FULL STACK DEVELOPMENT

CIE-1

Aim:

Develop a Live Event Feedback Dashboard for a virtual event platform. The dashboard must display a personalized greeting using the participant's first name and surname. It should enable participants to submit real-time feedback on the session's quality, offering options such as Excellent, Good, Average, and Poor. The interface must also show the current time, updating every second. A submission counter is required to track the number of feedback entries made by the participant. The counter should include controls to increment, decrement, reset, and increment by five. Additionally, the system should simulate feedback from other users by randomly increasing the count of one feedback category every two seconds.

1. Greeting Section:

- Two input boxes for the participant to enter their first name and surname.
- Show a personalized greeting like: ➤ Welcome, Priyanka Padhiyar !

2. Live Clock:

- Show the current local date and time, updating every second using React concepts (no plain JS DOM).

3. Feedback Voting Panel:

- Four buttons labeled: Excellent, Good, Average, Poor.
- When the user clicks a feedback button, increment the corresponding feedback count and update the results immediately.

Display the total feedback count per category dynamically.

4. Simulated Real-Time Voting:

- Use a timer to simulate votes from other participants.
- Every 2 seconds, randomly increase the vote count for one of the four feedback options.

5. Counter Panel:

- Display a separate feedback counter for how many votes the user has personally submitted.
- Provide buttons: Increment, Decrement, Reset, and Increment by 5 to modify this counter.

App.js

```
import React, { useState, useEffect } from "react";

import "./App.css";


const App = () => {

  const [firstName, setFirstName] = useState("");

  const [surname, setSurname] = useState("");
  const [currentTime, setCurrentTime] = useState(new Date());
  const [feedback, setFeedback] = useState({
    Excellent: 0,
    Good: 0,
    Average: 0,
    Poor: 0,
  });
  const [userCount, setUserCount] = useState(0);

  useEffect(() => {
    const clock = setInterval(() => setCurrentTime(new Date()), 1000);
    return () => clearInterval(clock);
  }, []);

  useEffect(() => {
    const options = ["Excellent", "Good", "Average", "Poor"];
    const interval = setInterval(() => {
      const random = options[Math.floor(Math.random() * options.length)];
      setFeedback((prev) => ({
        ...prev,
        [random]: prev[random] + 1,
      }));
    }, 2000);
    return () => clearInterval(interval);
  }, []);

  const handleVote = (type) => {
    setFeedback((prev) => ({
      ...prev,
      [type]: prev[type] + 1,
    }));
    setUserCount((prev) => prev + 1);
  };

  return (
    <div className="container">
      <h1 className="main-title">Live Event Feedback Dashboard</h1>

      <div className="section card">
        <h2>Participant Details</h2>
        <div className="input-row">
          <input
            type="text"
            placeholder="First Name"
            value={firstName}
            onChange={(e) => setFirstName(e.target.value)}
          />
          <input
            type="text"
            placeholder="Surname"
            value={surname}
            onChange={(e) => setSurname(e.target.value)}
          />
        </div>
        {firstName && surname && (
          <p className="greeting">Welcome, {firstName} {surname}!</p>
        )}
      </div>

      <div className="section card">
```

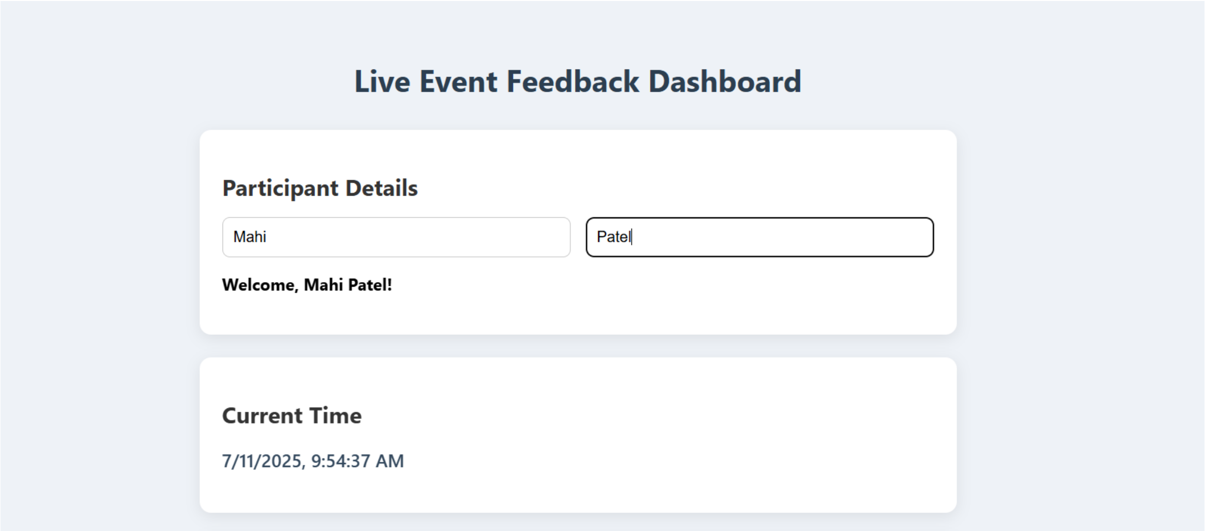
```
        <h2>Current Time</h2>
        <p className="clock">{currentTime.toLocaleString()}</p>
      </div>

      <div className="section card">
        <h2>Submit Feedback</h2>
        <div className="button-group">
          {["Excellent", "Good", "Average", "Poor"].map((option) => (
            <button key={option} onClick={() => handleVote(option)}>
              {option}
            </button>
          ))}
        </div>
      </div>

      <div className="section card">
        <h2>Feedback Summary</h2>
        <div className="summary-grid">
          {Object.entries(feedback).map(([key, value]) => (
            <div className="summary-box" key={key}>
              <h3>{key}</h3>
              <p>{value}</p>
            </div>
          ))}
        </div>
      </div>

      <div className="section card">
        <h2>Feedback Count</h2>
        <p className="counter">{userCount}</p>
        <div className="button-group small">
          <button onClick={() => setUserCount(userCount + 1)}>+1</button>
          <button onClick={() => setUserCount(userCount - 1)} disabled={userCount === 0}>-1</button>
          <button onClick={() => setUserCount(0)}>Reset</button>
          <button onClick={() => setUserCount(userCount + 5)}>+5</button>
        </div>
      </div>
    </div>
  );
};

export default App;
```

App.css

```
body {
  margin: 0;
  padding: 0;
  background: #eef2f7;
  font-family: 'Segoe UI', sans-serif;
}

.container {
  max-width: 800px;
  margin: 2rem auto;
  padding: 1rem;
}
```

```
.main-title {
  text-align: center;
  font-size: 2rem;
  color: #2c3e50;
  margin-bottom: 2rem;
}

.card {
  background: white;
  padding: 1.5rem;
  border-radius: 12px;
  box-shadow: 0 4px 15px rgba(0, 0, 0, 0.08);
  margin-bottom: 1.5rem;
  transition: all 0.3s ease;
}

.section h2 {
  margin-bottom: 1rem;
  color: #333;
}

.input-row {
  display: flex;
  gap: 1rem;
  flex-wrap: wrap;
}

.input-row input {
  flex: 1;
  padding: 0.7rem;
  border: 1px solid #ccc;
  border-radius: 8px;
  font-size: 1rem;
}

.greeting {
  margin-top: 1rem;
  font-size: 1.1rem;
  color: black;
  font-weight: bold;
}

.clock {
  font-size: 1.2rem;
  color: #34495e;
  font-weight: 500;
}

.button-group {
  display: flex;
  gap: 1rem;
  flex-wrap: wrap;
}

.button-group button {
  flex: 1 1 40%;
  padding: 0.8rem 1rem;
  font-weight: bold;
  border: none;
  border-radius: 8px;
  background: linear-gradient(135deg, #b358da, #9b59b6);
  color: white;
  cursor: pointer;
  transition: transform 0.2s ease;
}

.button-group button:hover {
  transform: scale(1.05);
}

.button-group.small button {
  flex: 1;
  background: #b358da;
}

.button-group.small button:hover {
  background: #b358da;
}

.button-group.small button:disabled {
  background: #bdc3c7;
  cursor: not-allowed;
}

.summary-grid {
  display: grid;
  grid-template-columns: repeat(2, 1fr);
  gap: 1rem;
}
```

```css
.summary-box {
  background: #dadfe4;
  /* border-left: 6px solid #8e44ad; */
  padding: 0.5rem;
  border-radius: 8px;
  text-align: center;
}

.summary-box h3 {
  margin: 0;
  color: #2c3e50;
}

.summary-box p {
  font-size: 1.5rem;
  font-weight: bold;
  color: #8e44ad;
  margin-top: 0.5rem;
}

.counter {
  font-size: 2rem;
  color: #8e44ad;
  font-weight: bold;
  margin: 1rem 0;
  text-align: center;
}
```



Live Event Feedback Dashboard

Participant Details

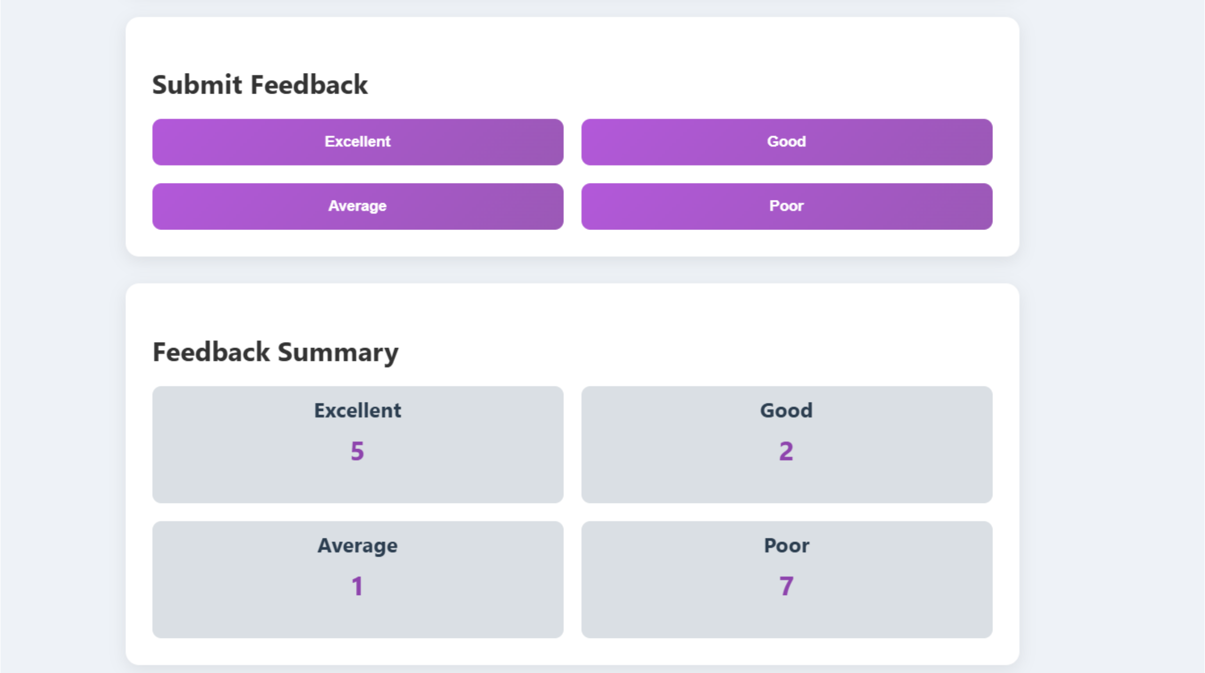
Mahi

Patel

Welcome, Mahi Patel!

Current Time

7/11/2025, 9:54:37 AM



Submit Feedback

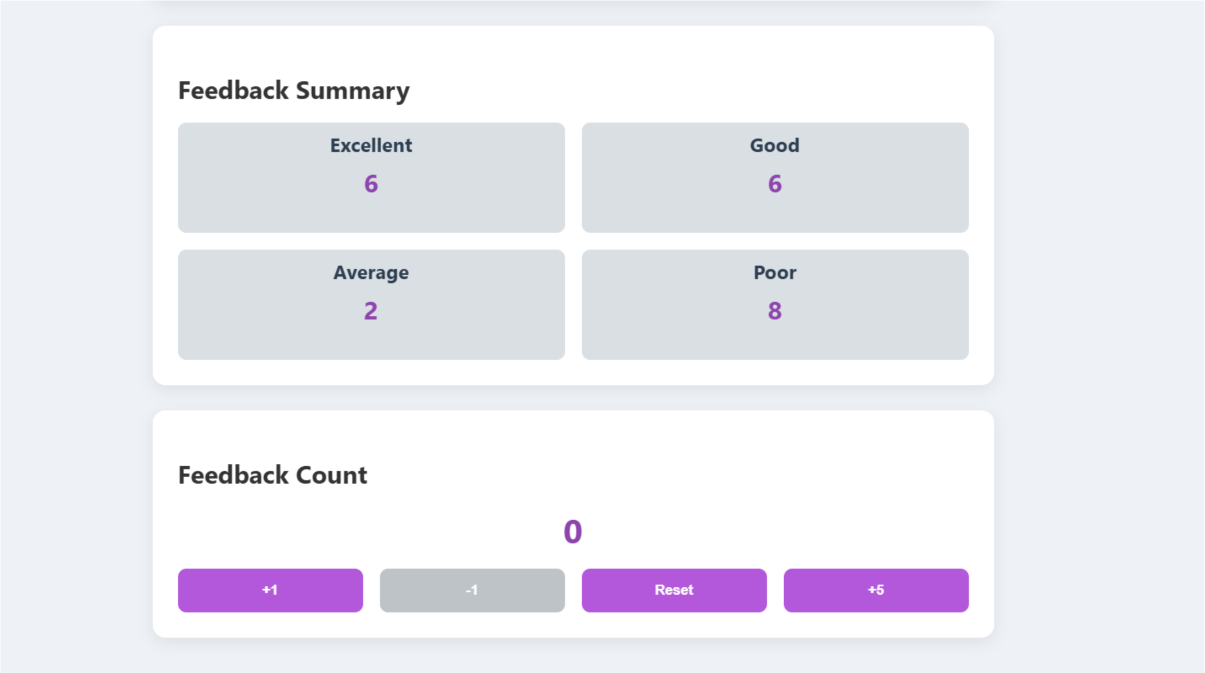
Feedback Summary

Excellent
5

Good
2

Average
1

Poor
7



Feedback Summary

Excellent
6

Good
6

Average
2

Poor
8

Feedback Count

0