<!doctype html>

<html lang=”en”>

<head>

<meta charset=”utf-8” />

<meta name=”viewport” content=”width=device-width, initial-scale=1” />

<title>Weather & News Hub • Public API Demo</title>

<!—Tailwind CSS via CDN for styling 🡪

<script src=<https://cdn.tailwindcss.com>></script>

<meta name=”color-scheme” content=”light dark” />

<style>

/\* small helper to clamp text \*/

.line-clamp-2 { display: -webkit-box; -webkit-line-clamp: 2; -webkit-box-orient: vertical; overflow: hidden; }

</style>

</head>

<body class=”min-h-screen bg-gradient-to-b from-slate-900 to-slate-800 text-slate-100”>

<div class=”max-w-4xl mx-auto p-5 space-y-8”>

<!—Weather Section 🡪

<section>

<header class=”flex items-center gap-3 mb-5”>

<svg xmlns=<http://www.w3.org/2000/svg> viewBox=”0 0 24 24” fill=”currentColor” class=”w-10 h-10 text-yellow-300”>

<path d=”M12 18a6 6 0 1 0 0-12 6 6 0 0 0 0 12Zm0 4a1 1 0 0 1-1-1v-1a1 1 0 1 1 2 0v1a1 1 0 0 1-1 1Zm0-18a1 1 0 0 1-1-1V2a1 1 0 1 1 2 0v1a1 1 0 0 1-1 1Zm10 8a1 1 0 0 1-1 1h-1a1 1 0 1 1 0-2h1a1 1 0 0 1 1 1ZM4 12a1 1 0 0 1-1 1H2a1 1 0 1 1 0-2h1a1 1 0 0 1 1 1Zm14.95 7.536a1 1 0 0 1-1.414 0l-.707-.707a1 1 0 0 1 1.414-1.415l.707.708a1 1 0 0 1 0 1.414ZM6.171 6.171a1 1 0 0 1-1.414 0l-.707-.707A1 1 0 0 1 5.464 4.05l.707.707a1 1 0 0 1 0 1.414Zm12.02-1.414a1 1 0 0 1 0 1.414l-.707.707A1 1 0 1 1 16.07 5.465l.707-.707a1 1 0 0 1 1.414 0ZM6.878 18.12a1 1 0 0 1 0-1.414l.707-.707a1 1 0 1 1 1.414 1.414l-.707.707a1 1 0 0 1-1.414 0Z”/>

</svg>

<div>

<h1 class=”text-2xl font-semibold”>Weather Now</h1>

<p class=”text-slate-300 text-sm”>Live weather from Open‑Meteo API</p>

</div>

</header>

<div class=”bg-white/5 backdrop-blur rounded-2xl p-4 shadow-xl space-y-3”>

<div class=”flex gap-2”>

<input id=”city” class=”flex-1 rounded-2xl px-4 py-2 bg-white/10 outline-none focus:ring-2 ring-yellow-300 placeholder:text-slate-400” placeholder=”Enter a city e.g., Hyderabad” />

<button id=”searchWeatherBtn” class=”rounded-2xl px-4 py-2 bg-yellow-300 text-slate-900 font-medium active:scale-[.98]”>Search</button>

<button id=”gpsBtn” class=”rounded-2xl px-3 py-2 bg-white/10 text-slate-100 font-medium active:scale-[.98]”>Use my location</button>

</div>

<p id=”statusWeather” class=”text-sm text-slate-300”></p>

<div id=”current” class=”grid grid-cols-2 gap-3”></div>

</div>

<h2 class=”text-lg font-semibold mt-6 mb-2”>Next 12 hours</h2>

<div id=”hours” class=”grid grid-cols-3 sm:grid-cols-4 gap-3”></div>

</section>

<!—News Section 🡪

<section>

<header class=”flex items-center gap-3 mb-5”>

<svg xmlns=<http://www.w3.org/2000/svg> fill=”none” viewBox=”0 0 24 24” stroke-width=”1.5” stroke=”currentColor” class=”w-10 h-10 text-blue-300”>

<path stroke-linecap=”round” stroke-linejoin=”round” d=”M3 5.25h18m-18 4.5h18m-18 4.5h18m-18 4.5h18” />

</svg>

<div>

<h1 class=”text-2xl font-semibold”>News Hub</h1>

<p class=”text-slate-300 text-sm”>Top stories from The Guardian Open Platform</p>

</div>

</header>

<div class=”bg-white/5 backdrop-blur rounded-2xl p-4 shadow-xl space-y-3”>

<div class=”flex gap-2”>

<input id=”query” class=”flex-1 rounded-2xl px-4 py-2 bg-white/10 outline-none focus:ring-2 ring-blue-300 placeholder:text-slate-400” placeholder=”Search news e.g., Technology” />

<button id=”searchNewsBtn” class=”rounded-2xl px-4 py-2 bg-blue-300 text-slate-900 font-medium active:scale-[.98]”>Search</button>

</div>

<p id=”statusNews” class=”text-sm text-slate-300”></p>

<div id=”news” class=”grid gap-4 mt-4”></div>

</div>

</section>

<footer class=”text-[11px] text-slate-400 mt-8 text-center”>Data: Open‑Meteo & The Guardian Open Platform • No personal key required (Guardian uses api‑key=test for demos)</footer>

</div>

<script>

// =============== WEATHER SECTION ===============

Const cityEl = document.getElementById(‘city’);

Const statusWeatherEl = document.getElementById(‘statusWeather’);

Const currentEl = document.getElementById(‘current’);

Const hoursEl = document.getElementById(‘hours’);

Const searchWeatherBtn = document.getElementById(‘searchWeatherBtn’);

Const gpsBtn = document.getElementById(‘gpsBtn’);

Function setStatusWeather(msg) { statusWeatherEl.textContent = msg || ‘’; }

Function weatherCard(html) { const div = document.createElement(‘div’); div.className=’bg-white/10 rounded-2xl p-4 shadow’; div.innerHTML=html; return div; }

Async function fetchJSON(url) {

Const res = await fetch(url);

If (!res.ok) throw new Error(‘Network error: ‘ + res.status);

Return await res.json();

}

Async function geocode(name) {

Const url = `https://geocoding-api.open-meteo.com/v1/search?name=${encodeURIComponent(name)}&count=1&language=en&format=json`;

Const data = await fetchJSON(url);

If (!data.results || data.results.length === 0) throw new Error(‘City not found’);

Return data.results[0];

}

Async function loadWeather({ latitude, longitude, name, country }) {

setStatusWeather(‘Loading weather…’);

const wurl = `https://api.open-meteo.com/v1/forecast?latitude=${latitude}&longitude=${longitude}&current=temperature\_2m,relative\_humidity\_2m,apparent\_temperature,precipitation,weather\_code,is\_day&hourly=temperature\_2m,precipitation\_probability,wind\_speed\_10m&timezone=auto`;

const w = await fetchJSON(wurl);

currentEl.innerHTML = ‘’;

const loc = `${name || ‘Your location’}${country ? ‘, ‘ + country : ‘’}`;

const c = w.current;

currentEl.append(

weatherCard(`<div class=”text-sm opacity-80”>${loc}</div>

<div class=”text-4xl font-bold”>${Math.round(c.temperature\_2m)}°C</div>

<div class=”text-sm”>Feels like ${Math.round(c.apparent\_temperature)}°C • Humidity ${c.relative\_humidity\_2m}%</div>`)

);

currentEl.append(

weatherCard(`<div class=”font-medium”>Precipitation</div>

<div class=”text-2xl”>${(c.precipitation ?? 0)} mm</div>`)

);

// Render next 12 hours

hoursEl.innerHTML = ‘’;

const times = w.hourly.time;

const temps = w.hourly.temperature\_2m;

const probs = w.hourly.precipitation\_probability;

const winds = w.hourly.wind\_speed\_10m;

const now = new Date();

for (let I = 0, shown = 0; I < times.length && shown < 12; i++) {

const t = new Date(times[i]);

if (t >= now) {

const hh = t.toLocaleTimeString([], { hour: ‘2-digit’, minute: ‘2-digit’ });

const item = document.createElement(‘div’);

item.className = ‘bg-white/10 rounded-2xl p-3 text-center’;

item.innerHTML = `<div class=”text-xs opacity-80”>${hh}</div>

<div class=”text-xl font-semibold”>${Math.round(temps[i])}°</div>

<div class=”text-xs”>Rain ${probs[i] ?? 0}%</div>

<div class=”text-[11px] opacity-80”>Wind ${Math.round(winds[i])} km/h</div>`;

hoursEl.append(item);

shown++;

}

}

setStatusWeather(‘’);

}

Async function handleWeatherSearch() {

Try {

currentEl.innerHTML = ‘’; hoursEl.innerHTML = ‘’;

const q = cityEl.value.trim() || ‘New Delhi’;

setStatusWeather(‘Searching…’);

const place = await geocode(q);

await loadWeather(place);

} catch (err) {

setStatusWeather(err.message);

}

}

Async function handleGPS() {

If (!navigator.geolocation) {

setStatusWeather(‘Geolocation not supported on this device.’);

return;

}

setStatusWeather(‘Getting your location…’);

navigator.geolocation.getCurrentPosition(async (pos) => {

try {

const { latitude, longitude } = pos.coords;

await loadWeather({ latitude, longitude, name: ‘Your location’ });

} catch (err) {

setStatusWeather(err.message);

}

}, (err) => {

setStatusWeather(err.message || ‘Could not get location.’);

}, { enableHighAccuracy: true, timeout: 10000, maximumAge: 300000 });

}

searchWeatherBtn.addEventListener(‘click’, handleWeatherSearch);

cityEl.addEventListener(‘keydown’, €=>{ if(e.key===’Enter’) handleWeatherSearch(); });

gpsBtn.addEventListener(‘click’, handleGPS);

handleWeatherSearch();

// =============== NEWS SECTION (The Guardian API) ===============

Const queryEl = document.getElementById(‘query’);

Const statusNewsEl = document.getElementById(‘statusNews’);

Const newsEl = document.getElementById(‘news’);

Const searchNewsBtn = document.getElementById(‘searchNewsBtn’);

Function setStatusNews(msg) { statusNewsEl.textContent = msg || ‘’; }

Const stripTags = (html) => (html || ‘’).replace(/<[^>]\*>/g, ‘’);

Function newsCard(article) {

Const thumb = article.fields && article.fields.thumbnail ? article.fields.thumbnail : ‘https://via.placeholder.com/300x200?text=No+Image’;

Const desc = article.fields && article.fields.trailText ? stripTags(article.fields.trailText) : ‘’;

Const div = document.createElement(‘div’);

div.className = ‘bg-white/10 rounded-2xl p-4 shadow flex flex-col sm:flex-row gap-4’;

div.innerHTML = `

<img src=”${thumb}” alt=”” class=”w-full sm:w-40 h-40 object-cover rounded-xl” />

<div class=”flex-1”>

<h3 class=”font-semibold text-lg mb-1 line-clamp-2”>${article.webTitle}</h3>

<p class=”text-sm opacity-80 mb-2 line-clamp-2”>${desc}</p>

<a href=”${article.webUrl}” target=”\_blank” class=”text-blue-300 text-sm”>Read more →</a>

</div>`;

Return div;

}

Async function fetchNews(topic = ‘technology’) {

Try {

setStatusNews(‘Loading news…’);

newsEl.innerHTML = ‘’;

const url = `https://content.guardianapis.com/search?q=${encodeURIComponent(topic)}&order-by=newest&page-size=10&show-fields=trailText,thumbnail&api-key=test`;

const res = await fetch(url);

if (!res.ok) throw new Error(‘Network error: ‘ + res.status);

const data = await res.json();

const results = data.response && data.response.results ? data.response.results : [];

if (results.length === 0) throw new Error(‘No news found’);

results.forEach(article => newsEl.append(newsCard(article)));

setStatusNews(‘’);

} catch (err) {

setStatusNews(err.message);

}

}

Function handleNewsSearch() {

Const q = queryEl.value.trim() || ‘technology’;

fetchNews(q);

}

searchNewsBtn.addEventListener(‘click’, handleNewsSearch);

queryEl.addEventListener(‘keydown’, €=>{ if(e.key===’Enter’) handleNewsSearch(); });

fetchNews();

</script>

</body>

</html>