**1. Understanding User Needs**

The first step was to understand the specific needs of different user groups: General, Farmers, Event Planners, and Travelers. Each group has unique requirements for weather data:

* **General Users**: Require a comprehensive overview of current weather conditions.
* **Farmers**: Need detailed information about rainfall, temperature, and humidity for crop management.
* **Event Planners**: Need to know about the likelihood of rain or extreme weather to make contingency plans.
* **Travelers**: Require information on weather conditions for safe and comfortable travel.

**2. Designing the Dashboard**

Based on the user needs, I designed a dashboard that includes the following components:

* **City Search Bar**: To allow users to enter a city name and get the weather forecast.
* **User Group Selection Buttons**: To switch between different user groups and their respective weather views.
* **Weather Information Display**: Showing temperature, humidity, rain description, wind speed, and an overall suitability message tailored to the user group.

**3. Fetching and Displaying Data**

I used the OpenWeatherMap API to fetch the weather data. This involved setting up the API call and handling the response to extract relevant data. I ensured that the data updates dynamically based on user input and selection.

**4. Handling Errors**

Error handling was implemented to display a user-friendly message when an invalid city name is entered or if there are issues with fetching data.

**5. Adding Forecast Feature**

To add a forecast feature, I extended the API call to include a three-day forecast. This involved processing the API response to extract and display weather data for the next three days, with the ability for users to manually change the date to view different forecasts.

Live link of Project Review

<https://weather-dashboard-task.netlify.app/>

### Step-by-Step Guide

#### 1. ****Set Up Your Development Environment****

1. **Install Node.js and npm**:
   * Download and install the latest version of Node.js from [nodejs.org](https://nodejs.org/).
2. **Install a Code Editor**:
   * Use a code editor like [Visual Studio Code](https://code.visualstudio.com/).

#### 2. ****Initialize Your React Project****

1. **Create a New React App**: Open your terminal or command prompt and run the following command:

bash

Copy code

npx create-react-app weather-dashboard

cd weather-dashboard

1. **Install Necessary Packages**: If you have additional packages to install, do so using npm or yarn. For example:

bash

Copy code

npm install

#### 3. ****Set Up Your Project Structure****

1. **Create Components**: Inside the src directory, create folders and files for your components. For example:

bash

Copy code

src/

├── components/

│ ├── Main.js

│ ├── WeatherData.js

│ ├── FarmerWeather.js

│ ├── EventPlannerWeather.js

│ ├── TravelerWeather.js

│ ├── search.svg

│ ├── external-link.svg

├── App.js

├── index.js

└── App.css

1. **Add Code to Components**: Add the code you shared for each component.

#### 4. ****Configure Your Application****

1. **Set Up API Key**: Add your OpenWeatherMap API key in Main.js.

#### 5. ****Run the Project Locally****

1. **Start the Development Server**:

bash

Copy code

npm start

This will start the development server, and you can view your app at http://localhost:3000.

#### 6. ****Deployment (Optional)****

If you want to deploy your application, you can use services like [Vercel](https://vercel.com/) or [Netlify](https://www.netlify.com/).

### Example Deployment Using Vercel

1. **Install Vercel CLI**:

bash

Copy code

npm install -g vercel

1. **Deploy to Vercel**: Inside your project directory, run:

bash

Copy code

vercel

Follow the prompts to deploy your application.

### Summary

1. Set up your development environment.
2. Initialize your React project and set up the structure.
3. Add your code and configure the application.
4. Run the project locally to test it.
5. (Optional) Deploy your application using Vercel or Netlify.