

****Title: CodeTech IT Solutions Internship – Task Documentation WEATHER FORECAST APP****

****Introduction: ****

This documentation provides a detailed explanation of the task assigned during the CodeTech IT Solutions Internship program. The task involves writing a code using HTML, CSS and JavaScript to design an webpage for Weather Forecast APP . This documentation will cover the implementation details, rationale behind the code structure , and insights into the programming techniques utilised. Additionally it will include information about the intern, Chandolu Sai amareshwar, and his assigned ID, COD4451.

****Intern information: ****

Name: Chandolu Sai amareshwar

Intern ID: COD4451

****Tak Description: ****

The task assigned to Chandolu Sai amareshwar during the CodeTech IT S olutions Internship programis to write a code for weather forecast app using HTML,CSS and JavaScript.

****Implementation: ****

The implementation of the task involves utilising HTML,CSS and JavaScript to create a simple program that gives the output for weather forecast app webpage. Below is the code implementation.

...HTML

```
<!DOCTYPE html>

<head>
  <link rel="stylesheet" href="style2.css">
  <link rel="stylesheet" href=
"https://cdnjs.cloudflare.com/ajax/libs/animate.css/4.1.1/animate.min.css">
  <link rel="stylesheet" href=
"https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.1/css/all.min.css">
  <link rel="stylesheet" href=
"https://fonts.googleapis.com/css2?family=Montserrat:wght@400;700&display=swap">
  <title>Weather Forecast App</title>
</head>

<body>
  <div class="container">
    <div class="weather-card">

      <h3>
        Weather App
      </h3>
      <input type="text" id="city-input"
        placeholder="Enter city name">
      <button id="city-input-btn"
        onclick="weatherFn($('#city-input').val())">
```

```

        Get Weather
    </button>
    <div id="weather-info"
        class="animate__animated animate__fadeIn">
        <h3 id="city-name"></h3>
        <p id="date"></p>
        <!-- <img id="weather-icon" src="" alt="Weather Icon"> -->
        <p id="temperature"></p>
        <p id="description"></p>
        <p id="wind-speed"></p>
    </div>
</div>
</div>
<script src=
"https://code.jquery.com/jquery-3.6.0.min.js">
</script>
<script src=
"https://momentjs.com/downloads/moment.min.js">
</script>
<script src="script2.js"></script>
</body>

</html>

```

****Code Explanation: ****

<!DOCTYPE html>: This declaration specifies the document type and version of HTML being used, which is HTML5 in this case.

<head>: This section contains metadata and links to external resources used by the document.

External Stylesheets:

- **style2.css**: This stylesheet defines custom styles for elements in the document.
- **animate.min.css**: This stylesheet provides animations through the animate.css library.
- **all.min.css**: This stylesheet contains Font Awesome icons styles.

External Fonts:

- **Montserrat**: font from google fonts is used for the document.

<title>: Sets the title of the document to "Weather Forecast App".

<body>: This section contains the content of the document visible to the user.

.container: This is a container for the content of the application.

.weather-card: This is a card-like container for the weather application.

<h3>: Displays the title "Weather App". It is heading tag and a paired tag.

<input>: Allows users to input the name of a city for weather information.

<button>: When clicked, triggers a JavaScript function to fetch weather information for the entered city.

<div id="weather-info">: This div displays weather information fetched via JavaScript.

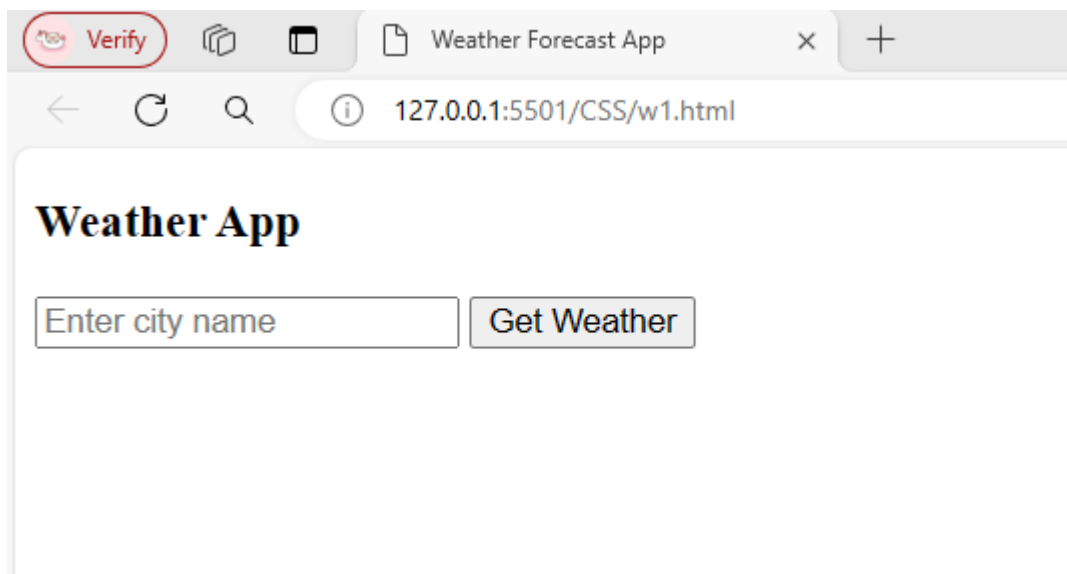
<h3 id="city-name">: Displays the name of the city.

<p id="date">: Displays the current date.

<p id="temperature">: Displays the temperature.

<p id="description">: Displays a brief description of the weather.

<p id="wind-speed">: Displays the wind speed.



The above shown is output after executing the HTML Code.

...CSS:

```
body {
  margin: 0;
  font-family: 'Montserrat', sans-serif;
  display: flex;
  justify-content: center;
  align-items: center;
  height: 100vh;
  background: linear-gradient(to right, #4CAF50, #2196F3);
}

.container {
  text-align: center;
}
```

```
.weather-card {  
  background-color: rgba(255, 255, 255, 0.95);  
  border-radius: 20px;  
  padding: 20px;  
  box-shadow: 0 0 30px rgba(0, 0, 0, 0.1);  
  transition: transform 0.3s ease-in-out;  
  width: 450px;  
}
```

```
.weather-card:hover {  
  transform: scale(1.05);  
}
```

```
#city-input {  
  padding: 15px;  
  margin: 10px 0;  
  width: 70%;  
  border: 1px solid #ccc;  
  border-radius: 5px;  
  font-size: 16px;  
}
```

```
#city-input:focus {  
  outline: none;  
  border-color: #2196F3;  
}
```

```
#city-input::placeholder {  
  color: #aaa;  
}
```

```
#city-input-btn {  
  padding: 10px;  
  background-color: #2196F3;  
  color: #fff;  
  border: none;  
  border-radius: 5px;  
  font-size: 16px;  
  cursor: pointer;  
}
```

```
#city-input-btn:hover {  
  background-color: #1565C0;  
}
```

```
#weather-info {  
  display: none;  
}
```

```
#weather-icon {  
  width: 100px;  
  height: 100px;  
}
```

```
#temperature {  
  font-size: 24px;  
  font-weight: bold;
```

```

margin: 8px 0;
}

#description {
  font-size: 18px;
  margin-bottom: 10px;
}

#wind-speed {
  font-size: 16px;
  color: rgb(255, 0, 0);
}

#date {
  font-size: 14px;
  color: rgb(255, 0, 0);
}

```

****Code Explanation: ****

body: Styles applied to the body element.

- **margin: 0;** Removes default margin.
- **font-family: 'Montserrat', sans-serif;** Sets the font family to Montserrat, a sans-serif font.
- **display: flex;** Uses Flexbox layout.
- **justify-content: center;** Centers content horizontally.
- **align-items: center;** Centers content vertically.
- **height: 100vh;** Sets the height to 100% of the viewport height.
- **background: linear-gradient(to right, #4CAF50, #2196F3);** Applies a linear gradient background from #4CAF50 to #2196F3, from left to right.

.container: Styles applied to elements with the class "container".

- **text-align: center;** Centers text horizontally.

.weather-card: Styles applied to elements with the class "weather-card".

- **background-color: rgba(255, 255, 255, 0.95);** Sets the background color to a semi-transparent white.
- **border-radius: 20px;** Rounds the corners with a radius of 20px.
- **padding: 20px;** Adds internal spacing of 20px.
- **box-shadow: 0 0 30px rgba(0, 0, 0, 0.1);** Adds a shadow effect.
- **transition: transform 0.3s ease-in-out;** Sets a transition effect for the transform property.
- **width: 450px;** Sets a fixed width of 450px.

.weather-card:hover: Styles applied when hovering over elements with the class "weather-card".

- **transform: scale(1.05);** Enlarges the element by 5% on hover.

#city-input: Styles applied to the element with the id "city-input".

- **padding: 15px;** Adds internal spacing of 15px.
- **margin: 10px 0;** Sets margin above and below the element to 10px.

- **width: 70%;**: Sets the width to 70% of its container.
- **border: 1px solid #ccc;**: Adds a 1px solid border with a light gray color.
- **border-radius: 5px;**: Rounds the corners with a radius of 5px.
- **font-size: 16px;**: Sets the font size to 16px.

#city-input:focus: Styles applied when the element with the id "city-input" is focused.

- **outline: none;**: Removes the default focus outline.
- **border-color: #2196F3;**: Changes the border color on focus to #2196F3.

#city-input::placeholder: Styles applied to the placeholder text of the element with the id "city-input".

- **color: #aaa;**: Sets the color of the placeholder text to light gray.

#city-input-btn: Styles applied to the element with the id "city-input-btn".

- **padding: 10px;**: Adds internal spacing of 10px.
- **background-color: #2196F3;**: Sets the background color to #2196F3.

color: #fff;: Sets the text color to white.

- **border: none;**: Removes the border.
- **border-radius: 5px;**: Rounds the corners with a radius of 5px.
- **font-size: 16px;**: Sets the font size to 16px.
- **cursor: pointer;**: Changes the cursor to a pointer on hover.

#city-input-btn:hover: Styles applied when hovering over the element with the id "city-input-btn".

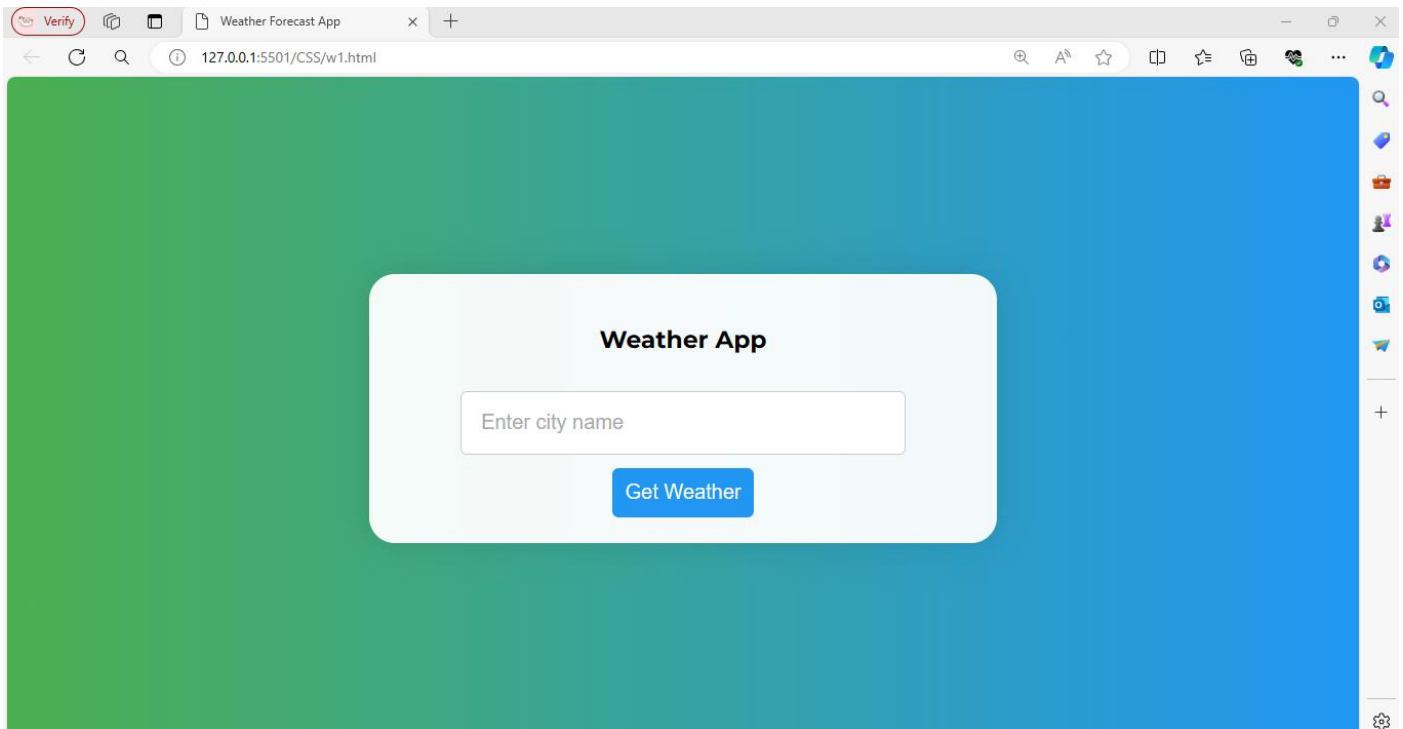
- **background-color: #1565C0;**: Changes the background color to #1565C0 on hover.

#weather-info: Styles applied to the element with the id "weather-info".

- **display: none;**: Hides the element by default.

#temperature, #description, #wind-speed, #date: Styles applied to elements with the corresponding ids.

- **font-size**: Sets the font size.
- **margin**: Sets the margin.
- **font-weight**: Sets the font weight for temperature.
- **color**: Sets the text color for wind-speed and date.



The above shown figure is the output after applying CSS Code.

...JavaScript

```
const url =  
  'https://api.openweathermap.org/data/2.5/weather';  
const apiKey =  
  'f00c38e0279b7bc85480c3fe775d518c';  
  
$(document).ready(function () {  
  weatherFn('Pune');  
});  
  
async function weatherFn(cName) {  
  const temp =  
    `${url}?q=${cName}&appid=${apiKey}&units=metric`;  
  try {  
    const res = await fetch(temp);  
    const data = await res.json();  
    if (res.ok) {  
      weatherShowFn(data);  
    } else {  
      alert('City not found. Please try again.');    }  
  } catch (error) {  
    console.error('Error fetching weather data:', error);  
  }  
}  
  
function weatherShowFn(data) {  
  $('#city-name').text(data.name);  
  $('#date').text(moment().  
    format('MMMM Do YYYY, h:mm:ss a'));  
  $('#temperature').  
    html(`${data.main.temp}°C`);
```

```

$('#description').
  text(data.weather[0].description);
$('#wind-speed').
  html(` Wind Speed: ${data.wind.speed} m/s `);
$('#weather-icon').
  attr('src',
    `...`);
$('#weather-info').fadeIn();
}

```

****Code Explanation: ****

Constants:

- `url`: Stores the base URL of the OpenWeatherMap API.
- `apiKey`: Stores the API key required for accessing the OpenWeatherMap API.

Document Ready Function:

- `$(document).ready(function () {...})`: This is a jQuery function that executes when the document (HTML) is fully loaded. It initiates the `weatherFn` function with the city name 'Pune'.

weatherFn Function:

- This function takes a city name (`cName`) as an argument.
- Constructs the API URL with the city name and API key.
- Attempts to fetch weather data from the OpenWeatherMap API using `fetch` and `await`.
- If the response is successful (`res.ok`), it calls the `weatherShowFn` function with the retrieved data. If not, it displays an alert indicating that the city was not found.
- Catches and handles any errors that may occur during the fetch process.

weatherShowFn Function:

- This function is responsible for displaying weather information on the webpage.
- It updates various HTML elements with the relevant weather data extracted from the API response.
- `$('#city-name').text(data.name)`: Updates the element with the id `city-name` with the name of the city.
- `$('#date').text(moment().format('MMMM Do YYYY, h:mm:ss a'))`: Updates the element with the id `date` with the current date and time using the Moment.js library.
- `$('#temperature').html(`${data.main.temp}°C`)`: Updates the element with the id `temperature` with the temperature in Celsius.
- `$('#description').text(data.weather[0].description)`: Updates the element with the id `description` with the weather description.
- `$('#wind-speed').html(Wind Speed: ${data.wind.speed} m/s)`: Updates the element with the id `wind-speed` with the wind speed.
- `$('#weather-icon').attr('src', '...')`: Updates the `src` attribute of the element with the id `weather-icon` with the appropriate weather icon URL (which is omitted in the provided code).
- `$('#weather-info').fadeIn()`: Fades in the element with the id `weather-info`, presumably displaying it on the webpage.



The final output of the code using HTML, CSS and JavaScript is shown above.

****Rationale: ****

The program utilizes the HTML code for providing the structure for the webpage, CSS Code provides the decoration for the web page for looking more attractive and JavaScript code provides the functionality for the web page so that it works effectively.

****Conclusion: ****

In conclusion, the task assigned to Chandolu Sai amreshwar during the CodeTech IT solutions internship program involved writing a code using Web technologies to provide a webpage for Weather Forecast APP. The implemented solution successfully accomplishes this task using HTML, CSS and JavaScript. This documentation provides insights into implementation details, code explanation, and rationale behind the chosen approach. Chandolu Sai amreshwar with Intern ID COD4451, has effectively completed this task as part of the internship program.

This concludes the documentation for the task “Weather Forecast APP” assigned during the CodeTech IT Solutions internship program.