: none;
  position: fixed;
  top: 0;
  left: 0;
  width: 100%;
  height: 100%;
  background: rgba(0, 0, 0, 0.8);
  justify-content: center;
  align-items: center;
}

.modal-content {
  max-width: 80%;
  max-height: 80%;
  border-radius: 10px;
}

.close {
  position: absolute;
  top: 20px;
  right: 30px;
  font-size: 40px;
  color: white;
  cursor: pointer;
}
```

4. JavaScript: Adding Lightbox Effect

We'll handle three things using JavaScript:

1. **Detect when an image is clicked** and open the modal.

2. Display the clicked image inside the modal.
3. Close the modal when clicking outside the image or pressing the close button.

```
js

// Select elements
const modal = document.querySelector(".modal");
const modalImg = document.getElementById("modal-img");
const closeBtn = document.querySelector(".close");
const galleryItems = document.querySelectorAll(".gallery-item");

// Function to open modal and show the clicked image
galleryItems.forEach((item) => {
  item.addEventListener("click", function () {
    modal.style.display = "flex"; // Show the modal
    modalImg.src = this.src;      // Set modal image to clicked image
  });
});

// Function to close modal when clicking the close button
closeBtn.addEventListener("click", function () {
  modal.style.display = "none";
});

// Function to close modal when clicking outside the image
modal.addEventListener("click", function (e) {
  if (e.target === modal) {
    modal.style.display = "none";
  }
});
```

Key Learnings from This Project

1. CSS Grid & Flexbox

- Grid for structuring the gallery.
- Flexbox for centering the modal.

2. JavaScript DOM Manipulation

- `document.querySelectorAll()` to select all images.
- `addEventListener()` to handle click events.

3. Modal Window Design

- Display a pop-up image inside a modal.
- Close it when clicking outside or pressing the close button.

Final Code (All Together)

Here's the complete code for your assignment.

index.html

```
html

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Image Gallery with Lightbox</title>
```

```
<link rel="stylesheet" href="style.css">
</head>
<body>
  <div class="gallery">
    
    
    
    
    
    
  </div>

  <div class="modal">
    <span class="close">&times;</span>
    <img class="modal-content" id="modal-img">
  </div>

  <script src="script.js"></script>
</body>
</html>
```

style.css

```
css

* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
}

.gallery {
  display: grid;
  grid-template-columns: repeat(auto-fit, minmax(200px, 1fr));
  gap: 10px;
  padding: 20px;
}

.gallery img {
  width: 100%;
  height: auto;
  border-radius: 8px;
  cursor: pointer;
  transition: transform 0.3s ease;
}

.gallery img:hover {
  transform: scale(1.05);
}

.modal {
  display: none;
  position: fixed;
  top: 0;
  left: 0;
  width: 100%;
  height: 100%;
  background: rgba(0, 0, 0, 0.8);
  justify-content: center;
  align-items: center;
}

.modal-content {
  max-width: 80%;
  max-height: 80%;
  border-radius: 10px;
}

.close {
  position: absolute;
  top: 20px;
```

```
right: 30px;
font-size: 40px;
color: white;
cursor: pointer;
}
```

script.js

```
js

const modal = document.querySelector(".modal");
const modalImg = document.getElementById("modal-img");
const closeBtn = document.querySelector(".close");
const galleryItems = document.querySelectorAll(".gallery-item");

galleryItems.forEach((item) => {
  item.addEventListener("click", function () {
    modal.style.display = "flex";
    modalImg.src = this.src;
  });
});

closeBtn.addEventListener("click", function () {
  modal.style.display = "none";
});

modal.addEventListener("click", function (e) {
  if (e.target === modal) {
    modal.style.display = "none";
  }
});
```

Next Steps

- Try adding **keyboard navigation** (pressing `Esc` should close the modal).
- Add **animations** to make the modal open smoothly.
- Load images from an **external API** to make the gallery dynamic.

Let me know if you have any doubts! 🚀

ChatGPT can make mistakes. Check important info.