A close up of a logo

Description automatically generated

**CSS Styling**

**Disclaimer: The content is curated from online/offline resources and used for educational purpose only**

**LAB MANUAL**

**Creating Interactive Card Components with CSS Effects**

**Objective:**

The objective of this activity is to help learners understand how to enhance web page interactivity and visual appeal by creating card components styled and animated using CSS. Participants will learn to apply CSS properties such as transitions, transforms, shadows, and hover effects to produce engaging, responsive card designs. By the end of the activity, learners will be able to craft visually dynamic card components that respond to user interaction smoothly.

**Equipment Required:**

* A computer with a text editor (e.g., VS Code, Sublime Text, Notepad++)
* A modern web browser (Chrome, Firefox, Edge, etc.) with developer tools

**Prerequisites:**

* Basic understanding of HTML structure and tags
* Familiarity with fundamental CSS concepts such as selectors, properties, and values
* Knowledge of saving .html files and linking CSS to HTML

**Problem Statement:**

In this activity, you are tasked with creating one or more interactive card components that showcase content such as an image, title, description, and a call-to-action button. The cards should incorporate CSS effects like smooth hover transitions, scale transformations, shadows, and color changes to provide an engaging user experience. You will practice writing both the HTML structure for cards and the CSS styling that makes them interactive and visually appealing.

**Procedure:**

1. Create a new file named ***interactive\_cards.html, interactive\_cards.css, and interactive\_cards.js***.
2. Write the basic HTML structure including <head> and <body>.
3. Inside the <body>, create one or more card components using HTML elements such as <div>, <img>, <h3>, <p>, and <button>.
4. Add a <style> block in the <head> or link an external CSS file.
5. Use CSS to style the card components including layout, fonts, colors, and sizing.
6. Add CSS effects such as hover transitions, box shadows, and scaling using :hover and transition properties.
7. Save the file and open it in a browser to test the interactive effects on mouse hover.

**Code**

***interactive\_cards.html***

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Interactive Card Components</title>

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

<!-- Linking the external CSS file -->

<link rel="stylesheet" href="interactive\_cards.css">

</head>

<body>

<!-- Card 1 -->

<div class="card">

    <img src="C:\Users\prash\Desktop\EDUNET\Module 2 practice\dog.jpg" alt="Dog Image">

    <div class="card-content">

        <h3>Card Title One</h3>

        <p>This is a description for the first card. It highlights interesting content and engages the user.</p>

        <button onclick="showMessage('Dog Card')">Learn More</button>

    </div>

</div>

<!-- Card 2 -->

<div class="card">

    <img src="C:\Users\prash\Desktop\EDUNET\Module 2 practice\elephant.jpg" alt="Elephant Image">

    <div class="card-content">

        <h3>Card Title Two</h3>

        <p>Here’s another card example with unique content and interactive CSS effects on hover.</p>

        <button onclick="showMessage('Elephant Card')">Discover</button>

    </div>

</div>

<!-- Card 3 -->

<div class="card">

    <img src="C:\Users\prash\Desktop\EDUNET\Module 2 practice\gorilla.jpg" alt="Gorilla Image">

    <div class="card-content">

        <h3>Card Title Three</h3>

        <p>This card showcases the gorilla image to complete the animal card set.</p>

        <button onclick="showMessage('Gorilla Card')">Explore</button>

    </div>

</div>

<div id="messageBox" class="message-box"></div>

<!-- Linking the external JavaScript file -->

<script src="interactive\_cards.js"></script>

</body>

</html>

***Interactive\_cards.css***

/\* Basic page styling \*/

body {

    font-family: 'Inter', sans-serif;

    background-color: #f0f2f5;

    margin: 0;

    padding: 40px;

    display: flex;

    justify-content: center;

    gap: 30px;

    flex-wrap: wrap;

}

/\* Card container styling \*/

.card {

    background-color: white;

    width: 300px;

    border-radius: 12px;

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

    overflow: hidden;

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

    cursor: pointer;

    display: flex;

    flex-direction: column;

    text-align: center;

}

/\* Hover effect: enlarge and deepen the shadow \*/

.card:hover {

    transform: scale(1.05);

    box-shadow: 0 12px 24px rgba(0,0,0,0.2);

}

/\* Card image styling \*/

.card img {

    width: 100%;

    height: 180px;

    object-fit: cover;

    display: block;

}

/\* Content inside the card \*/

.card-content {

    padding: 20px;

    flex-grow: 1;

}

/\* Card title \*/

.card-content h3 {

    margin: 0 0 12px 0;

    font-size: 1.4rem;

    color: #333;

}

/\* Card description text \*/

.card-content p {

    font-size: 1rem;

    color: #666;

    margin-bottom: 20px;

}

/\* Call-to-action button \*/

.card-content button {

    background-color: #007bff;

    border: none;

    color: white;

    padding: 12px 25px;

    font-size: 1rem;

    border-radius: 25px;

    cursor: pointer;

    transition: background-color 0.3s ease;

    outline: none;

}

/\* Button hover effect \*/

.card-content button:hover {

    background-color: #0056b3;

}

/\* Message box styling \*/

.message-box {

    position: fixed;

    bottom: 20px;

    left: 50%;

    transform: translateX(-50%);

    background-color: #334155;

    color: white;

    padding: 15px 30px;

    border-radius: 8px;

    box-shadow: 0 10px 15px -3px rgba(0, 0, 0, 0.1);

    z-index: 1000;

    opacity: 0;

    visibility: hidden;

    transition: opacity 0.5s, visibility 0.5s;

}

.message-box.show {

    opacity: 1;

    visibility: visible;

}

***interactive\_cards.js***

// Simple JavaScript interaction for card buttons

function showMessage(cardName) {

    const messageBox = document.getElementById('messageBox');

    if (messageBox) {

        messageBox.textContent = "You clicked on " + cardName;

        messageBox.classList.add('show');

        setTimeout(() => {

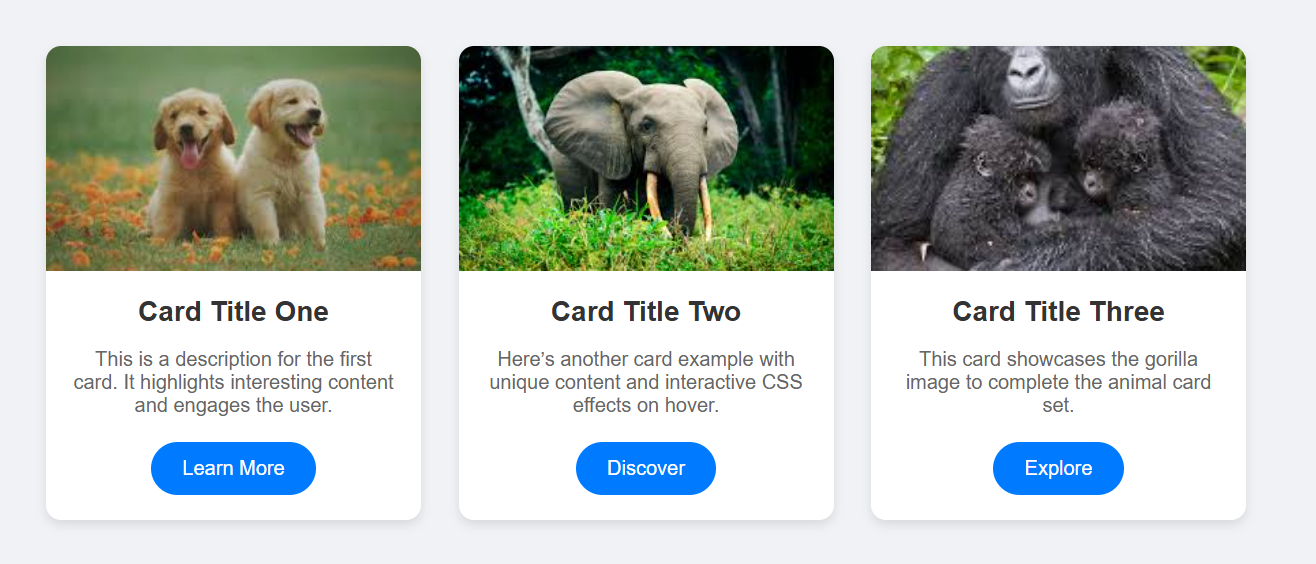
            messageBox.classList.remove('show');

        }, 3000);

    }

}

**Output**

****