Step 1: Set Up the React Project

1. **Open Terminal**: Open a terminal window on your computer.
2. **Create a New React App**: Run the following command to create a new React application:

   ```
   npx create-react-app weather-app
   ```

3. **Navigate to the Project Directory**:

   ```
   cd weather-app
   ```

Step 2: Install Required Dependencies

Install the necessary libraries for making API calls and visualizing data:

```
npm install axios chart.js react-chartjs-2
```

Step 3: Get Your OpenWeatherMap API Key

1. **Sign Up at OpenWeatherMap**: Go to the OpenWeatherMap website.
2. **Create an Account**: If you don't have one, create an account.
3. **Generate an API Key**: After logging in, go to the API section and generate a new API key.

Step 4: Create Folder Structure

Create the following folder structure within the `src` directory of your project:

```
weather-app/
└── src/
    ├── components/
    │   └── Weather.js
    └── services/
        └── weatherService.js
```

Step 5: Create the Weather Service

1. **Create `weatherService.js`**:
   - Navigate to the `src/services/` directory.
   - Create a new file named `weatherService.js`.
   - Add the following code to `weatherService.js`:

   ```
   javascript

   import axios from 'axios';

   const API_KEY = '9867fd8fc6dc694da17006f7274a85b9';  // Your actual API key
   const BASE_URL = 'https://api.openweathermap.org/data/2.5/weather';

   // Fetch weather data for a given city
   export const fetchWeather = async (city) => {
     try {
       const response = await
   axios.get(`${BASE_URL}?q=${city}&appid=${API_KEY}&units=metric`);
       return response.data;
     } catch (error) {
       if (error.response) {
         // Server responded with a status other than 200 range
         console.error("Error response:", error.response.data);
   ```

```
      throw new Error(error.response.data.message);  // Throw the actual error
message returned from API
    } else if (error.request) {
      // No response was received
      console.error("No response received:", error.request);
      throw new Error("No response received from the server");
    } else {
      // Something else caused the error
      console.error("Error:", error.message);
      throw new Error(error.message);
    }
  }
};
```

Step 6: Create the Weather Component

1. **Create `Weather.js`**:
   - Navigate to the `src/components/` directory.
   - Create a new file named `Weather.js`.
   - Add the following code to `Weather.js`:

```
javascript

import React, { useState } from 'react';
import { fetchWeather } from '../services/weatherService'; // Adjust the path if
needed
import { Line } from 'react-chartjs-2';
import { Chart, registerables } from 'chart.js';

// Register the required components for Chart.js
Chart.register(...registerables);

const Weather = () => {
  const [city, setCity] = useState('');
  const [weather, setWeather] = useState(null);
  const [error, setError] = useState('');

  const getWeather = async () => {
    try {
      const data = await fetchWeather(city);
      console.log("Weather data:", data);
      setWeather(data);
      setError('');  // Clear the error if successful
    } catch (error) {
      console.error("Error fetching weather:", error.message);
      setError(error.message);  // Display the actual error message to the user
      setWeather(null);
    }
  };

  const temperatureData = {
    labels: ['Today', 'Tomorrow', 'Day 3', 'Day 4', 'Day 5'], // Replace with
actual dates if available
    datasets: [
      {
        label: 'Temperature (°C)',
        data: [weather ? weather.main.temp : 0, 22, 21, 24, 23], // Use actual
data if available
        borderColor: 'rgba(75, 192, 192, 1)',
        backgroundColor: 'rgba(75, 192, 192, 0.2)',
        fill: true,
      },
    ],
  };
```

```
  return (
    <div className="container mt-5">
      <h1 className="text-center">Weather App</h1>
      <input
        type="text"
        className="form-control mb-3"
        placeholder="Enter city name"
        value={city}
        onChange={(e) => setCity(e.target.value)}
      />
      <button className="btn btn-primary" onClick={getWeather}>Get
Weather</button>
      {error && <div className="text-danger">{error}</div>}
      {weather && (
        <div className="mt-4">
          <h2>{weather.name}</h2>
          <p>Temperature: {weather.main.temp}°C</p>
          <Line data={temperatureData} />
        </div>
      )}
    </div>
  );
};

export default Weather;
```

Step 7: Update the Main Application File

1. **Update `App.js`**:
   - Open `src/App.js`.
   - Replace the existing code with the following:

```
javascript

import React from 'react';
import Weather from './components/Weather';

const App = () => {
  return (
    <div>
      <Weather />
    </div>
  );
};

export default App;
```

Step 8: Run the Application

1. **Start the Development Server**: In the terminal, run the following command to start the React application:

```
npm start
```

2. **Open in Browser**: Your React app should now be running locally at `http://localhost:3000`.

Step 9: Test the Application

1. **Enter a City Name**: In the input field, enter a valid city name (e.g., "London").
2. **Click "Get Weather"**: Click the button to fetch and display the weather data.

Final Notes

- Make sure to handle errors gracefully and check for the validity of the API key and city name.
- Customize the chart and app as needed.