Weather Info Card UI

UI/UX Design Fundamentals - Christ University

Submitted By:
Team Members:-

Full Name & Roll Number:
ALLAN PREM VARGHESE-2462030
ABEL FRANCIS-2462005
ARYA A.S. -2462189

College Email id:
allan.prem@btech.christuniversity.in
abel.francis@btech.christuniversity.in
arya.as@btech.christuniversity.in

Instructor Name:
Ms. Nagaveena

Index:

➔ Abstract

➔ Objectives

➔ Scope of the Project

➔ Tools & Technologies Used

➔ HTML Structure Overview

➔ CSS Styling Strategy

➔ Key Features

➔ Challenges Faced & Solutions

➔ Outcome

➔ Future Enhancements

➔ Sample Code

➔ Screenshots of Final Output

➔ Conclusion

➔ References

1. Abstract:

The Weather Info Card UI is a beautiful and responsive weather dashboard developed using HTML and CSS. It showcases weather data for several global cities using individual animated cards that depict varying weather conditions like sunny, cloudy, rainy, and stormy. The major objective of this project is to depict weather information in a beautiful and user-friendly interface while illustrating the implementation of innovative CSS animations for dynamic visual representations. Central technologies employed are HTML for structure and CSS for styling, layout, and animations. The end result is an interactive and visually appealing UI that can be used as a design concept stand-alone or combined with real-time weather APIs for dynamic updates. The project is beneficial for demonstrating front-end design capabilities, increasing user interaction, and presenting an easy way to show weather data.

2.Objectives:

- Create an intuitive interface based on contemporary UI/UX standards to display weather information in a well-readable manner.

- Create a totally responsive design with pure HTML and CSS to provide broad support for its usage across different devices and screen sizes.

- Use well-structured HTML5 semantic elements to improve readability, maintainability, and SEO-friendliness.

- Use CSS styling and animations to enrich branding, visual appearance, and interactivity.

- ➔ Provide accessibility and readability by ensuring correct contrast, font size, and content legibility on all devices.

- ➔ Design unique animated weather cards to denote various weather conditions, enhancing user experience.

3. Scope of the Project:

The project is exclusively front-end design, demonstrating the application of HTML and CSS to create an interactive user interface. There is no JavaScript functionality and no server-side integration, so all data shown remains static and for demonstration only. The design is responsive, and optimized for desktop, tablet, and mobile viewports to provide a consistent user experience across devices. Only pure code and open-source libraries have been utilized, without any dependency on external libraries or frameworks. The scope includes layout generation, styling, and weather card animation but not live weather data fetching or backend connectivity.

4. Tools & Technologies Used:

Tool/Technology	Purpose
HTML	Markup and content structure
CSS3	Styling and layout management
VS Code	Code editor

Chrome DevTools	Testing and debugging

5. HTML Structure Overview:

- Used semantic tags such as <header> and <main> to provide clear document structure and improve accessibility.

- Structured the layout into reusable weather card sections, each representing a specific city and weather condition.

- Each card contains a city name, weather icon, temperature, description, and additional info like wind speed and humidity.

- Weather condition effects are implemented using dedicated <div> elements (.sun-animation, .cloud-animation, .rain-animation, .lightning-animation) for styling and animation purposes.

- The design avoids unnecessary wrappers, keeping the HTML clean, readable, and maintainable while ensuring proper separation of content and presentation.

6. CSS Styling Strategy:

- Used an external CSS file (`style.css`) to separate styling from structure, ensuring cleaner and more maintainable code.
- Organized styles with comments and logical sections for easier navigation and updates.
- Applied CSS Grid for the main weather card layout and Flexbox for arranging elements inside the cards.
- Implemented media queries to make the design fully responsive across desktop, tablet, and mobile devices.
- Used hover effects and smooth transitions to enhance interactivity and visual appeal.
- Designed animated backgrounds (sun, clouds, rain, lightning) using keyframes for a dynamic user experience.
- Followed a mobile-first design approach, ensuring optimal performance and readability on smaller screens before scaling up to larger devices.

7. Key Features:

Feature	**Description**
Responsive Design	Adapts seamlessly to all screen sizes
Animated Weather Effects	Dynamic CSS animations for sunny, cloudy, rainy, and stormy conditions to enhance visual appeal.
Weather Info Cards	Individual cards displaying city name, temperature, weather icon, description, wind speed, and humidity.
Hover Interactions	Smooth transitions and scaling effects on card hover for better interactivity.
Clean & Accessible UI	High-contrast colors, legible fonts, and clear layouts for improved readability.
Pure Front-End Build	Created entirely with HTML and CSS, with no external libraries or JavaScript.

8. Challenges Faced & Solutions:

Challenge	**Solution**
Overlapping elements on small screens	Used media queries to stack elements and adjust spacing
Difficulty aligning items using `float`	Shifted to Flexbox and Grid for easier alignment and layout control
Typography scaling inconsistencies	Used relative units (em/rem) instead of fixed px values for better responsiveness

9. Outcome:

- Achieved a clean, consistent, and visually engaging front-end layout
- All key components function as intended using just HTML and CSS
- Learned about layout responsiveness and UI hierarchy in depth

10. Future Enhancements:

- Add JavaScript for interactivity (form validation, dynamic content)
- Integrate animations or transitions
- Backend integration for form submission
- Theme toggler (light/dark mode)

11.Sample Code:

This is a sample code not the actual code:-

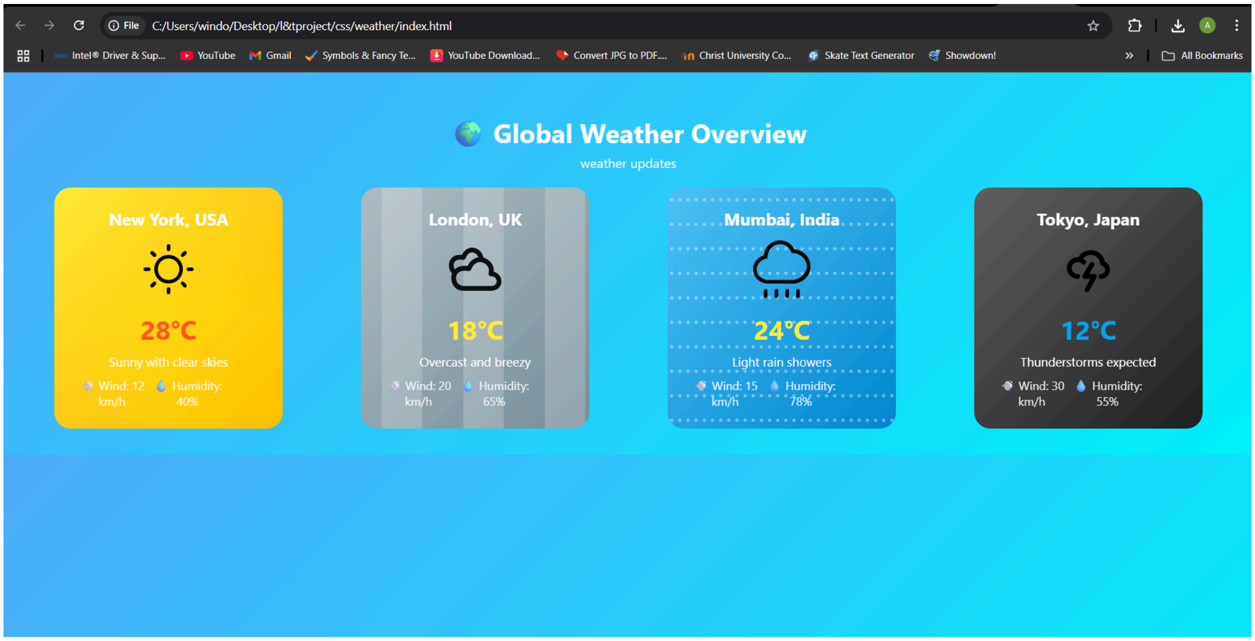
```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width,
initial-scale=1.0"/>
  <title>Weather Info Card UI</title>
  <style>
    * { margin: 0; padding: 0; box-sizing: border-box; }
    body {
      background: linear-gradient(135deg, #4facfe, #00f2fe);
      font-family: 'Segoe UI', sans-serif;
      padding: 2rem;
    }
    .dashboard-header {
      text-align: center;
      color: #fff;
      margin: 20px 0;
    }
    .dashboard-header h1 { font-size: 2rem; margin-bottom: 5px; }
    .dashboard-header p { font-size: 1rem; opacity: 0.85; }

    .weather-dashboard {
      display: grid;
      gap: 2rem;
      grid-template-columns: repeat(auto-fit, minmax(250px, 1fr));
      justify-items: center;
    }
    .weather-card {
      position: relative;
      padding: 1.5rem;
      border-radius: 20px;
      text-align: center;
      width: 100%;
      max-width: 280px;
      transition: transform 0.3s ease;
      color: white;
      overflow: hidden;
    }
    .weather-card:hover { transform: translateY(-5px); }
    .city { font-size: 1.3rem; margin-bottom: 0.8rem; }
    .weather-icon { width: 70px; height: 70px; margin-bottom: 0.8rem; }
    .temperature { font-size: 2rem; font-weight: bold; margin-bottom:
0.5rem; }
    .temperature.hot { color: #ff5722; }
    .temperature.mild { color: #ffeb3b; }
    .temperature.cold { color: #03a9f4; }
    .description { font-size: 0.95rem; }

    .sunny { background: linear-gradient(135deg, #ffeb3b, #ff9800); }
```

```
  .cloudy { background: linear-gradient(135deg, #b0bec5, #78909c); }
  .rain { background: linear-gradient(135deg, #4fc3f7, #0288d1); }
  .storm { background: linear-gradient(135deg, #616161, #212121); }

  /* Animations */
  .sun-animation {
    position: absolute; top: -20%; left: -20%;
    width: 200%; height: 200%;
    background: radial-gradient(circle, rgba(255,255,0,0.4) 20%, transparent 70%);
    animation: sunnyGlow 5s infinite alternate;
    z-index: 0;
  }
  @keyframes sunnyGlow {
    0% { transform: rotate(0deg) scale(1); opacity: 0.6; }
    100% { transform: rotate(360deg) scale(1.2); opacity: 1; }
  }

  .cloud-animation {
    position: absolute; top: 0; left: -50%;
    width: 200%; height: 100%;
    background: repeating-linear-gradient(90deg, rgba(255,255,255,0.2) 0 50px, transparent 50px 100px);
    animation: cloudDrift 15s linear infinite;
    z-index: 0;
  }
  @keyframes cloudDrift {
    0% { transform: translateX(0); }
    100% { transform: translateX(50%); }
  }

  .rain-animation {
    position: absolute; top: 0; left: 0;
    width: 100%; height: 100%;
    background-image: radial-gradient(circle, rgba(255,255,255,0.4) 2px, transparent 2px);
    background-size: 10px 30px;
    animation: rainFall 0.5s linear infinite;
    z-index: 0;
  }
  @keyframes rainFall {
    0% { background-position: 0 0; }
    100% { background-position: 0 30px; }
  }

  .lightning-animation {
    position: absolute; top: 0; left: 0;
    width: 100%; height: 100%;
    background: white; opacity: 0;
    animation: lightningFlash 4s infinite;
    z-index: 0;
  }
  @keyframes lightningFlash {
      0%, 95%, 100% { opacity: 0; }
      40% { opacity: 0.5; }
      42% { opacity: 0; }
      44% { opacity: 0.8; }
      46% { opacity: 0; }
    }

    .weather-card *:not(div:first-child) {
      position: relative;
      z-index: 1;
    }
    .extra-info {
      margin-top: 10px;
      font-size: 0.9rem;
      display: flex;
      justify-content: space-between;
      width: 80%;
      color: #fff;
      opacity: 0.9;
    }
    @media (max-width: 500px) {
      body { padding: 1rem; }
    }
  </style>
</head>
<body>
```

```
<header class="dashboard-header">
  <h1>Weather Info Card UI</h1>
  <p>Global weather updates with animated effects</p>
</header>

<main class="weather-dashboard">

  <div class="weather-card sunny">
    <div class="sun-animation"></div>
    <h2 class="city">New York, USA</h2>
    <img src="sunny.svg" alt="Sunny" class="weather-icon">
    <div class="temperature hot">28°C</div>
    <p class="description">Sunny with clear skies</p>
    <div class="extra-info">
      <span>Wind: 12 km/h</span>
      <span>Humidity: 40%</span>
    </div>
  </div>

  <div class="weather-card cloudy">
    <div class="cloud-animation"></div>
    <h2 class="city">London, UK</h2>
    <img src="cloudy.svg" alt="Cloudy" class="weather-icon">
    <div class="temperature mild">18°C</div>
    <p class="description">Overcast and breezy</p>
  <div class="extra-info">
    <span>Wind: 20 km/h</span>
    <span> Humidity: 65%</span>
  </div>
</div>

<div class="weather-card rain">
  <div class="rain-animation"></div>
  <h2 class="city">Mumbai, India</h2>
  <img src="rain.svg" alt="Rain" class="weather-icon">
  <div class="temperature mild">24°C</div>
  <p class="description">Light rain showers</p>
  <div class="extra-info">
    <span>Wind: 15 km/h</span>
    <span>Humidity: 78%</span>
  </div>
</div>

<div class="weather-card storm">
  <div class="lightning-animation"></div>
  <h2 class="city">Tokyo, Japan</h2>
  <img src="storm.svg" alt="Storm" class="weather-icon">
  <div class="temperature cold">12°C</div>
  <p class="description">Thunderstorms expected</p>
  <div class="extra-info">
    <span> Wind: 30 km/h</span>
```

```
        <span>Humidity: 55%</span>
      </div>
    </div>

  </main>

</body>
</html>
```

12.Screenshots of Final Output



Global Weather Overview

weather updates

New York, USA

28°C

Sunny with clear skies

Wind: 12 km/h Humidity: 40%

London, UK

18°C

Overcast and breezy

Wind: 20 km/h Humidity: 65%

Mumbai, India

24°C

Light rain showers

Wind: 15 km/h Humidity: 78%

Tokyo, Japan

12°C

Thunderstorms expected

Wind: 30 km/h Humidity: 55%

13. Conclusion:

This is a Weather Info Card UI that visually displays weather data for various cities via animated weather cards. It is coded using only HTML and CSS to emphasize creating an interactive, responsive, and accessible front-end design. The project helped solidify my front-end skills, especially with CSS Grid, Flexbox, and keyframe animations. I learned first-hand about responsive design, structuring of layouts, and UI look, as well as honed my skills in building user-friendly and visually appealing interfaces. The hands-on application of design principles gave me a clearer grasp of user-focused web design and showed how to use animations to enhance engagement without JavaScript.

14. References:

L&T LMS:

https://learn.lntedutech.com/Landing/MyCourse