“Design and Implementation of an Interactive Image Gallery using HTML, CSS, and JavaScript”

**BATCH-18**

# 231FA19011

***mubeena***

***Cyber Security(ACSE)***

**Section-A Vignan University**

[**231fa19011@gmail.com**](mailto:231fa19011@gmail.com)

# 231FA19032

***susmita***

***Cyber Security(ACSE)***

**Section-A Vignan University**

[**231fa19032@gmail.com**](mailto:231fa19032@gmail.com)

# 231FA19038

***santosh***

***Cyber Security(ACSE)***

**Section-A Vignan University**

[**231fa19038@gmail.com**](mailto:231fa19038@gmail.com)

# 231FA19042

***haneesh***

***Cyber Security(ACSE)***

**Section-A Vignan University**

[**231fa19042@gmail.com**](mailto:231fa19042@gmail.com)

***Abstract*—This project presents the design and implementation of an Interactive Image Gallery using HTML, CSS, and JavaScript. The gallery displays images in a grid layout, allowing users to view thumbnails in an organized and visually appealing manner. On clicking a thumbnail, a modal window opens to display the full-size version of the image, providing a focused viewing experience. The modal includes a close button and supports closing when the user clicks outside the image. Additional styling elements such as a gradient background, hover effects, shadows, rounded corners, and smooth animations enhance the user experience and overall aesthetics. This project demonstrates fundamental concepts of web design and interactivity, combining structure (HTML), styling (CSS), and dynamic behavior (JavaScript).**

1. Introduction

Interactive Image Gallery provides an effective way to display and explore collections of images on a website. This project is built using HTML, CSS, and JavaScript, where images are arranged in a grid layout. By clicking on a thumbnail, users can view the image in a full-screen modal window with a close option for easy navigation. The gallery includes visual enhancements like background gradients, hover effects, and animations, making it both functional and visually appealing. It demonstrates the use of core web technologies to create an engaging and user-friendly interface.

1. System Sections
2. For developing the Interactive Image Gallery, the system was designed using HTML, CSS, and JavaScript due to their simplicity, compatibility, and efficiency in web development.
3. HTML (HyperText Markup Language): Chosen to provide the structure of the gallery by organizing images in a grid layout and defining modal components.
4. CSS (Cascading Style Sheets): Selected for styling and presentation, enabling features like background gradients, hover effects, shadows, and responsive design.
5. JavaScript: Used to implement interactivity, such as opening images in a modal, closing the modal, and handling user events effectively.
6. These technologies were selected because they are lightweight, platform-independent, and widely supported across all

modern browsers. Together, they allow for the creation of a responsive, interactive, and visually attractive image gallery without requiring complex frameworks or heavy back-end systems

1. Quiz Termination and Result Display

The developed Interactive Image Gallery successfully displays images in a grid layout with hover effects and styling. On clicking a thumbnail, the selected image opens in a modal window with a close option. The gallery is responsive, user-friendly, and visually appealing, meeting the objectives of the project.The countdown timer reaches zero, leading to automatic.

* Test Interface Design

The gallery interface was tested for thumbnail display,

opening/closing, responsiveness, and browser

compatibility.All functions worked correctly, ensuring

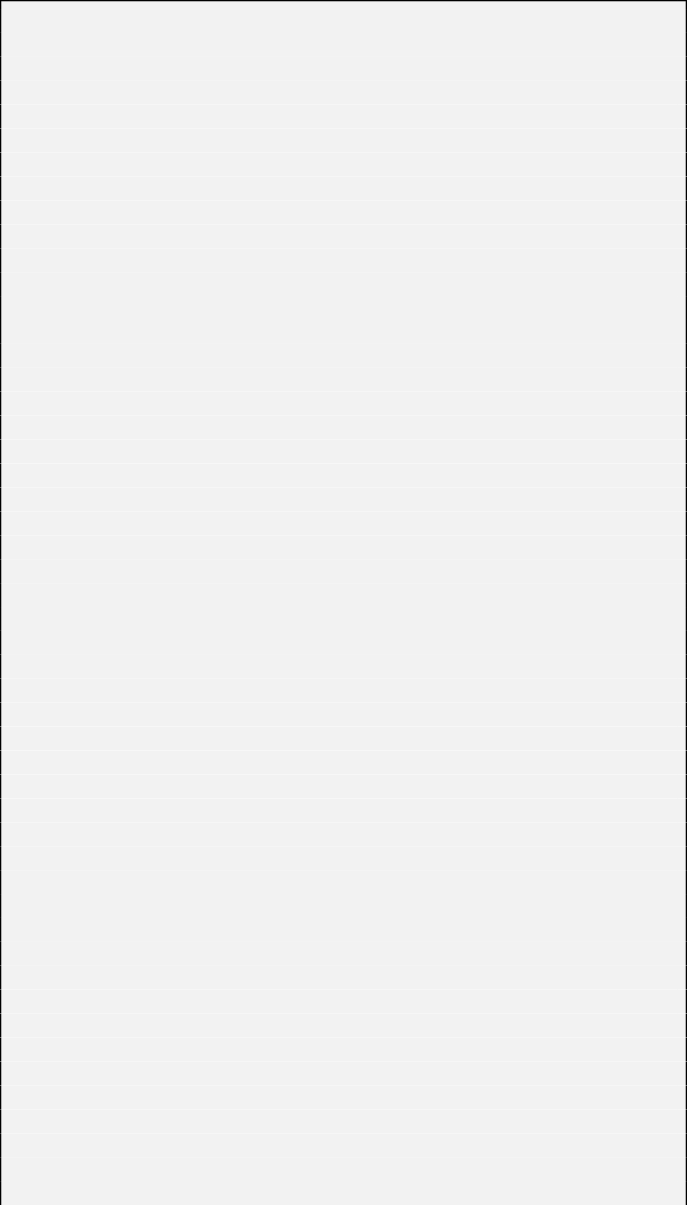
a smooth, interactive, and user-friendly design

1. Implementation Code

The following code snippet shows the core structure of the image website using HTML, CSS, and JavaScript:

**Algorithm 1**

1. **Create the Gallery Layout**
   * Use HTML to display a grid of image thumbnails.
   * Each thumbnail has a small-sized image.
2. **Design the Modal**
   * Create a hidden modal box in HTML that will display the full-size image.
   * Include:
     + A container (div) for the image.
     + A close button (× symbol).
3. **Style with CSS**
   * Use CSS Grid or Flexbox to arrange thumbnails in a grid.
   * Style the modal to:
     + Cover the full screen (position: fixed).
     + Darken the background.
     + Center the full-size image.
4. **Add JavaScript Interactivity**
   * **Step 1:** Select all thumbnail images.
   * **Step 2:** Add an event listener (onclick) to each thumbnail.
     + When clicked, show the modal and set the src of the modal image to match the clicked thumbnail.
   * **Step 3:** Add event listener to the **close button**.
     + When clicked, hide the modal



<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Interactive Image Gallery</title>

<style>

body {

font-family: "Segoe UI", Tahoma, Geneva, Verdana, sans-serif;

margin: 0;

background: linear-gradient(135deg, #89f7fe, #66a6ff);

color: #333;

}

/\* Heading Style \*/

h1 {

text-align: center;

padding: 20px;

color: #fff;

text-shadow: 2px 4px rgba(0,0,0,0.4);

font-size: 2.5rem;

}

/\* Gallery Grid \*/

.gallery {

display: grid;

grid-template-columns: repeat(auto-fill, minmax(180px, 1fr));

gap: 15px;

padding: 20px;

}

}

gallery img {

width: 100%;

height: auto;

cursor: pointer;

border-radius: 12px;

box-shadow: 0px 4px 12px rgba(0,0,0,0.2);

transition: transform 0.3s, box-shadow 0.3s;

}

.gallery img:hover {

transform: scale(1.07);

box-shadow: 0px 6px 16px rgba(0,0,0,0.3);

}

/\* Modal styles \*/

.modal {

display: none;

position: fixed;

z-index: 1000;

left: 0;

top: 0;

width: 100%;

height: 100%;

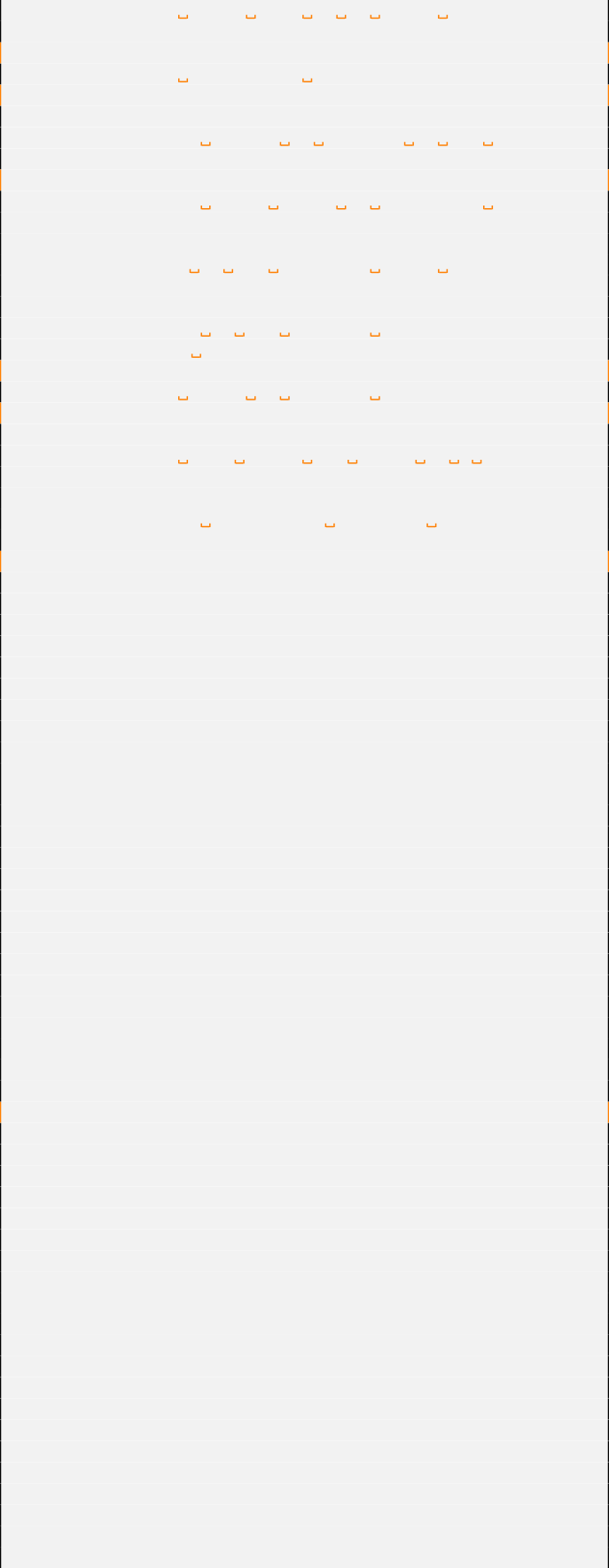
background-color: rgba(0,0,0,0.85);

justify-content: center;

align-items: center;

animation: fadeIn 0.4s;

}

.close {

position: absolute;

top: 20px;

right: 40px;

font-size: 40px;

font-weight: bold;

color: #fff;

cursor: pointer;

transition: color 0.3s;

}

.close:hover {

color: #ff4c4c;

}

/\* Animations \*/

@keyframes fadeIn {

from { opacity: 0; }

to { opacity: 1; }

}

@keyframes zoomIn {

from { transform: scale(0.7); opacity: 0; }

to { transform: scale(1); opacity: 1; }

}

</style>

</head>

<body>

<h1>🌟 My Stylish Image Gallery 🌟</h1>

<!-- Image Gallery -->

<div class="gallery">

<img src="https://via.placeholder.com/300x200?

text=Image+1" alt="Image 1">

<img src="https://via.placeholder.com/300x200?

text=Image+2" alt="Image 2">

<img src="https://via.placeholder.com/300x200?

text=Image+3" alt="Image 3">

<img src="https://via.placeholder.com/300x200?

text=Image+4" alt="Image 4">

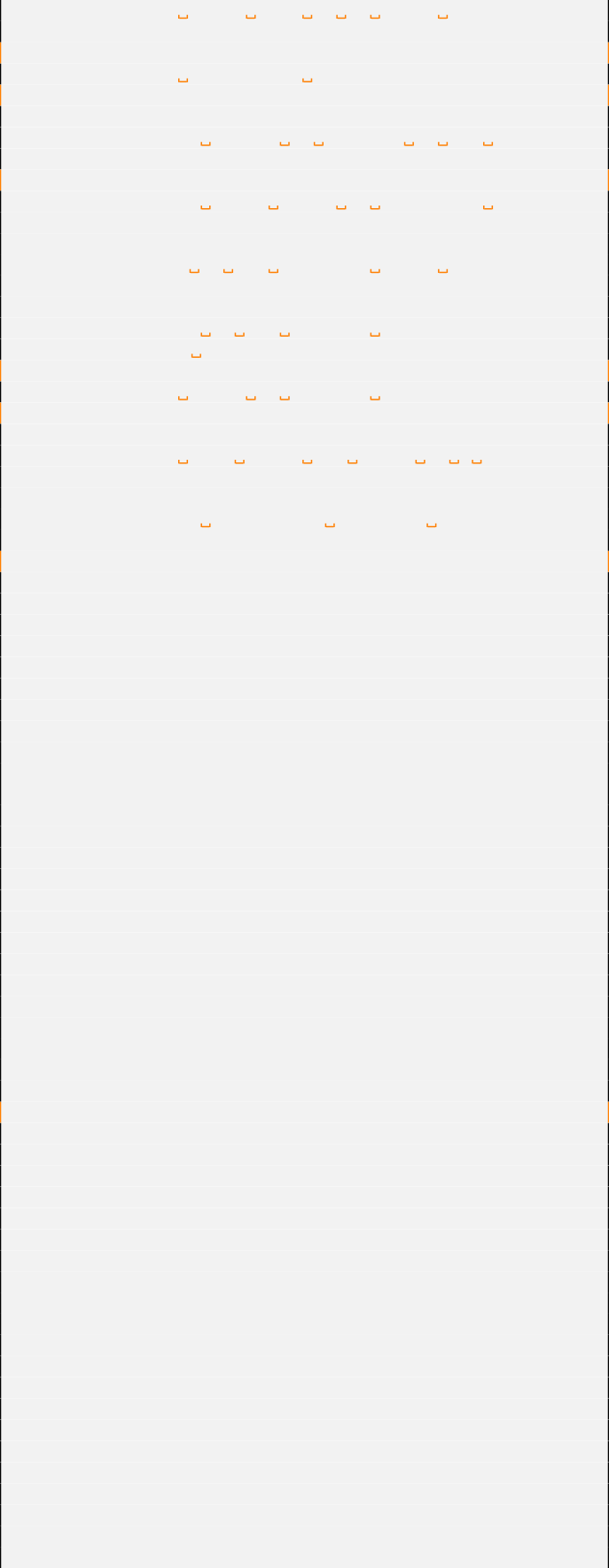
<img src="https://via.placeholder.com/300x200?

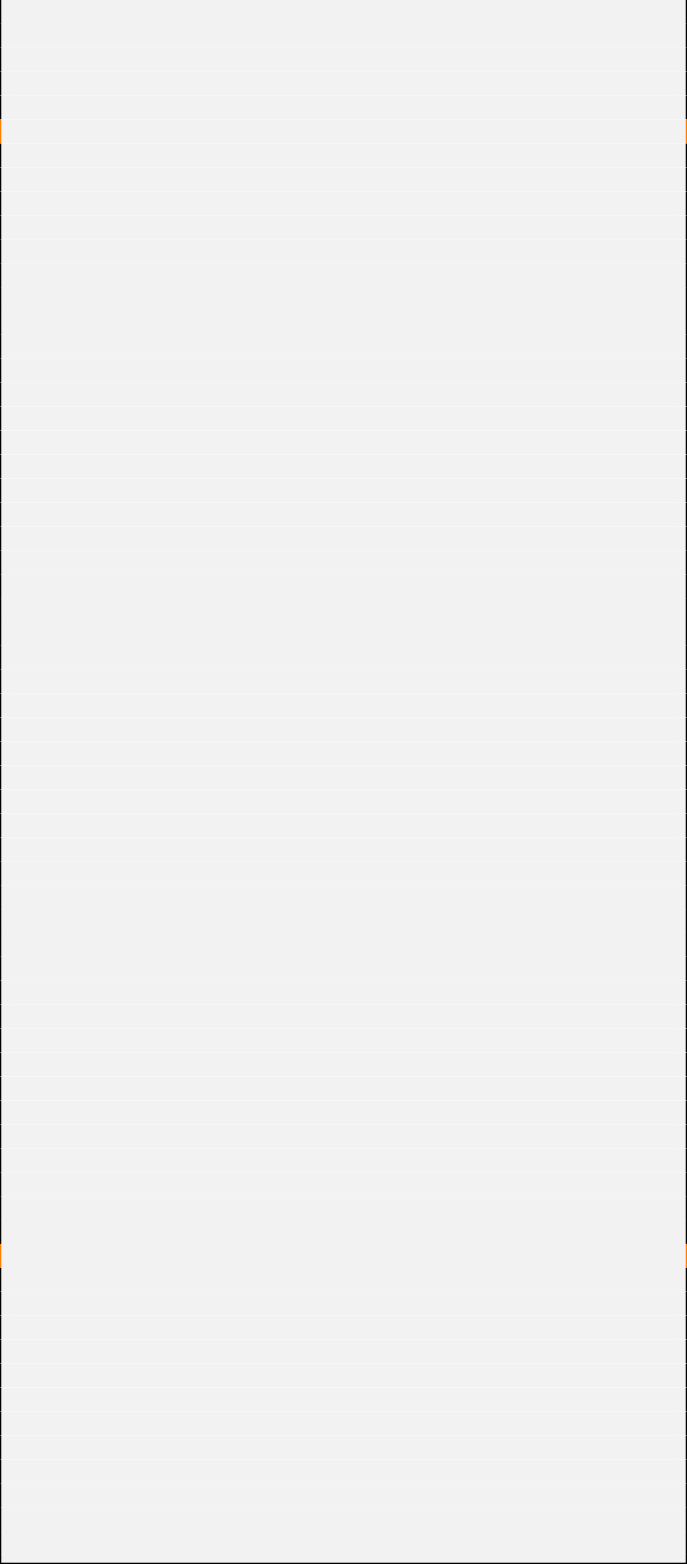
text=Image+5" alt="Image 5">

<img src="https://via.placeholder.com/300x200?

text=Image+6" alt="Image 6">

</div>





<!-- Modal -->

<div id="myModal" class="modal">

<span class="close">&times;</span>

<img class="modal-content" id="modalImage">

</div>

<script>

// Get modal and elements

const modal = document.getElementById("myModal");

const modalImg = document.getElementById

("modalImage");

const closeBtn = document.querySelector

(".close");

// Select all images in gallery

const images =

document.querySelectorAll(".gallery img");

// Add click event to each thumbnail

images.forEach(img => {

img.addEventListener("click", () => {

modal.style.display = "flex";

modalImg.src = img.src;

});

});

// Close modal when close button is clicked

closeBtn.addEventListener("click", () => {

modal.style.display = "none";

});

// Close modal when user clicks outside the

image

modal.addEventListener("click", (e) => {

if (e.target === modal) {

modal.style.display = "none";

}

});

gallery img {

width: 100%;

height: auto;

cursor: pointer;

border-radius: 12px;

box-shadow: 0px 4px 12px rgba(0,0,0,0.2);

transition: transform 0.3s, box-shadow 0.3s;

}

.gallery img:hover {

transform: scale(1.07);

box-shadow: 0px 6px 16px rgba(0,0,0,0.3);

}

/\* Modal styles \*/

.modal {

display: none;

position: fixed;

z-index: 1000;

left: 0;

top: 0;

width: 100%;

height: 100%;

background-color: rgba(0,0,0,0.85);

justify-content: center;

align-items: center;

animation: fadeIn 0.4s;

}

.close {

position: absolute;

top: 20px;

right: 40px;

font-size: 40px;

font-weight: bold;

color: #fff;

cursor: pointer;

transition: color 0.3s;

}

.close:hover {

color: #ff4c4c;

}

/\* Animations \*/

@keyframes fadeIn {

from { opacity: 0; }

to { opacity: 1; }

}

@keyframes zoomIn {

from { transform: scale(0.7); opacity: 0; }

to { transform: scale(1); opacity: 1; }

}

</style>

</head>

<body>

<h1>🌟 My Stylish Image Gallery 🌟</h1>

<!-- Image Gallery -->

<div class="gallery">

<img src="https://via.placeholder.com/300x200?

text=Image+1" alt="Image 1">

<img src="https://via.placeholder.com/300x200?

text=Image+2" alt="Image 2">

<img src="https://via.placeholder.com/300x200?

text=Image+3" alt="Image 3">

<img src="https://via.placeholder.com/300x200?

text=Image+4" alt="Image 4">

<img src="https://via.placeholder.com/300x200?

text=Image+5" alt="Image 5">

<img src="https://via.placeholder.com/300x200?

text=Image+6" alt="Image 6">

</div>

</script>

</body>

</html>

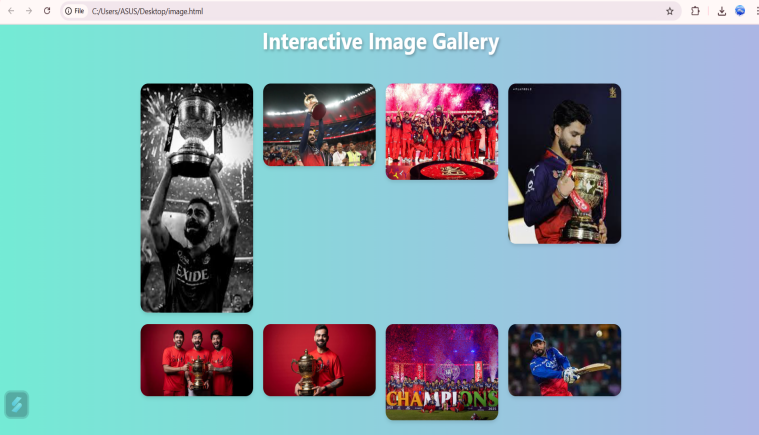


Fig. 1. Img gallery.

1. RESULTS

The Interactive Image Gallery successfully displays thumbnails in a grid and allows users to view images in a full-size modal with a close option, providing a responsive and user-friendly interface.

1. Conclusion

The Interactive Image Gallery effectively combines HTML, CSS, and JavaScript to create a simple, responsive, and user-friendly system. It fulfills the objective of providing an interactive and visually appealing way to display and view images on the web.

References

1. J. Duckett, *HTML and CSS: Design and Build Websites*. Wiley, 2011.
2. M. Flanagan, *JavaScript: The Definitive Guide*, 7th ed. O’Reilly Media, 2020.
3. A. Al-Azawei, P. Parslow, and K. Lundqvist, “Barriers and opportunities of e-learning implementation,” *IR- RODL*, vol. 17, no. 5, pp. 126–146, 2016.
4. IEEE, “IEEE Editorial Style Manual,” 2023. [Online]. Available: https://journals.ieeeauthorcenter.ieee.org/.