NAME: JANHVI SINGH

REG NO: 22BCE8544

MOD 6

HTML CODE:

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

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

  <title>Feedback Form</title>

</head>

<body>

  <h1>Employee Feedback Form</h1>

  <form id="feedbackForm">

    <label for="userName">Name:</label>

    <input type="text" id="userName" name="userName" required>

    <br><br>

    <label for="userEmail">Email:</label>

    <input type="email" id="userEmail" name="userEmail" required>

    <br><br>

    <label for="feedbackScore">Rate Us (1-5):</label>

    <div id="ratingOptions">

      <input type="radio" id="rating1" name="rating" value="1">

      <label for="rating1">1</label>

      <input type="radio" id="rating2" name="rating" value="2">

      <label for="rating2">2</label>

      <input type="radio" id="rating3" name="rating" value="3">

      <label for="rating3">3</label>

      <input type="radio" id="rating4" name="rating" value="4">

      <label for="rating4">4</label>

      <input type="radio" id="rating5" name="rating" value="5">

      <label for="rating5">5</label>

    </div>

    <p id="ratingMessage"></p>

    <br><br>

    <button type="submit">Submit Feedback</button>

    <button type="button" id="resetButton">Reset</button>

  </form>

  <script src="feedback\_6.js"></script>

</body>

</html>

JS CODE:

// Function to handle rating change

function handleRatingChange(event) {

  const selectedRating = event.target.value;

  document.getElementById("ratingMessage").textContent = `Thank you for rating us: ${selectedRating}`;

}

// Function to validate fields

function validateFields() {

  const userName = document.getElementById("userName").value.trim();

  const userEmail = document.getElementById("userEmail").value.trim();

  const selectedRating = document.querySelector('input[name="rating"]:checked');

  if (userName === "") {

    alert("Name cannot be empty.");

    return false;

  }

  if (!userEmail.includes("@")) {

    alert("Invalid email format.");

    return false;

  }

  if (!selectedRating) {

    alert("Please select a rating.");

    return false;

  }

  return true;

}

// Function to handle form submission

function submitFeedback(event) {

  event.preventDefault(); // Prevent default form submission

  // Validate form fields

  if (!validateFields()) {

    return;

  }

  const feedbackScores = [5, 4, 3, 5, 2];

  console.log("Feedback scores:");

  feedbackScores.forEach((score) => console.log(score));

  alert("Feedback submitted successfully!");

}

// Function to reset the form

function resetForm() {

  document.getElementById("feedbackForm").reset();

  document.getElementById("ratingMessage").textContent = "";

}

// Attach event listeners

document.getElementById("feedbackForm").addEventListener("submit", submitFeedback);

document.getElementById("resetButton").addEventListener("click", resetForm);

document.querySelectorAll('input[name="rating"]').forEach((radio) => {

  radio.addEventListener("change", handleRatingChange);

});

README :

**# Feedback Form - JavaScript Interactivity**

**## Overview**

This project enhances the feedback form by adding dynamic interactivity using JavaScript. It includes features like form validation, event handling, and dynamic messages.

**## Features**

1. **\*\*Conditionals for Validation:\*\***

   - Checks if the name is empty.

   - Validates if the email contains an "@" symbol.

   - Ensures a rating is selected.

2. **\*\*Loops for Dynamic Feedback:\*\***

   - Displays feedback scores from an array using a `forEach` loop.

3. **\*\*Event Handling:\*\***

   - Attaches event listeners for form submission, reset button, and rating change.

4. **\*\*Dynamic Rating System:\*\***

   - Displays a thank-you message when a user selects a rating.

**## Challenges and Solutions**

- **\*\*Challenge:\*\*** Dynamically validating multiple fields.

  - **\*\*Solution:\*\*** Created a reusable `validateFields` function.

- **\*\*Challeng# Feedback Form - JavaScript Interactivity**

**## Overview**

**This project enhances the feedback form by adding dynamic interactivity using JavaScript. It includes features like form validation, event handling, and dynamic messages.**

**## Features**

**1. \*\*Conditionals for Validation:\*\***

**- Checks if the name is empty.**

**- Validates if the email contains an "@" symbol.**

**- Ensures a rating is selected.**

**2. \*\*Loops for Dynamic Feedback:\*\***

**- Displays feedback scores from an array using a `forEach` loop.**

**3. \*\*Event Handling:\*\***

**- Attaches event list# Feedback Form - JavaScript Interactivity**

**## Overview**

**This project enhances the feedback form by adding dynamic interactivity using JavaScript. It includes features like form validation, event handling, and dynamic messages.**

**## Features**

**1. \*\*Conditionals for Validation:\*\***

**- Checks if the name is empty.**

**- Validates if the email contains an "@" symbol.**

**- Ensures a rating is selected.**

**2. \*\*Loops for Dynamic Feedback:\*\***

**- Displays feedback scores from an array using a `forEach` loop.**

**3. \*\*Event Handling:\*\***

**- Attaches event listeners for form submission, reset button, and rating change.**

**4. \*\*Dynamic Rating System:\*\***

**- Displays a thank-you message when a user selects a rating.**

**## Challenges and Solutions**

**- \*\*Challenge:\*\* Dynamically validating multiple fields.**

**- \*\*Solution:\*\* Created a reusable `validateFields` function.**

**- \*\*Challenge:\*\* Resetting dynamic messages.**

**- \*\*Solution:\*\* Used the `resetForm` function to clear form fields and messages.**

**## How to Run**

**1. Open `feedback\_form.html` in a web browser.**

**2. Fill out the form and interact with the rating system.**

**3. Submit the form to see validations and logs in the console.**

**4. Reset the form using the reset button.eners for form submission, reset button, and rating change.**

**4. \*\*Dynamic Rating System:\*\***

**- Displays a thank-you message when a user selects a rating.**

**## Challenges and Solutions**

**- \*\*Challenge:\*\* Dynamically validating multiple fields.**

**- \*\*Solution:\*\* Created a reusable `validateFields` function.**

**- \*\*Challenge:\*\* Resetting dynamic messages.**

**- \*\*Solution:\*\* Used the `resetForm` function to clear form fields and messages.**

**## How to Run**

**1. Open `feedback\_form.html` in a web browser.**

**2. Fill out the form and interact with the rating system.**

**3. Submit the form to see validations and logs in the console.**

**4. Reset the form using the reset button.e:\*\*** Resetting dynamic messages.

  - **\*\*Solution:\*\*** Used the `resetForm` function to clear form fields and messages.

**## How to Run**

1. Open `feedback\_form.html` in a web browser.

2. Fill out the form and interact with the rating system.

3. Submit the form to see validations and logs in the console.

4. Reset the form using the reset button.

SCREENSHOT:

