EXPERIMENT 2

Aim: Weather App using React Hooks (useEffect, useContext, Custom Hooks)

Tools & Technologies:

- **React.js** (Frontend framework)
- **JavaScript (ES6+)**
- **React Router** (for navigation)
- **OpenWeather API** (for fetching live weather data)
- **CSS / Tailwind CSS** (for styling and responsiveness)

Prerequisites:

- Basic knowledge of **JavaScript and ES6+ features**
- Familiarity with **React.js fundamentals** (components, props, state)
- Understanding of **React Hooks** (useState, useEffect, useContext, custom hooks)
- API handling using fetch or axios
- Basic knowledge of **React Router** for navigation

Theory:

React Hooks are functions that let you use React features in functional components without writing class components.

- **useEffect**: Allows side effects like data fetching, DOM manipulation, or setting up timers.

- **useContext**: Provides a way to pass data across components without manually sending props at every level. It works like a global state accessible anywhere inside the provider.

- **Custom Hooks**: Let us extract and reuse component logic. Instead of repeating code for API calls or local storage, we can write a custom hook once and reuse it.

These hooks simplify code, reduce duplication, and improve maintainability by promoting reusability and separation of concerns.

Working & Functionalities (Use of Hooks in Project):

In this Weather App:

- **useEffect** is used to fetch weather data dynamically from the OpenWeather API whenever the user searches for a new city. It ensures the data updates whenever dependencies like the city name change.

- **useContext** is implemented for theme and global state sharing, which allows multiple components (like weather info, modal popup, and router navigation) to access shared data without repetitive prop drilling.

- **Custom Hook (useWeather)** was created to encapsulate the API call logic, error handling, and state management. This way, fetching weather data can be reused in different components without rewriting the logic.

30% Extra Implementation:

Beyond the aim of just implementing hooks, additional features were added:

1. **Routing with React Router** – Users can navigate to a new route where they get suggestions for “Best Places to Visit” in the searched city, depending on the weather.

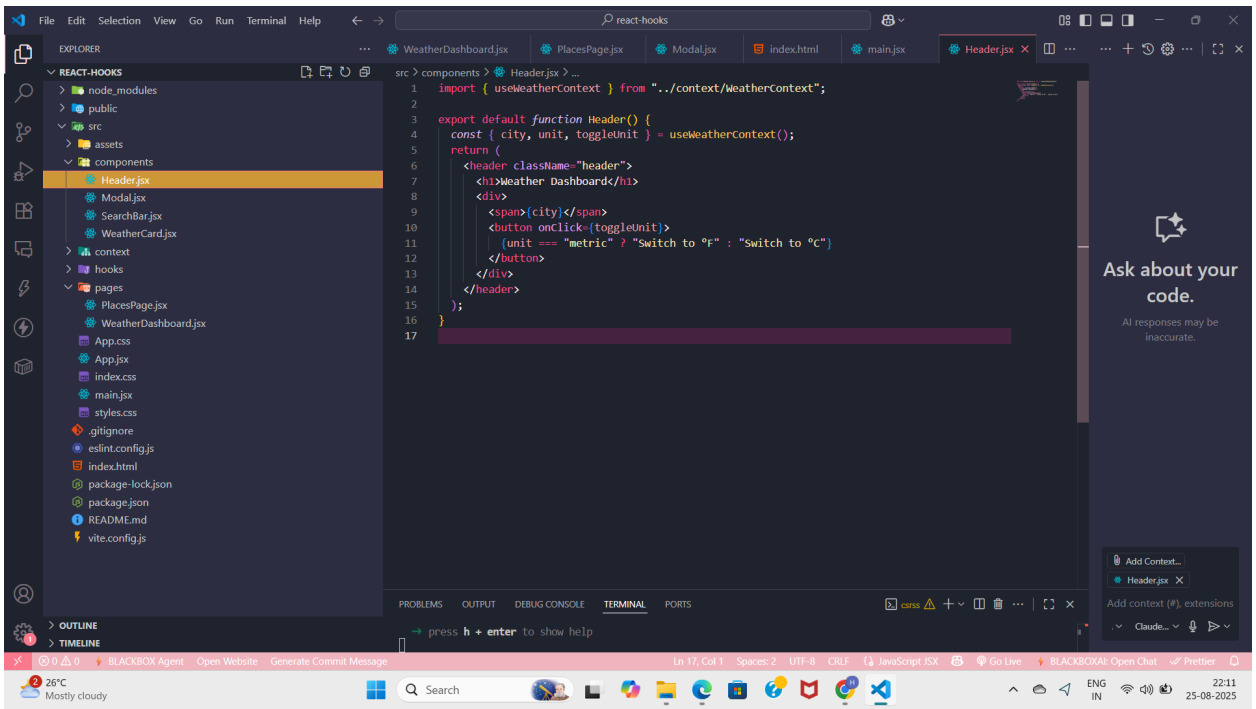
2. **Travel Booking Integration** – From the suggestions page, users are given a link to book tickets (taking them to a common booking website).

3. **Popup Modal** – A modal is displayed to show important travel/weather alerts, making the UI interactive.

4. **Search Option Enhancement** – The search bar allows users to explore weather conditions of any city in real-time.

5. **Responsive UI** – The application layout is styled and enhanced with CSS to be responsive and user-friendly.

Code:



File Edit Selection View Go Run Terminal Help

react-hooks

EXPLORER

REACT-HOOKS

- node_modules
- public
- src
  - assets
  - components
    - Header.jsx
    - Modal.jsx
    - SearchBar.jsx
    - WeatherCard.jsx
  - context
  - hooks
  - pages
    - PlacesPage.jsx
    - WeatherDashboard.jsx
  - App.css
  - App.jsx
  - index.css
  - main.jsx
  - styles.css
- .gitignore
- eslint.config.js
- index.html
- package-lock.json
- package.json
- README.md
- vite.config.js

OUTLINE

TIMELINE

WeatherDashboard.jsx | PlacesPage.jsx | Modal.jsx | index.html | main.jsx | Header.jsx

src > components > Modal.jsx > ...

```
import ReactDOM from "react-dom";

export default function Modal({ children, onClose }) {
  return ReactDOM.createPortal(
    <div className="modal-overlay" onClick={onClose}>
      <div className="modal" onClick={(e) => e.stopPropagation()}>
        <button className="close" onClick={onClose} aria-label="Close">×</button>
        {children}
      </div>
    </div>,
    document.getElementById("modal-root")
  );
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

→ press h + enter to show help

Ask about your code.

AI responses may be inaccurate.

Add Context...

Modal.jsx

Add context (#), extensions

Claude...

BLACKBOX Agent Open Website Generate Commit Message

Ln 14, Col 1 Spaces: 2 UTF-8 CRLF JavaScript JSX Go Live BLACKBOXAI: Open Chat Prettier

26°C Mostly cloudy

Search

ENG IN 22:11 25-08-2025

File Edit Selection View Go Run Terminal Help

react-hooks

EXPLORER

REACT-HOOKS

- node_modules
- public
- src
  - assets
  - components
    - Header.jsx
    - Modal.jsx
    - SearchBar.jsx
    - WeatherCard.jsx
  - context
  - hooks
  - pages
    - PlacesPage.jsx
    - WeatherDashboard.jsx
  - App.css
  - App.jsx
  - index.css
  - main.jsx
  - styles.css
- .gitignore
- eslint.config.js
- index.html
- package-lock.json
- package.json
- README.md
- vite.config.js

OUTLINE

TIMELINE

SearchBar.jsx | WeatherCard.jsx | App.jsx | WeatherDashboard.jsx | PlacesPage.jsx | Modal

src > components > SearchBar.jsx > ...

```
import { useState } from "react";
import { useCity } from "../context/CityContext.jsx";

export default function SearchBar() {
  const { setCity } = useCity();
  const [input, setInput] = useState("");

  const handleSubmit = (e) => {
    e.preventDefault();
    const q = input.trim();
    if (!q) return;
    setCity(q);
    setInput("");
  };

  return (
    <form className="search" onSubmit={handleSubmit}>
      <input
        className="input"
        type="text"
        placeholder="Search city (e.g., Mumbai, London)…"
        value={input}
        onChange={(e) => setInput(e.target.value)}
      />
      <button className="btn" type="submit">Search</button>
    </form>
  );
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

→ press h + enter to show help

Ask about your code.

AI responses may be inaccurate.

Add Context...

SearchBar.jsx

Add context (#), extensions

Claude...

BLACKBOX Agent Open Website Generate Commit Message

Ln 29, Col 1 Spaces: 2 UTF-8 CRLF JavaScript JSX Go Live BLACKBOXAI: Open Chat Prettier

26°C Mostly cloudy

Search

ENG IN 22:12 25-08-2025

File Edit Selection View Go Run Terminal Help

react-hooks

EXPLORER

REACT-HOOKS

- node_modules
- public
- src
  - assets
  - components
    - Header.jsx
    - Modal.jsx
    - SearchBar.jsx
    - WeatherCard.jsx
  - context
  - hooks
  - pages
    - PlacesPage.jsx
    - WeatherDashboard.jsx
  - App.css
  - App.jsx
  - index.css
  - main.jsx
  - styles.css
- .gitignore
- eslint.config.js
- index.html
- package-lock.json
- package.json
- README.md
- vite.config.js

OUTLINE

TIMELINE

SearchBar.jsx | WeatherCard.jsx | App.jsx | WeatherDashboard.jsx | PlacesPage.jsx | Modal

src > components > WeatherCard.jsx > ...

```jsx
import { useTemperatureConversion } from "../hooks/useTemperatureConversion";

export default function WeatherCard({ weather }) {
  if (!weather) return <div class="card"><p>Loading weather…</p></div>;

  const { temperature, unit, toggleUnit } = useTemperatureConversion(
    weather.main.temp
  );

  const iconCode = weather.weather[0].icon;
  const iconUrl = `https://openweathermap.org/img/wn/${iconCode}@2x.png`;

  return (
    <div class="card">
      <h2 class="city">{weather.name}</h2>
      <img class="icon" src={iconUrl} alt={weather.weather[0].description} />
      <p class="temp">
        {temperature}°{unit}
      </p>
      <p class="desc">{weather.weather[0].description}</p>

      <div class="metrics">
        <div><span>Feels like</span><strong>{Math.round(weather.main.feels_like)}°C</strong>
        <div><span>Humidity</span><strong>{weather.main.humidity}%</strong></div>
        <div><span>Wind</span><strong>{weather.wind?.speed ?? 0} m/s</strong></div>
      </div>

      <button class="btn secondary full" onClick={toggleUnit}>
        Show in °{unit === "C" ? "F" : "C"}
      </button>

      <p class="updated">Last updated: {new Date(weather.dt * 1000).toLocaleTimeString()
    </div>
```

PROBLEMS | OUTPUT | DEBUG CONSOLE | TERMINAL | PORTS

→ press h + enter to show help

Ln 36, Col 1 | Spaces: 2 | UTF-8 | CRLF | JavaScript JSX

Ask about your code.

AI responses may be inaccurate.

Add Context...

WeatherCard.jsx

Add context (#), extensions

Claude...

Rain warning
In effect

Search

ENG IN 22:12 25-08-2025

File Edit Selection View Go Run Terminal Help

react-hooks

EXPLORER

REACT-HOOKS

- node_modules
- public
- src
  - assets
  - components
    - Header.jsx
    - Modal.jsx
    - SearchBar.jsx
    - WeatherCard.jsx
  - context
    - CityContext.jsx
    - WeatherContext.jsx
  - hooks
    - useLocalStorage.js
    - useTemperatureConversion.js
    - useWeather.js
  - pages
    - PlacesPage.jsx
    - WeatherDashboard.jsx
  - App.css
  - App.jsx
  - index.css
  - main.jsx
  - styles.css
- .gitignore
- eslint.config.js
- index.html
- package-lock.json
- package.json
- README.md

OUTLINE

TIMELINE

CityContext.jsx | useTemperatureConversion.js | SearchBar.jsx | WeatherCard.jsx | App.jsx

src > context > CityContext.jsx > ...

```jsx
import { createContext, useContext, useState } from "react";

const CityContext = createContext();

export function CityProvider({ children }) {
  const [city, setCity] = useState("Mumbai"); // ✅ default Mumbai
  return (
    <CityContext.Provider value={{ city, setCity }}>
      {children}
    </CityContext.Provider>
  );
}

export function useCity() {
  const ctx = useContext(CityContext);
  if (!ctx) throw new Error("useCity must be used within CityProvider");
  return ctx;
}
```

PROBLEMS | OUTPUT | DEBUG CONSOLE | TERMINAL | PORTS

→ press h + enter to show help

Ln 19, Col 1 | Spaces: 2 | UTF-8 | CRLF | JavaScript JSX

Ask about your code.

AI responses may be inaccurate.

Add Context...

CityContext.jsx

Add context (#), extensions

Claude...

Rain warning
In effect

Search

ENG IN 22:12 25-08-2025

File Edit Selection View Go Run Terminal Help

react-hooks

EXPLORER

REACT-HOOKS

- node_modules
- public
- src
  - assets
  - components
    - Header.jsx
    - Modal.jsx
    - SearchBar.jsx
    - WeatherCard.jsx
  - context
    - CityContext.jsx
    - WeatherContext.jsx
  - hooks
    - useLocalStorage.js
    - useTemperatureConversion.js
    - useWeather.js
  - pages
    - PlacesPage.jsx
    - WeatherDashboard.jsx
  - App.css
  - App.jsx
  - index.css
  - main.jsx
  - styles.css
  - .gitignore
  - eslint.config.js
  - index.html
  - package-lock.json
  - package.json
  - README.md

OUTLINE

TIMELINE

WeatherContext.jsx | useLocalStorage.js | useWeather.js | CityContext.jsx | useTemperatureConversion

src > context > WeatherContext.jsx > ...

```
import { createContext, useContext } from "react";
import useLocalStorage from "../hooks/useLocalStorage";

const WeatherContext = createContext();

export function WeatherProvider({ children }) {
  const [city, setCity] = useLocalStorage("city", "Mumbai");
  const [unit, setUnit] = useLocalStorage("unit", "metric"); // metric = °C

  const value = {
    city,
    unit,
    setCity,
    toggleUnit: () => setUnit(u => (u === "metric" ? "imperial" : "metric"))
  };

  return (
    <WeatherContext.Provider value={value}>
      {children}
    </WeatherContext.Provider>
  );
}

export function useWeatherContext() {
  return useContext(WeatherContext);
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

→ press h + enter to show help

Ask about your code.

AI responses may be inaccurate.

Add Context...

WeatherContext.jsx

Add context (#), extensions

Claude...

Ln 27, Col 1 Spaces: 2 UTF-8 CRLF JavaScript JSX Go Live BLACKBOXAI: Open Chat Prettier

Rain warning In effect

Search

ENG IN 22:12 25-08-2025

File Edit Selection View Go Run Terminal Help

react-hooks

EXPLORER

REACT-HOOKS

- node_modules
- public
- src
  - assets
  - components
    - Header.jsx
    - Modal.jsx
    - SearchBar.jsx
    - WeatherCard.jsx
  - context
    - CityContext.jsx
    - WeatherContext.jsx
  - hooks
    - useLocalStorage.js
    - useTemperatureConversion.js
    - useWeather.js
  - pages
    - PlacesPage.jsx
    - WeatherDashboard.jsx
  - App.css
  - App.jsx
  - index.css
  - main.jsx
  - styles.css
  - .gitignore
  - eslint.config.js
  - index.html
  - package-lock.json
  - package.json
  - README.md

OUTLINE

TIMELINE

sion.js | SearchBar.jsx | WeatherCard.jsx | App.jsx | WeatherDashboard.jsx | PlacesPage.jsx

src > pages > PlacesPage.jsx > ...

```
export default function PlacesPage() {

  useEffect(() => {
    if (!city) {
      navigate("/");
      return;
    }

    // Simple recommendation rules by temperature (°C)
    if (temp < 20) {
      setPlaces([
        "City Heritage Walk",
        "Art Museums",
        "Historic Forts",
        "Local Cafés with Views",
      ]);
    } else if (temp >= 20 && temp <= 30) {
      setPlaces([
        "Waterfront Promenades",
        "Urban Parks",
        "Popular Beaches",
        "Street Food Spots",
      ]);
    } else {
      setPlaces([
        "Aquariums & Planetariums",
        "Indoor Malls",
        "Cultural Centers",
        "Cave/Temple Excursions",
      ]);
    }

    // You could also incorporate condition (rainy -> more indoor)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

→ press h + enter to show help

Ask about your code.

AI responses may be inaccurate.

Add Context...

PlacesPage.jsx

Add context (#), extensions

Claude...

Ln 91, Col 1 Spaces: 2 UTF-8 CRLF JavaScript JSX Go Live BLACKBOXAI: Open Chat Prettier

Rain warning In effect

Search

ENG IN 22:13 25-08-2025

File Edit Selection View Go Run Terminal Help

react-hooks

EXPLORER

REACT-HOOKS

- node_modules
- public
- src
  - assets
  - components
    - Header.jsx
    - Modal.jsx
    - SearchBar.jsx
    - WeatherCard.jsx
  - context
    - CityContext.jsx
    - WeatherContext.jsx
  - hooks
    - useLocalStorage.js
    - useTemperatureConversion.js
    - useWeather.js
  - pages
    - PlacesPage.jsx
    - WeatherDashboard.jsx
  - App.css
  - App.jsx
  - index.css
  - main.jsx
  - styles.css
  - .gitignore
  - eslint.config.js
  - index.html
  - package-lock.json
  - package.json
  - README.md

OUTLINE

TIMELINE

sion.js | SearchBar.jsx | WeatherCard.jsx | App.jsx | WeatherDashboard.jsx | PlacesPage.jsx

src > pages > WeatherDashboard.jsx > ...

```jsx
export default function WeatherDashboard() {

  // 🌤 Dynamic background class based on condition
  let bgClass = "bg-default";
  if (weather) {
    const cond = weather.weather[0].main.toLowerCase();
    if (cond.includes("rain")) bgClass = "bg-rainy";
    else if (cond.includes("cloud")) bgClass = "bg-cloudy";
    else if (cond.includes("clear")) bgClass = "bg-sunny";
    else bgClass = "bg-night";
  }

  return (
    <main className={`app-shell ${bgClass}`}>
      <section className="panel">
        <h1 className="title">Weather Dashboard</h1>
        <SearchBar />

        {error && <p className="error">{error}</p>}
        <WeatherCard weather={weather} />

        {/* Always show a “best places” link for the current city */}
        {weather && (
          <Link
            to="/places"
            state={{
              city: weather.name,
              temp: weather.main.temp,
              condition: weather.weather[0].main,
            }}
            className="btn primary mt"
          >
            Best places to visit in {weather.name}
```

PROBLEMS | OUTPUT | DEBUG CONSOLE | TERMINAL | PORTS

→ press h + enter to show help

Ln 48, Col 1 | Spaces: 2 | UTF-8 | CRLF | JavaScript JSX | Go Live | BLACKBOXAI: Open Chat | Prettier

Ask about your code.

AI responses may be inaccurate.

Add Context...

WeatherDashboard.js

Add context (#), extensions

Claude...

Rain warning
In effect

Search

ENG IN | 22:13 25-08-2025

File Edit Selection View Go Run Terminal Help

react-hooks

EXPLORER

REACT-HOOKS

- node_modules
- public
- src
  - assets
  - components
    - Header.jsx
    - Modal.jsx
    - SearchBar.jsx
    - WeatherCard.jsx
  - context
    - CityContext.jsx
    - WeatherContext.jsx
  - hooks
    - useLocalStorage.js
    - useTemperatureConversion.js
    - useWeather.js
  - pages
    - PlacesPage.jsx
    - WeatherDashboard.jsx
  - App.css
  - App.jsx
  - index.css
  - main.jsx
  - styles.css
  - .gitignore
  - eslint.config.js
  - index.html
  - package-lock.json
  - package.json
  - README.md

OUTLINE

TIMELINE

useWeather.js | CityContext.jsx | useTemperatureConversion.js | SearchBar.jsx | WeatherCard.jsx

src > hooks > useWeather.js > useWeather > useEffect() callback

```jsx
import { useState, useEffect } from 'react';

export function useWeather(city) { // ✅ named export
  const [weather, setWeather] = useState(null);
  const [error, setError] = useState(null);

  useEffect(() => {
    if (!city) return;

    fetch(`https://api.openweathermap.org/data/2.5/weather?q=${city}&appid=ee435ac7b69e4614
      .then((res) => res.json())
      .then((data) => {
        if (data.cod === 200) {
          setWeather(data);
          setError(null);
        } else {
          setError(data.message);
          setWeather(null);
        }
      })
      .catch((err) => {
        setError(err.message);
        setWeather(null);
      });
  }, [city]);

  return { weather, error };
}
```

PROBLEMS | OUTPUT | DEBUG CONSOLE | TERMINAL | PORTS

→ press h + enter to show help

Ln 10, Col 108 | Spaces: 2 | UTF-8 | CRLF | JavaScript | Go Live | BLACKBOXAI: Open Chat | Prettier

Ask about your code.

AI responses may be inaccurate.

Add Context...

useWeather.js

Add context (#), extensions

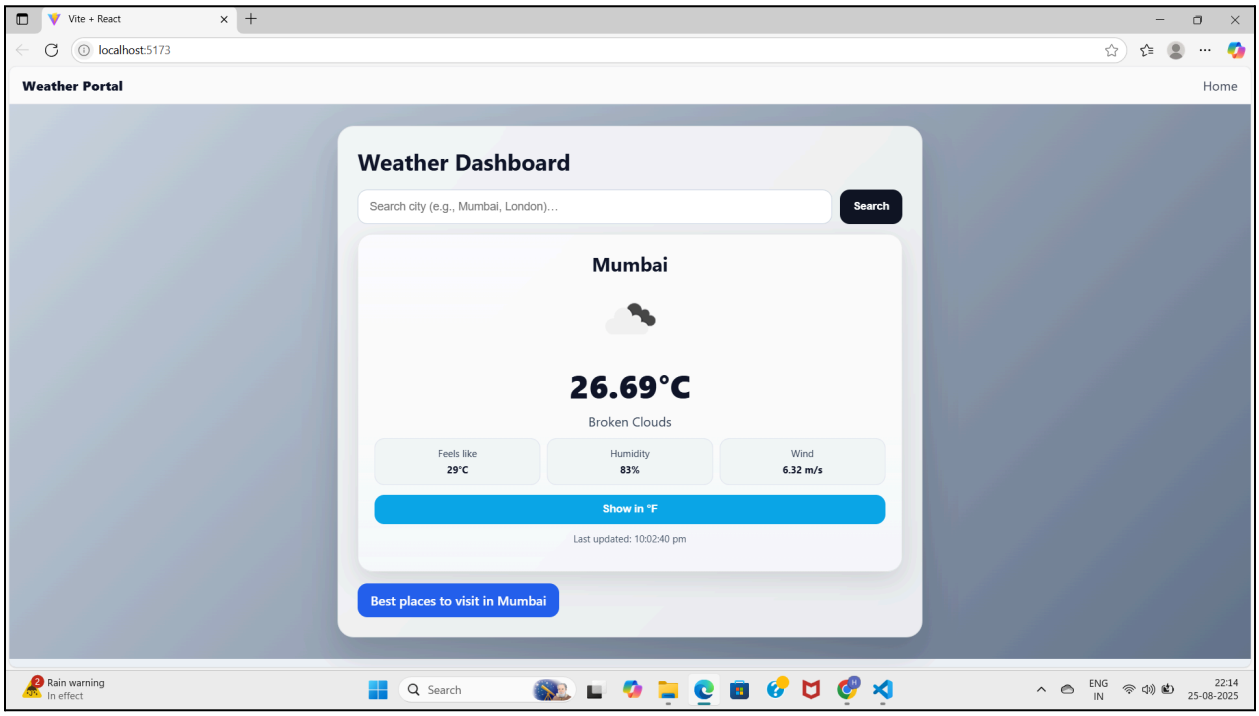
Claude...

Rain warning
In effect

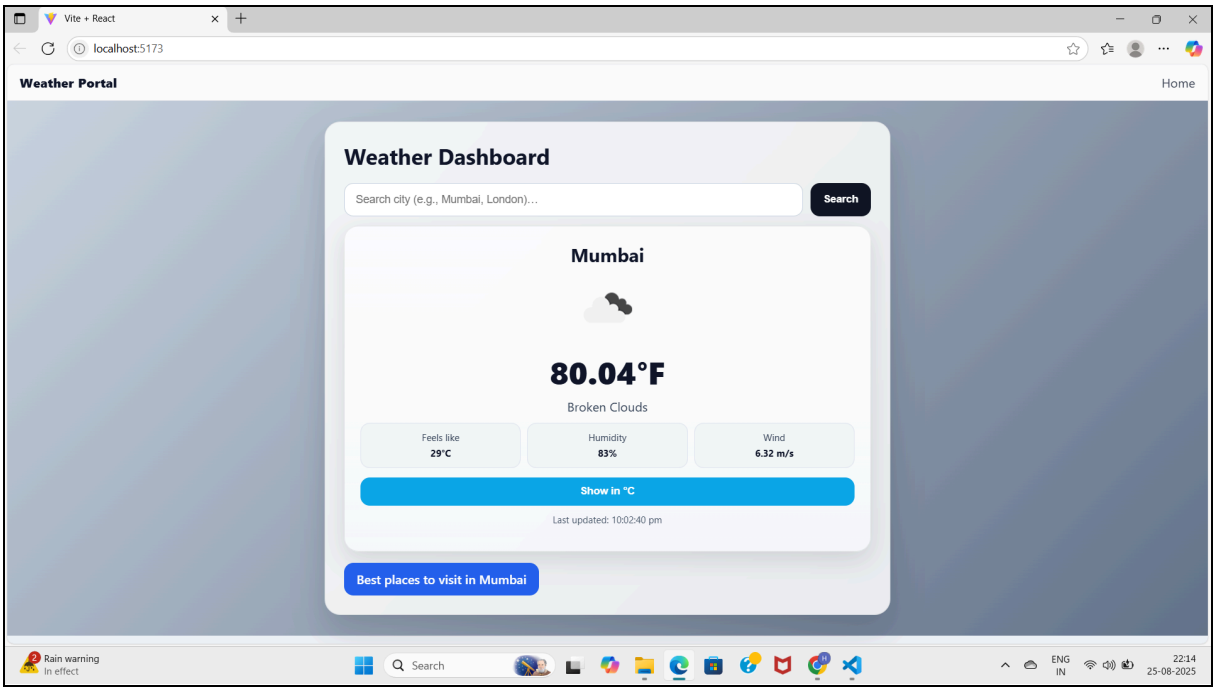
Search

ENG IN | 22:13 25-08-2025

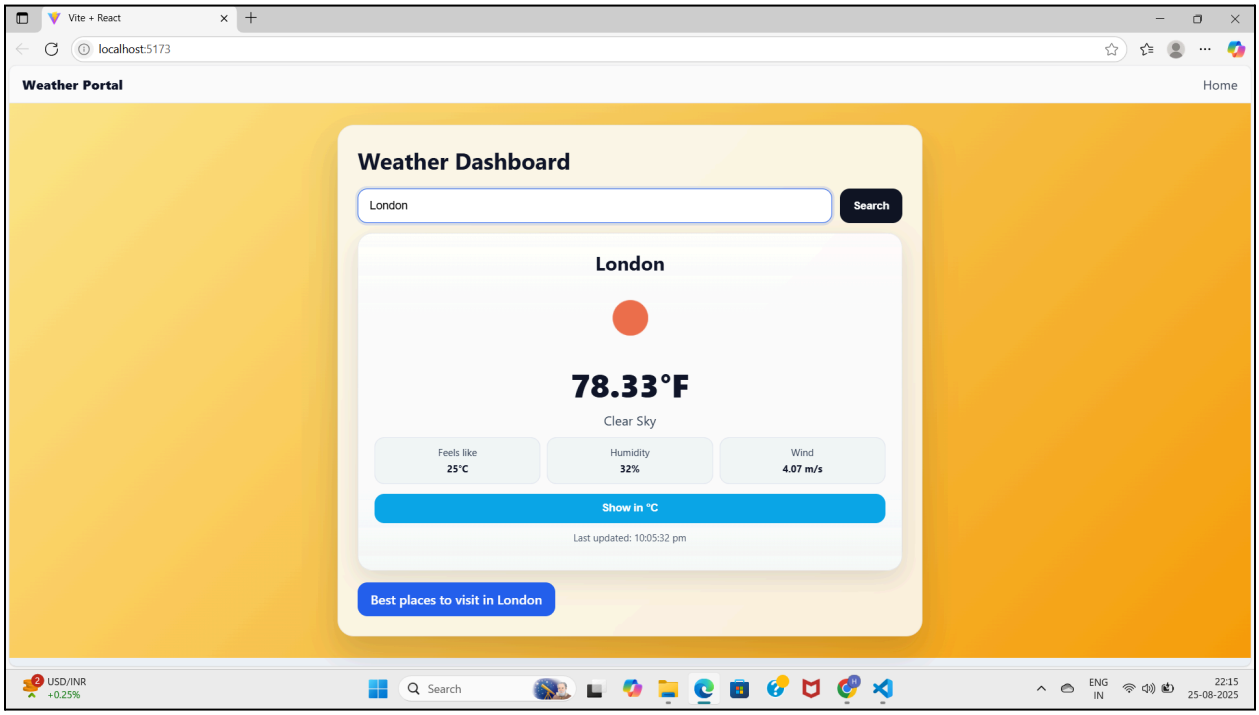
Output:



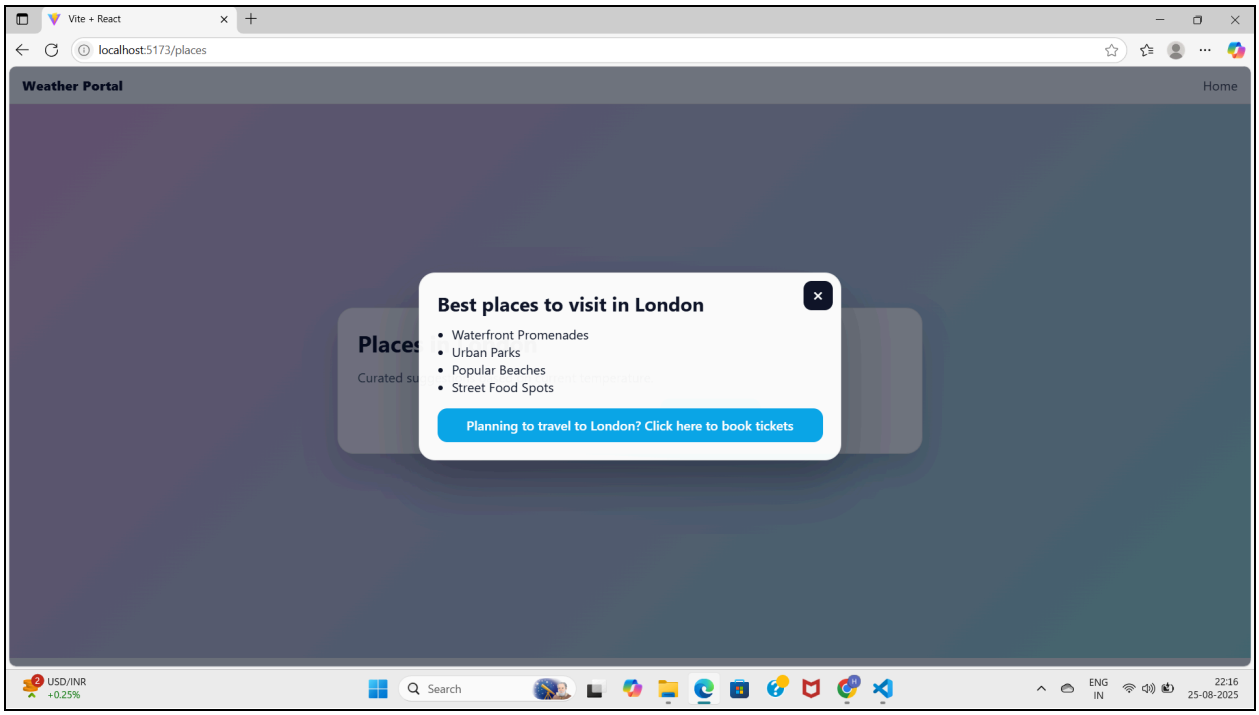
a. Landing page of portal showing temperature by fetching from API



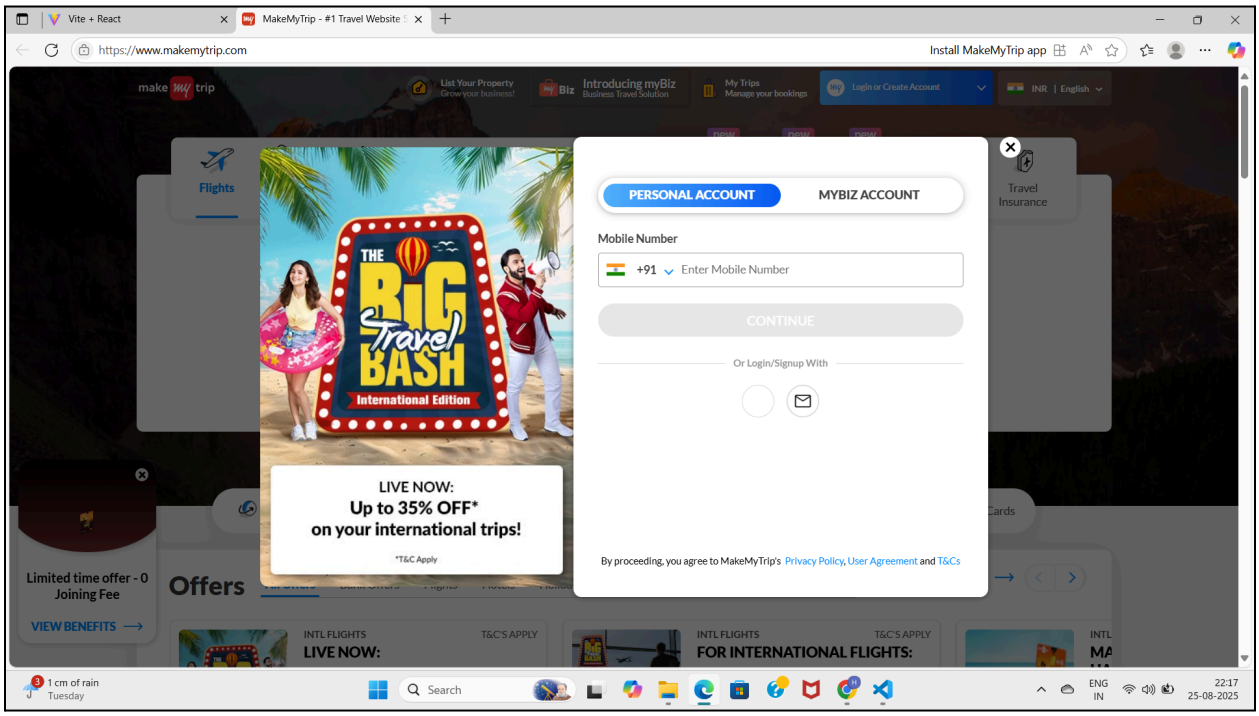
b. Temperature conversion



c. Search option



d. Routing showing best places to visit



e. Redirecting to the booking website

Conclusion:

This experiment successfully demonstrated the practical use of **React Hooks (useEffect, useContext, and Custom Hooks)** in building a dynamic weather application. The hooks allowed efficient state management, side effects handling, and code reusability. Additionally, the extended functionalities such as **routing, modal integration, travel booking links, and responsive design** made the project more practical, user-friendly, and engaging, achieving more than the initial objective.