**Exp-6 Feedback form using JS**

**Aim:** To make a feedback form using javascript.

**Theory:**

A feedback form is an essential user interface component used to collect opinions, ratings, and suggestions from users. It helps businesses and developers understand user experiences and improve their services.

In this experiment, a feedback form is created with input fields for the user's name, email, and mobile number, along with an option to upload a file. The form submission is handled using JavaScript, which prevents the default form submission behavior, collects input values, and displays them in the console for verification. Additionally, an alert is shown to confirm successful submission.

A star rating system is integrated into the form to allow users to rate their experience. Each star represents a numeric value, and clicking on a star highlights it along with all previous stars to visually indicate the selected rating. The rating is then displayed as text to provide immediate feedback to the user. Hover effects are also implemented to enhance the user experience by temporarily highlighting stars as the user moves their mouse over them.

Overall, this experiment demonstrates how JavaScript can be used to enhance form interactions, ensuring a smooth and responsive user experience.

**Code:**

**HTML:**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Sneh Naik - feedback form</title>

    <style>

        \* {

            padding: 5px;

            margin: 0%;

            box-sizing: border-box;

        }

        input {

            margin-bottom: 10px;

        }

        .stars {

            display: flex;

            font-size: 2rem;

            cursor: pointer;

        }

        .star {

            color: lightgray;

            transition: color 0.2s;

        }

        .star {

            color: gray;

            cursor: pointer;

            transition: color 0.2s;

        }

        .star.hover,

        .star.active {

            color: yellow;

        }

    </style>

</head>

<body>

    <div class="main">

        <form style="display: flex;flex-direction: column; width: 300px;">

            <label for="uname">Enter name:</label>

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

            <label for="email">Enter email:</label>

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

            <label for="mobile">Enter mobile number:</label>

            <input type="tel" name="mobile" id="" required>

            <label for="file">Upload file:</label>

            <input type="file" name="file" id="">

            <div class="stars" id="stars">

                <span class="star" data-value="1">&#9733;</span>

                <span class="star" data-value="2">&#9733;</span>

                <span class="star" data-value="3">&#9733;</span>

                <span class="star" data-value="4">&#9733;</span>

                <span class="star" data-value="5">&#9733;</span>

            </div>

            <p id="ratingText">Rating: 0</p>

            <input type="submit" value="Submit">

        </form>

    </div>

    <script src="form.js"></script>

</body>

</html>

**Javascipt:**

document.querySelector("form").addEventListener("submit", function (event) {

    event.preventDefault();

    const formData = new FormData(this);

    const data = {

        name: formData.get("uname"),

        email: formData.get("email"),

        mobile: formData.get("mobile"),

        file: formData.get("file") ? formData.get("file").name : "No file uploaded"

    };

    console.log("Form Submitted:", data);

    alert("Form submitted successfully!");

});

const stars = document.querySelectorAll('.star');

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

stars.forEach(star => {

    star.addEventListener('click', () => {

        const value = star.dataset.value;

        ratingText.textContent = `Rating: ${value}`;

        stars.forEach(s => s.classList.remove('active'));

        star.classList.add('active');

        let prev = star;

        while (prev.previousElementSibling) {

            prev = prev.previousElementSibling;

            prev.classList.add('active');

        }

    });

    star.addEventListener('mouseover', () => {

        stars.forEach(s => s.classList.remove('hover'));

        star.classList.add('hover');

        let prev = star;

        while (prev.previousElementSibling) {

            prev = prev.previousElementSibling;

            prev.classList.add('hover');

        }

    });

    star.addEventListener('mouseleave', () => {

        stars.forEach(s => s.classList.remove('hover'));

    });

});

**Output:**



