README:

Step 1: Introduction+

- A **Live Weather Dashboard** is a web-based application that displays **real-time weather data** (temperature, humidity, wind, etc.) for any city.
- It connects to a **weather API** (like OpenWeatherMap) to fetch current and forecasted weather information.

Step 2: Objective

- To design a dashboard that shows **live weather updates** for any city.
- To integrate a **real-time API** for accurate data.
- To create a **responsive and user-friendly interface** for easy access.

Step 3: Tools & Technologies Used

Category	**Technology**
Frontend	React.js, HTML, CSS, JavaScript
API	OpenWeatherMap API
Library	Axios (for fetching data)
Styling	Bootstrap / Tailwind CSS
IDE	Visual Studio Code
Version Control	Git & GitHub

Step 4: System Requirements

Software:

- Node.js, Browser, Internet Connection, VS Code

Hardware:

- Minimum 4 GB RAM
- Dual-core processor or higher

Step 5: Working Principle

1. User enters a **city name**.
2. The app sends a request to **OpenWeatherMap API**.
3. The API returns **JSON data** containing weather details.
4. The app displays data dynamically (temperature, humidity, condition).

Flow:
User → App (React) → API → Weather Data → Display

Step 6: Implementation Steps

1. **Create React project** using npx create-react-app.
2. **Get API key** from OpenWeatherMap.
3. **Set up .env file** to store the key securely.
4. **Create components:**
    - SearchBar.jsx → for city input
    - WeatherCard.jsx → for displaying results
5. **Use Axios** to call API and display data in UI.
6. **Add CSS** for design and responsiveness.

Step 7: Sample Output

Input:
City → *"Delhi"*

Displayed Output:

City: Delhi

Temperature: 30°C

Condition: Clear

Humidity: 40%

Wind: 2.5 m/s

(Include screenshots of your dashboard on this slide.)

Step 8: Advantages

- Provides **real-time weather updates** instantly.
- **Simple and user-friendly** interface.
- Works for **any city worldwide**.
- Great **learning project** for API integration and frontend development.

Step 9: Limitations

- Needs an **active internet connection**.
- Dependent on **external API** (may have request limits).
- **Free API** access may have fewer features (e.g., hourly forecast).

Step 10: Conclusion & Future Scope

Conclusion:
The Live Weather Dashboard effectively demonstrates **real-time data fetching and display** using React and APIs.

Future Enhancements:

- Add **5-day forecast charts**
- Include **location auto-detect**
- Support **dark/light mode**
- Add **AI-based weather tips**