

```

import React, { useState, useEffect } from 'react';
import { Card, CardContent } from '@components/ui/card';
import { Button } from '@components/ui/button';
import { LineChart, Line, XAxis, YAxis, CartesianGrid, Tooltip, Legend, ResponsiveContainer } from
'recharts';

const WeatherDashboard = () => {
  const [weatherData, setWeatherData] = useState([]);
  const [loading, setLoading] = useState(false);

  // Function to fetch data from Google Weather API or equivalent
  const fetchWeatherData = async () => {
    setLoading(true);
    try {
      const response = await fetch('https://api.open-
meteo.com/v1/forecast?latitude=37.7749&longitude=-122.4194&hourly=temperature_2m'); //
Example API for weather data
      if (!response.ok) {
        throw new Error(`HTTP error! status: ${response.status}`);
      }
      const data = await response.json();

      // Transform the data to match the format expected by the UI
      const transformedData = data.hourly.time.map((time, index) => ({
        timestamp: time,
        temperature: data.hourly.temperature_2m[index],
      }));

      setWeatherData(transformedData);
    } catch (error) {
      console.error('Error fetching weather data:', error);
    } finally {
      setLoading(false);
    }
  };

  useEffect(() => {
    fetchWeatherData();
    const interval = setInterval(fetchWeatherData, 60000); // Fetch data every minute
    return () => clearInterval(interval);
  }, []);

  return (
    <div className="min-h-screen p-4 bg-gradient-to-r from-blue-500 to-blue-300">
      <h1 className="text-2xl font-bold text-center text-white mb-4">Google-Based Weather
Monitoring Dashboard</h1>
      <div className="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-3 gap-4">
        {weatherData.map((entry, index) => (
          <Card key={index} className="bg-white shadow-md rounded-xl">
            <CardContent>
              <h2 className="text-xl font-semibold mb-2">Temperature Data</h2>
              <p className="text-gray-700">Temperature: {entry.temperature} °C</p>
              <p className="text-gray-700">Time: {new Date(entry.timestamp).toLocaleTimeString()}</p>
            </CardContent>
          </Card>
        ))}
      </div>
    </div>
  );
};

```

```

<div className="mt-8">
  <Card className="bg-white shadow-md rounded-xl">
    <CardContent>
      <h2 className="text-xl font-semibold text-center mb-4">Temperature Trends</h2>
      {weatherData.length > 0 ? (
        <ResponsiveContainer width="100%" height={300}>
          <LineChart data={weatherData}>
            <CartesianGrid strokeDasharray="3 3" />
            <XAxis dataKey="timestamp" tickFormatter={({tick}) => new Date(tick).toLocaleTimeString()}
              />
            <YAxis />
            <Tooltip labelFormatter={({label}) => new Date(label).toLocaleTimeString()} />
            <Legend />
            <Line type="monotone" dataKey="temperature" stroke="#8884d8" activeDot={{ r: 8 }} />
          </LineChart>
        </ResponsiveContainer>
      ) : (
        <p className="text-center text-gray-500">No data available</p>
      )}
    </CardContent>
  </Card>
</div>

<div className="flex justify-center mt-4">
  <Button onClick={fetchWeatherData} disabled={loading}>
    {loading ? 'Fetching...' : 'Refresh Data'}
  </Button>
</div>
</div>
);
};

export default WeatherDashboard;

```