COLLEGE CODE : 9222

COLLEGE NAME : THENI KAMMAVAR SANGAM COLLEGE OF TECHNOLOGY

DEPARTMENT : B.TECH(INFORMATION TECHNOLOGY)

STUDENT NM-ID : A857FF10960EB4C8CDB46821869AA67B

REG NO : 922223205012

DATE : 17.10.2025

Completed the project named as Phase_5_ TECHNOLOGY

PROJECT NAME : IBM-NJ- FEEDBACK COLLECTION SYSTEM

SUBMITTED BY,

NAME : GOVINDARAJAN R

MOBILE : 8825464206

FEEDBACK COLLECTION SYSTEM

PROJECT DEMONSTRATION AND DOCUMENTATION

Final Demo Walkthrough :

Design & Responsiveness:

Showtheform: Pointoutthe clean layout and aesthetic choices (e.g., color scheme, spacing).

Demonstrate Responsiveness: Briefly resize the browser window to show how the form adjusts for mobile devices (if applicable).

Form Elements:

Gothrougheach input field and explain its purpose:

Name/Email: Standard contact fields.

Feedback Type (Radio/Dropdown): Crucial – Show how the user can select a category like "Bug Report," "Suggestion," or "General Comment."

Rating (Optional): If you included a star rating or emoji selection, demonstrate how selecting a rating works using CSS pseudo-classes or JavaScript listeners.

Client-Side Functionality Demo :

This section proves your JavaScript works.

Input Validation :

Required Fields: Attempt to submit the form without filling a required field. Show the native HTML5 validation messages (e.g., "Please fill out this field") and/or any custom JavaScript validation logic you added (e.g., checking for a valid email format).

Real-time Feedback: If you have any real-time effects (like character count on the <textarea>), demonstrate those.

Submission Process :

Fill the form completely: Enter sample data (e.g., Name: John Doe, Email: test@example.com, Feedback: "Love the new layout!").

Click "Submit Feedback":

Show the event handler: Explain (verbally) that the JavaScript addEventListener('submit', ...) function is triggered.

Prevent Default: Mention that event.preventDefault() is used to stop a page refresh.

PROJECT REPORT:

Key Evaluation Areas & Feedback Questions :

Organize your form or system into the following key feedback categories, each with suggested question types.

Structure & Organization :

- Is the report well-structured and logically organized? (Yes/No)
- Rate the flow and coherence of sections. (Scale: 1 = Poor → 5 = Excellent)
- Are headings and subheadings used effectively?

Clarity & Language :

- Is the writing clear and concise?
- Were any parts difficult to understand?
- Does the report use appropriate technical or non-technical language?

Content Completeness :

- Does the report cover all key components (e.g. problem statement, methodology, results)?
- Is there enough technical detail to understand the implementation?
- Are assumptions, limitations, or future work clearly stated?

Visuals & Supporting Materials :

- Are diagrams, tables, screenshots, or charts clear and relevant?
- Do they support and enhance the written content?

Overall Quality :

- Overall, how would you rate the quality of the report? (Scale: 1–5)
- Does the report effectively communicate the essence of the project

PROGRAM :

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Simple Feedback System</title>

  <style>
    body {
font-family: Arial, sans-serif;
background-color: #f4f4f9;
display: flex;
```

```
 justify-content: center;
align-items: center;
min-height: 100vh;
margin: 0;
}

 .container {
 background: #fff;
padding: 30px;
border-radius: 8px;
box-shadow: 0 4px 12px rgba(0, 0, 0, 0.1);
width: 100%;
max-width: 450px;
}

 h1 {
 text-align: center;
color: #333;
margin-bottom: 20px;
}

 label {
 display: block;
margin-bottom: 5px;
font-weight: bold;
color: #555;
}

 input[type="text"],
 input[type="email"],
textarea {
width: 100%;
padding: 10px;
margin-bottom: 20px;
border: 1px solid #ccc;
border-radius: 4px;
box-sizing: border-box;
 }

 textarea {
 resize: vertical;
}
```

```
.rating-group {
margin-bottom: 20px;
padding: 10px;
border: 1px solid #eee;
border-radius: 4px;
}

 .rating-group input[type="radio"] {
 margin-right: 5px;
}

 .rating-group label {
 font-weight: normal;
display: inline-block;
margin-right: 15px;
color: #666;
}

 button {
 background-color: #007bff;
color: white;
padding: 12px 20px;
border: none;
border-radius: 4px;
cursor: pointer;
font-size: 16px;
width: 100%;
transition: background-color 0.3s;
}

 button:hover {
 background-color: #0056b3;
}
</style>
</head>
<body>

 <div class="container">
 <h1>Share Your Feedback</h1>

 <form id="feedbackForm" action="#">

 <label for="name">Name :</label>
 <input type="text" id="name" name="name">

 <label for="email">Email :</label>
 <input type="email" id="email" name="email">
```

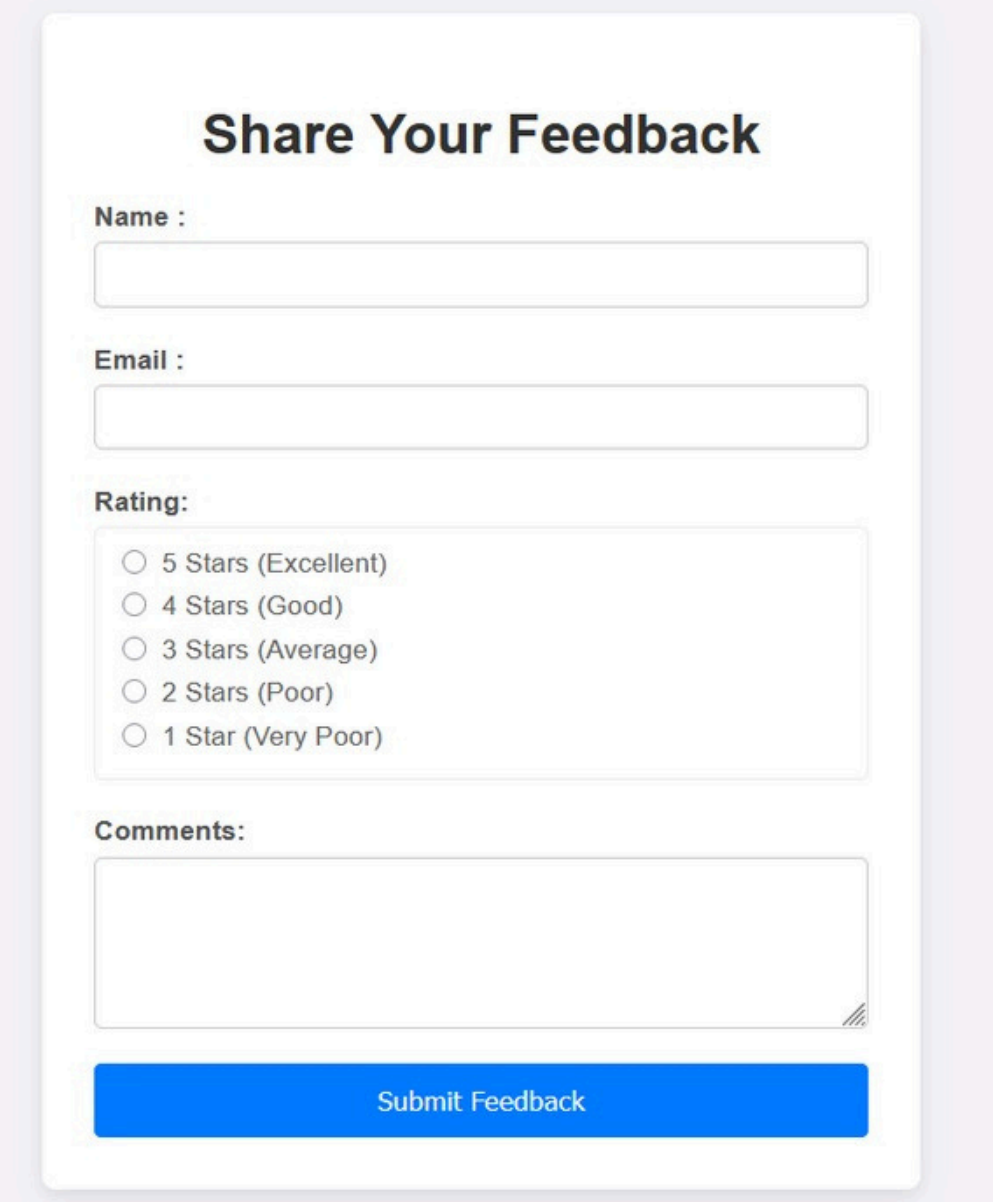
```
<label for="rating">Rating:</label>
<div class="rating-group">
<input type="radio" id="r5" name="rating" value="5" required>
<label for="r5">5 Stars (Excellent)</label><br>
<input type="radio" id="r4" name="rating" value="4">
<label for="r4">4 Stars (Good)</label><br>
<input type="radio" id="r3" name="rating" value="3">
<label for="r3">3 Stars (Average)</label><br>
<input type="radio" id="r2" name="rating" value="2">
<label for="r2">2 Stars (Poor)</label><br>
<input type="radio" id="r1" name="rating" value="1">
<label for="r1">1 Star (Very Poor)</label>
</div>

 <label for="comments">Comments:</label>
 <textarea id="comments" name="comments" rows="5" required></textarea>

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

 <script>
 document.getElementById('feedbackForm').addEventListener('submit', function(event) {
// Stop the form from performing the default submission (page reload)
event.preventDefault();

 // Collect data
 const name = document.getElementById('name').value;
const email = document.getElementById('email').value;
const comments = document.getElementById('comments').value;

 const selectedRatingElement = document.querySelector('input[name="rating"]:checked');
 const rating = selectedRatingElement ? selectedRatingElement.value + ' Stars' : 'Not Rated';

 // Prepare the collected feedback data
 const feedbackData = {
 Name: name || 'Anonymous',
 Email: email || 'N/A',
 Rating: rating,
 Comments: comments
 };
```

```
// Display the collected data in an alert box for demonstration
alert('Thank you for your feedback!\n\nCollected Data:\n' + JSON.stringify(feedbackData, null, 2));

 // In a real application, you would replace the alert above with code
 // to send this data to a server or a third-party service (e.g., Formspree).

 // Clear the form
 this.reset();
});
</script>

</body>
</html>
```

Screenshots / API Documentation :

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Simple Feedback System</title>

    <style>
        body {
            font-family: Arial, sans-serif;
            background-color: #f4f4f9;
            display: flex;
            justify-content: center;
            align-items: center;
            min-height: 100vh;
            margin: 0;
        }

        .container {
            background: #fff;
            padding: 30px;
            border-radius: 8px;
            box-shadow: 0 4px 12px rgba(0, 0, 0, 0.1);
            width: 100%;
            max-width: 450px;
        }

        h1 {
            text-align: center;
            color: #333;
            margin-bottom: 20px;
        }

        label {
            display: block;
            margin-bottom: 5px;
            font-weight: bold;
            color: #555;
```

OUTPUT :


Share Your Feedback
Name :
Email :
Rating:
5 Stars (Excellent)
4 Stars (Good)
3 Stars (Average)
2 Stars (Poor)
1 Star (Very Poor)
Comments:
Submit Feedback

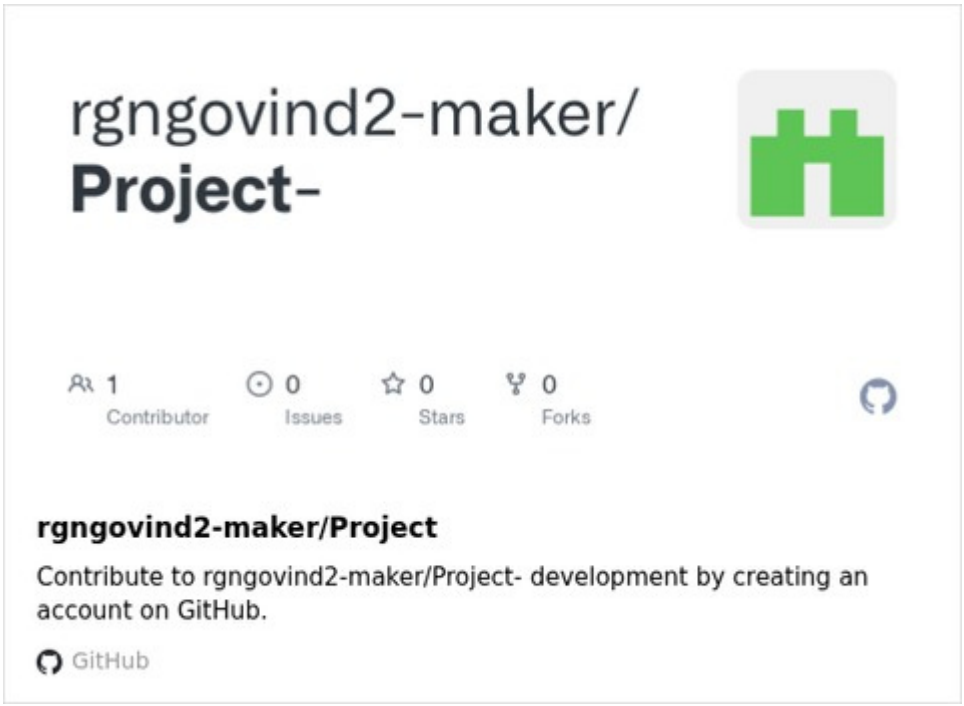
Challenges :

Clarity :
Arethechallenges described clearly and understandably?
Relevance :
Dothechallenges relate directly to the project goals and scope?
Completeness :
Areallsignificant challenges encountered during the project included?
Specificity :
Arethechallenges detailed with specific examples rather than vague statements?
Impact :
Istheimpactofeach challenge on the project clearly explained?
Complexity :
Arethechallenges described at an appropriate technical depth?
Prioritization :
Arethemostcritical challenges identified and emphasized?
Context :
Isthereenough background or context to understand why the challenge occurred?
Uniqueness :
Areanyunique or unexpected challenges described?
Presentation :
Isthechallenges section well-organized and easy to navigate?

Solutions :

Effectiveness :
Dothesolutionseffectively address the challenges described?
Clarity :
Arethesolutionsexplained clearly with step-by-step details?
Feasibility :
Arethesolutionspractical and realistically implementable?
Innovation :
Dothesolutionsshowcreativity or innovative thinking?
Technical Accuracy :
Aretechnicalaspectsofthe solutions sound and well-supported?
Alternative Approaches :
Arealternativesolutionsor backup plans discussed?
Lessons Learned :
Dothesolutionsreflectlessons learned from trial and error?
Resources Used :
Arethetools,technologies, or resources used to implement the solutions described?
Results :
Areoutcomesorimprovements resulting from the solutions clearly stated?
Presentation & Flow :
Isthesolutionssectionlogically organized and easy to follow?

GitHub README & Setup Guide :



Final Submission :