







Task-6

Interactive Web Application Description:

Create an interactive web application that responds to user input. Implement dynamic content updates, interactive forms, or real-time data display. Focus on enhancing user engagement through seamless interactions and responsiveness.

.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Interactive Student Feedback Dashboard</title>
  <link rel="stylesheet" href="../static/stylefeedback.css">
</head>
<body>
  <div class="container">
    <h1> Student Feedback Dashboard</h1>
    <form id="feedbackForm">
      <input type="text" id="name" placeholder="Your Name" required>
      <select id="rating" required>
        <option value="">Select Rating</option>
        <option value="5"> Excellent</option>
        <option value="4"> Good</option>
        <option value="3"> Average</option>
        <option value="2"> Poor</option>
        <option value="1"> Very Poor</option>
      </select>
      <textarea id="comment" placeholder="Your feedback..." rows="3" required></textarea>
      <button type="submit">Submit Feedback</button>
    </form>
    <div class="dashboard">
      <div class="card">
        <h2 id="totalResponses">0</h2>
        <p>Total Responses</p>
      </div>
      <div class="card">
        <h2 id="avgRating">0</h2>
        <p>Average Rating</p>
      </div>
      <div class="card">
        <h2 id="lastFeedback">N/A</h2>
      </div>
    </div>
  </div>
</body>
</html>
```

```

    <p>Last Feedback By</p>
  </div>
</div>
<div class="feedback-list" id="feedbackList">
  <h3>📝 Recent Feedback</h3>
  <p>No new feedback submitted yet.</p>
</div>
</div>

<script>
  let allFeedbacks = JSON.parse(localStorage.getItem("feedbacks") || "[]");
  let recentFeedbacks = [];
  function updateDashboard() {
    const total = allFeedbacks.length;
    document.getElementById("totalResponses").textContent = total;
    const avg = total > 0 ? (allFeedbacks.reduce((sum, f) => sum + f.rating, 0) /
total).toFixed(1) : 0;
    document.getElementById("avgRating").textContent = avg;
    const last = total > 0 ? allFeedbacks[allFeedbacks.length - 1].name : "N/A";
    document.getElementById("lastFeedback").textContent = last;
    // Update recent feedback list (new only)
    const list = document.getElementById("feedbackList");
    if(recentFeedbacks.length === 0){
      list.innerHTML = "<h3>📝 Recent Feedback</h3><p>No new feedback submitted
yet.</p>";
      return;
    }
    list.innerHTML = "<h3>📝 Recent Feedback</h3>";
    recentFeedbacks.slice().reverse().forEach(f => {
      const div = document.createElement("div");
      div.className = "feedback-item";
      div.innerHTML = `<strong>${f.name}</strong> rated
<b>${f.rating}/5</b><br>"${f.comment}"`;
      list.appendChild(div);
    });
  }
  updateDashboard();
  document.getElementById("feedbackForm").addEventListener("submit", function(e){
    e.preventDefault();
    const name = document.getElementById("name").value.trim();
    const rating = parseInt(document.getElementById("rating").value);
    const comment = document.getElementById("comment").value.trim();

```

```

const exists = allFeedbacks.some(f => f.name === name && f.comment === comment);
if(exists){
  alert("You have already submitted this feedback!");
  return;
}
const feedback = { name, rating, comment };
allFeedbacks.push(feedback);
recentFeedbacks.push(feedback); // Only new feedback
localStorage.setItem("feedbacks", JSON.stringify(allFeedbacks));
updateDashboard();
this.reset();
});
</script>
</body>
</html>

```

Stylefeedback.css

```

body {
  font-family: Arial, sans-serif;
  background-image: url("images/feedback.jpg");
  margin: 0;
  height: 100vh;
  display: flex;
  background-size: cover;
  background-position: left;
  background-repeat: no-repeat;
  line-height: 1.8;
  justify-content: center;
  align-items: center;
}
.container {
  max-width: 900px;
  width: 100%;
  background: url("images/gray.jpg") no-repeat center center/cover;
  padding: 25px;
  border-radius: 15px;
  box-shadow: 0 6px 15px rgba(20, 20, 20, 0.1);
}
h1 { text-align: center; color: #d3d3d4; }
h2 { text-align: center; color: #000000; }
h3 { text-align: center; color: #0c0c0c; }
form {
  display: flex;

```

```
flex-direction: column;
gap: 12px;
margin-bottom: 20px;
}
input, select, textarea, button {
padding: 10px;
border: 1px solid #d4aeae;
border-radius: 8px;
font-size: 15px;
width: 100%;
}
button {
background: #63ad73;
color: rgb(0, 0, 0);
border: none;
cursor: pointer;
transition: 0.3s;
}
button:hover { background: #3b663d; }
.dashboard {
margin-top: 20px;
display: grid;
grid-template-columns: repeat(auto-fit, minmax(250px, 1fr));
gap: 20px;
}
.card {
background: #afa8bd;
padding: 20px;
border-radius: 12px;
text-align: center;
box-shadow: 0 4px 10px rgba(0,0,0,0.1);
}
.card h2 { margin: 10px 0; }
.feedback-list {
margin-top: 30px;
}
.feedback-item {
background: #65a1cc;
padding: 12px;
border-radius: 8px;
margin-bottom: 10px;
}
```

```
.feedback-item strong { color: #1565c0; }
```