

# Building a Weather Application Frontend with HTML, CSS & JavaScript

## Why Build a Weather App?

</>

#### **Hands-On Learning**

A practical, real-world project to solidify your understanding of core frontend technologies.

썀

#### **Integrated Skills**

Seamlessly combines UI design, external API integration, and dynamic user interactivity.

0

#### **High Utility**

Millions rely on daily weather information, making this a highly relevant and popular application.



#### **Fundamental Practice**

Perfect for honing your HTML structuring, CSS styling, and JavaScript logic skills.

## Step 1: Setting Up Your Project Environment

- **Project Folder:** Create a dedicated folder containing three essential files for your application:
  - index.html for defining the page structure.
  - styles.css for handling all visual design and layout.
  - script.js for implementing interactive functionality.
- Code Editor: Utilise a robust code editor, such as Visual Studio Code, for efficient development and debugging.
- API Key: Obtain an API key from OpenWeatherMap to fetch live, real-time weather data for global locations.



### Step 2: HTML Structure — The Skeleton

```
(find weather = fel legult, at)
```

The **HTML (index.html)** file forms the essential skeleton of your weather application.

- Define containers for search input, city name, temperature, and weather description.
- Use semantic tags like header, main, and footer for better organisation and accessibility.
- Include placeholders for dynamic content that JavaScript will populate, such as weather icons and daily forecasts.

## Step 3: CSS Styling — The Aesthetics

With the HTML skeleton in place, **CSS (styles.css)** brings your weather app to life visually.

- Apply styles for responsive layouts, ensuring functionality across devices.
- Design an intuitive user interface with clear typography and captivating background elements.
- Implement dynamic visual feedback for weather conditions-imagine a sunny day with bright colours versus a cloudy, muted palette.
- Ensure an engaging and visually consistent user experience.

```
elcemác lefül)>
         ites tripely Jeflt bolomel; On let to pain
    finagress haivds, contfculay>
Lingch fow = reatisfball; ster fellscation | dealers
est cogg: (1 inlscance = bestffowell ('caclssile'), reserved
ortar-lating intenttapsis/surless
   lest shill: steg/flexdayy retlerfel-fection (Clother) has seen below to
  tast-ApplstnatiVt erpraciest seller, enteri(Mest and
wility threlerish.resofiet. = bestbyrfiuml((lost)
       el: lest attite//gryfeactive_retteration(Keatle
        adite( (actune: tnngerful cclos))
             ablet weh: fligsly, = mlestery, color, of that i
          Clesst (lighrfaater-wated-"talll aliserater com
        gersrtatlig): fest hes/fāailer, spected comme
         congroling: adtibats, stactificile/(eller to be
```