

## **Full Stack Web 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.

**Code:**

1) App.jsx

```
import { useState } from 'react'
```

```
import reactLogo from './assets/react.svg'
```

```
import viteLogo from '/vite.svg'
```

```
import './App.css'
```

```
import Dashboard from './Component/Dashboard'
```

```
function App() {
```

```
  const [count, setCount] = useState(0)
```

```
  return (
```

```
    <>
```

```
      <Dashboard />
```

```
    </>
```

```
  )
```

```
}
```

```
export default App
```

## 2) Dashboard.jsx

```
import React, { useState, useEffect } from 'react';
import './styles/Dashboard.css'; // match your actual file name

export default function Dashboard() {
  const [firstName, setFirstName] = useState("");
  const [surname, setSurname] = useState("");
  const [time, setTime] = useState(new Date());

  const [feedbackCounts, setFeedbackCounts] = useState({
    Excellent: 0,
    Good: 0,
    Average: 0,
    Poor: 0
  });

  const [personalCounter, setPersonalCounter] = useState(0);

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

  useEffect(() => {
    const simulate = setInterval(() => {
      const categories = ['Excellent', 'Good', 'Average', 'Poor'];
```

```

    const random = categories[Math.floor(Math.random() *
categories.length)];

    setFeedbackCounts(prev => ({ ...prev, [random]: prev[random] + 1 }));

    }, 2000);

    return () => clearInterval(simulate);

  }, []);

```

```

const handleVote = (category) => {
  setFeedbackCounts(prev => ({ ...prev, [category]: prev[category] + 1 }));
  setPersonalCounter(c => c + 1);
};

```

```

return (
  <div className="dashboard-container">
    <div className="dashboard-card">
      <h1 className='text-2xl font-bold '>Live Event Feedback
Dashboard</h1>
      <div className="inputs ">
        <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 className="time">
  📅 {time.toLocaleDateString()} 🕒 {time.toLocaleTimeString()}
</div>

```

```

<div className="feedback-panel">
  <h2 className='text-xl font-bold'>Rate this session</h2>
  <div className="feedback-buttons">
    {[ 'Excellent', 'Good', 'Average', 'Poor' ].map(category => (
      <button key={category} onClick={() => handleVote(category)}>
        {category} ({feedbackCounts[category]})
      </button>
    ))}
  </div>
</div>

```

```

<div className="counter-panel">
  <h3 className='text-lg font-bold'>Your Submission Counter:
  {personalCounter}</h3>

```

```

    <div className="counter-buttons">
      <button onClick={() => setPersonalCounter(c => c +
1)}}>Increment</button>
      <button onClick={() => setPersonalCounter(c => Math.max(0, c -
1))}>Decrement</button>
      <button onClick={() => setPersonalCounter(0)}>Reset</button>
      <button onClick={() => setPersonalCounter(c => c + 5)}>Increment by
5</button>
    </div>
  </div>
</div>
</div>
);
}

```

### 3) Dashboard.css

```

/* Background and base styling */
.dashboard-container {
  min-height: 100vh;
  background: linear-gradient(90deg, #4f756d, #08c2d7);
  display: flex;
  align-items: center;
  justify-content: center;
  padding: 2rem;
  color: rgb(255, 255, 255);
}

```

```
.dashboard-card {  
  background: rgba(255, 255, 255, 0.2);  
  backdrop-filter: blur(10px);  
  padding: 2rem;  
  border-radius: 20px;  
  width: 100%;  
  max-width: 700px;  
  box-shadow: 0 8px 32px rgba(0,0,0,0.2);  
}
```

```
h1, h2, h3 {  
  margin-bottom: 1rem;  
}
```

```
.inputs {  
  display: flex;  
  flex-direction: column;  
  gap: 1rem;  
}
```

```
.inputs input {  
  padding: 0.6rem 1rem;  
  border-radius: 10px;  
  border: #e0e0e0 solid 2px;  
  outline: black solid 1px;
```

```
font-size: 1rem;  
width: 100%;  
}
```

```
.greeting {  
  margin-top: 0.5rem;  
  font-size: 1.2rem;  
}
```

```
.time {  
  margin-top: 1rem;  
  font-size: 1.1rem;  
}
```

```
/* Feedback panel */  
.feedback-panel {  
  margin-top: 2rem;  
}
```

```
.feedback-buttons {  
  display: flex;  
  flex-wrap: wrap;  
  gap: 1rem;  
}
```

```
.feedback-buttons button {
```



```
flex: 1;
min-width: 120px;
padding: 0.8rem 1rem;
border: none;
border-radius: 10px;
background: white;
color: black;
font-weight: bold;
cursor: pointer;
transition: background 0.3s;
}
```

```
.feedback-buttons button:hover {
  background: #e0e0e0;
}
```

```
/* Counter panel */
.counter-panel {
  margin-top: 2rem;
}
```

```
.counter-buttons {
  display: flex;
  flex-wrap: wrap;
  gap: 1rem;
  margin-top: 1rem;
```

```
}
```

```
.counter-buttons button {  
  flex: 1;  
  min-width: 120px;  
  padding: 0.8rem 1rem;  
  border: none;  
  border-radius: 10px;  
  background: #00c6ff;  
  color: white;  
  font-weight: bold;  
  cursor: pointer;  
  transition: background 0.3s;  
}
```

```
.counter-buttons button:hover {  
  background: #0072ff;  
}
```



**ScreenShot:**

### Live Event Feedback Dashboard

Shubh

Patel

► Welcome, Shubh Patel !

 11/07/2025  10:08:38

#### Rate this session

Excellent (21)

Good (37)

Average (22)

Poor (26)

Your Submission Counter: 7

Increment

Decrement

Reset

Increment by 5