**CODE**

**HTML**

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="utf-8" />

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

  <title>WeatherSim — Weather Dashboard</title>

  <!-- Bootstrap & Google Font -->

  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet">

  <link href="https://fonts.googleapis.com/css2?family=Poppins:wght@400;600;700&display=swap" rel="stylesheet">

  <!-- Leaflet Map -->

  <link rel="stylesheet" href="https://unpkg.com/leaflet@1.9.4/dist/leaflet.css" />

  <script src="https://unpkg.com/leaflet@1.9.4/dist/leaflet.js"></script>

  <!-- jsPDF for report download -->

  <script src="https://cdn.jsdelivr.net/npm/jspdf@2.5.1/dist/jspdf.umd.min.js"></script>

  <!-- Chart.js -->

  <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>

  <!-- Local CSS -->

  <link rel="stylesheet" href="style.css">

</head>

<body>

  <!-- animation overlay -->

  <div id="weatherAnimation" class="pointer-events-none"></div>

  <!-- AUTH OVERLAY -->

  <div id="authOverlay" class="auth-overlay d-none">

    <div class="auth-graphics">

      <span class="orb orb1"></span>

      <span class="orb orb2"></span>

      <span class="orb orb3"></span>

      <span class="orb orb4"></span>

    </div>

    <div class="auth-center">

      <div class="text-center mb-3">

        <span class="logo-glass me-2"><img src="images/logo.jpg" alt="logo" width="44"></span>

      </div>

      <h2 class="auth-title mb-2">Welcome to WeatherSim</h2>

      <p class="auth-subtitle mb-4">Sign up or sign in to continue</p>

      <div class="auth-card-dark p-4">

        <ul class="nav nav-pills mb-3" id="authTabs">

          <li class="nav-item"><button class="nav-link active" id="tabSignup">Sign Up</button></li>

          <li class="nav-item"><button class="nav-link" id="tabSignin">Sign In</button></li>

        </ul>

        <div id="signupForm">

          <div class="mb-2"><input id="suName" type="text" class="form-control form-control-dark" placeholder="Full name"></div>

          <div class="mb-2"><input id="suEmail" type="email" class="form-control form-control-dark" placeholder="Email"></div>

          <div class="mb-3"><input id="suPass" type="password" class="form-control form-control-dark" placeholder="Password"></div>

          <button id="signupBtn" class="btn btn-primary w-100">Create Account</button>

        </div>

        <div id="signinForm" class="d-none">

          <div class="mb-2"><input id="siEmail" type="email" class="form-control form-control-dark" placeholder="Email"></div>

          <div class="mb-3"><input id="siPass" type="password" class="form-control form-control-dark" placeholder="Password"></div>

          <button id="signinBtn" class="btn btn-primary w-100">Sign In</button>

        </div>

      </div>

      <p class="text-secondary small mt-3 mb-0">By continuing you agree to the demo terms.</p>

    </div>

  </div>

  <!-- NAVBAR -->

  <nav class="navbar navbar-expand-lg navbar-dark bg-dark shadow-sm glass-nav">

    <div class="container">

      <a class="navbar-brand d-flex align-items-center" href="#">

        <span class="logo-glass me-2"><img src="images/logo.jpg" alt="logo" width="40"></span>

        WeatherSim

      </a>

      <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navMenu">

        <span class="navbar-toggler-icon"></span>

      </button>

      <div id="navMenu" class="collapse navbar-collapse">

        <ul class="navbar-nav ms-auto align-items-lg-center">

          <li class="nav-item"><a class="nav-link active" href="#">Home</a></li>

          <li class="nav-item"><a class="nav-link" href="#forecastSection">Forecast</a></li>

          <li class="nav-item"><a class="nav-link" href="#comparisonSection">Compare</a></li>

          <li class="nav-item"><a class="nav-link" href="#aboutSection">About</a></li>

          <li class="nav-item ms-lg-3">

            <button id="themeToggle" class="btn btn-outline-light btn-sm">Light Mode</button>

          </li>

          <li class="nav-item ms-lg-2">

            <button id="signOutBtn" class="btn btn-outline-light btn-sm">Sign Out</button>

          </li>

        </ul>

      </div>

    </div>

  </nav>

  <!-- WELCOME HERO -->

  <section id="welcome" class="vh-75 d-flex flex-column justify-content-center align-items-center text-center px-3">

    <span class="logo-glass mb-3"><img src="images/logo.jpg" alt="App Logo" width="120"></span>

    <h1 class="display-5 fw-bold">Welcome to WeatherSim</h1>

    <p class="lead text-white mb-3">Simulate weather, view hourly & 5-day forecasts, compare cities, and explore UI features — powered by live API.</p>

    <div class="input-group w-50 mx-auto" style="max-width:640px;">

      <input id="cityInputHome" type="text" class="form-control" placeholder="Type a city (e.g., London, New York, Tokyo)">

      <button id="startSearchBtn" class="btn btn-primary">Start</button>

    </div>

  </section>

  <!-- APP CONTENT (hidden until first search) -->

  <main class="container mb-5 d-none" id="appContent">

    <!-- SEARCH + HISTORY + CONTROLS -->

    <div class="row mt-4 align-items-center g-2">

      <div class="col-md-6">

        <div class="mb-2 d-flex align-items-center gap-2">

          <button id="historyToggle" class="btn btn-outline-light btn-sm">See History</button>

          <div id="searchHistory" class="history-dropdown d-none"></div>

        </div>

        <div class="input-group">

          <input id="cityInput" type="text" class="form-control" placeholder="Enter city">

          <button id="searchBtn" class="btn btn-primary">Search</button>

        </div>

      </div>

      <div class="col-md-6 d-flex justify-content-end gap-2">

        <button id="unitToggle" class="btn btn-primary">Show °F</button>

        <button id="geoBtn" class="btn btn-primary">Use My Location</button>

        <button id="downloadBtn" class="btn btn-primary">Download Report</button>

      </div>

    </div>

    <!-- HERO WEATHER -->

    <div id="heroContainer" class="mt-4 transition-fade">

      <div id="hero" class="p-4 rounded-3 glass text-center">

        <h2 id="cityName" class="fw-bold">--</h2>

        <img id="heroIcon" alt="icon" width="100" src="images/sunny\_icon.jpg">

        <p id="heroTemp" class="fs-3 fw-semibold mt-2">-- °C</p>

        <p id="heroExtra" class="mb-0 text-white">Wind: -- | Humidity: -- | UV: --</p>

      </div>

      <div id="heroSkeleton" class="p-4 rounded-3 glass skeleton d-none">

        <div class="skeleton-line skeleton-title w-50 mx-auto mb-3"></div>

        <div class="skeleton-circle mx-auto mb-3"></div>

        <div class="skeleton-line w-25 mx-auto mb-2"></div>

        <div class="skeleton-line w-75 mx-auto"></div>

      </div>

    </div>

    <!-- DRESSING ADVICE -->

    <div id="advice" class="mt-3 transition-fade"></div>

    <!-- HOURLY -->

    <section class="mt-4 transition-fade">

      <h5 class="mb-2">Next 24 Hours</h5>

      <div id="hourlyForecast" class="d-flex overflow-auto py-2"></div>

      <div id="hourlySkeleton" class="d-flex overflow-auto py-2 d-none">

        <div class="skeleton-card me-2"></div>

        <div class="skeleton-card me-2"></div>

        <div class="skeleton-card me-2"></div>

        <div class="skeleton-card me-2"></div>

        <div class="skeleton-card me-2"></div>

        <div class="skeleton-card me-2"></div>

      </div>

    </section>

    <!-- TEMP CHART -->

    <section class="mt-4 transition-fade">

      <h5 class="mb-2">Temperature Trend</h5>

      <div class="card p-3 glass">

        <canvas id="tempChart"></canvas>

      </div>

    </section>

    <!-- MAP & PLACES -->

    <section class="mt-4 transition-fade">

      <h5 class="mb-2">Map & Places</h5>

      <div class="card p-3 map-card">

        <div class="row g-2 align-items-center mb-3">

          <div class="col-md-5"><input id="placeSearch" class="form-control" type="text" placeholder="Search a place (city, landmark)"></div>

          <div class="col-md-3"><input id="latInput" class="form-control" type="text" placeholder="Latitude" readonly></div>

          <div class="col-md-3"><input id="lonInput" class="form-control" type="text" placeholder="Longitude" readonly></div>

          <div class="col-md-1 text-end"><button id="useCoordsBtn" class="btn btn-primary w-100">Use</button></div>

        </div>

        <div id="map" class="map-white"></div>

      </div>

    </section>

    <!-- 5-DAY FORECAST -->

    <section id="forecastSection" class="mt-4 transition-fade">

      <h5 class="mb-2">5-Day Forecast</h5>

      <div id="forecast" class="row gy-3"></div>

      <div id="forecastSkeleton" class="row gy-3 d-none">

        <div class="col-6 col-md-2"><div class="skeleton-tile"></div></div>

        <div class="col-6 col-md-2"><div class="skeleton-tile"></div></div>

        <div class="col-6 col-md-2"><div class="skeleton-tile"></div></div>

        <div class="col-6 col-md-2"><div class="skeleton-tile"></div></div>

        <div class="col-6 col-md-2"><div class="skeleton-tile"></div></div>

        <div class="col-6 col-md-2"><div class="skeleton-tile"></div></div>

      </div>

    </section>

    <!-- COMPARISON -->

    <section id="comparisonSection" class="mt-4 transition-fade">

      <h5 class="mb-2">Compare Cities</h5>

      <div class="card p-3 glass mb-3">

        <div class="row g-2 align-items-center">

          <div class="col-md-5">

            <input id="compareCityA" class="form-control" type="text" placeholder="Enter first city">

          </div>

          <div class="col-md-5">

            <input id="compareCityB" class="form-control" type="text" placeholder="Enter second city">

          </div>

          <div class="col-md-2 text-end">

            <button id="compareBtn" class="btn btn-success">Compare</button>

          </div>

        </div>

        <canvas id="comparisonChart" class="mt-3"></canvas>

      </div>

    </section>

    <!-- ABOUT -->

    <section id="aboutSection" class="mt-4">

      <h6>About</h6>

      <p class="text-light-strong">This simulator uses live API data to provide accurate weather forecasts. All UI features are fully interactive and professional. Enjoy hourly and 5-day outlooks, compare cities side-by-side, switch between °C/°F, and experience subtle UI animations and a theme toggle. Your recent searches are saved for quick access.</p>

      <ul class="text-light-strong mb-0">

        <li>Live weather powered by OpenWeather.</li>

        <li>Interactive charts for trends and comparisons.</li>

        <li>Mobile-friendly, accessible design with keyboard shortcuts.</li>

      </ul>

    </section>

  </main>

  <!-- SITE FOOTER -->

  <footer class="site-footer mt-5">

    <div class="container py-4">

      <div class="row gy-4">

        <div class="col-12 col-md-4">

          <div class="d-flex align-items-center mb-1 small-footer-logo">

            <span class="logo-glass me-2"><img src="images/logo.jpg" alt="logo" width="28"></span>

            <strong class="fs-6">WeatherSim</strong>

          </div>

          <p class="mb-0 text-light-strong small">A modern, educational weather dashboard showcasing clean UI and live data.</p>

        </div>

        <div class="col-6 col-md-4">

          <h6 class="text-light mb-2 small">Explore</h6>

          <ul class="list-unstyled small footer-links small">

            <li><a href="#welcome">Home</a></li>

            <li><a href="#forecastSection">Forecast</a></li>

            <li><a href="#comparisonSection">Compare</a></li>

            <li><a href="#aboutSection">About</a></li>

          </ul>

        </div>

        <div class="col-6 col-md-4">

          <h6 class="text-light mb-2 small">Contact</h6>

          <ul class="list-unstyled small text-light-strong mb-0">

            <li>Email: contact@weathersim.in</li>

            <li>API: OpenWeather</li>

            <li>Theme: Light/Dark with glass UI</li>

          </ul>

        </div>

      </div>

      <hr class="footer-sep my-4">

      <div class="d-flex flex-column flex-md-row justify-content-between small text-secondary">

        <span>© <span id="footerYear">2025</span> WeatherSim. All rights reserved.</span>

        <span>Made for UI/UX practice.</span>

      </div>

    </div>

  </footer>

  <!-- Local JS -->

  <script src="script.js"></script>

</body>

</html>

**CSS**

:root{

  --glass-bg: rgba(255,255,255,0.06);

  --accent: #cc1ea0;

  --muted: #b0b0b0;

  --text-color: #eaeaea;

}

/\* Global dark theme \*/

body{

  margin:0;

  background: url('images/default.jpg') center/cover no-repeat fixed;

  color: var(--text-color);

  font-family: 'Poppins', sans-serif;

  transition: background 0.9s ease-in-out;

}

/\* Light theme \*/

.light body, body.light{

  color:#1e1e1e;

}

.light .glass, body.light .glass{ color:#1e1e1e; }

.light .navbar, body.light .navbar{ background-color: rgba(255,255,255,0.75) !important; }

.light .nav-link, body.light .nav-link{ color:#222 !important; }

.light .btn-outline-light, body.light .btn-outline-light{ color:#222; border-color:#bbb; }

.light footer, body.light footer{ color:#444; }

/\* Glassy navbar \*/

.glass-nav{

  background: linear-gradient(180deg, rgba(0,0,0,0.5), rgba(0,0,0,0.2)) !important;

  backdrop-filter: blur(8px);

  border-bottom: 1px solid rgba(255,255,255,0.08);

}

/\* glass cards \*/

.glass{

  background: var(--glass-bg);

  border-radius: 12px;

  box-shadow: 0 8px 30px rgba(2,6,23,0.6);

  backdrop-filter: blur(6px);

  border: 1px solid rgba(255,255,255,0.05);

  color: #eaeaea;

}

/\* buttons hover \*/

.btn-primary{ background-image: linear-gradient(45deg, #7b5cff, #22d3ee); border: none; }

.btn-primary:hover{ filter: brightness(1.05); transform: translateY(-1px); }

.btn-info:hover, .btn-outline-light:hover{ transform: translateY(-1px); }

.btn-info, .btn-success, .btn-outline-light{ background-image: none; background: #7b5cff; border: none; color: #fff; }

.btn-success:hover{ filter: brightness(1.05); }

.btn{ transition: transform .15s ease, filter .2s ease; }

/\* navbar brand \*/

.navbar-brand img{ border-radius:8px; }

/\* glass frame for logos \*/

.logo-glass{ display:inline-flex; align-items:center; justify-content:center; padding:6px; border-radius:10px; background: var(--glass-bg); backdrop-filter: blur(6px); border:1px solid rgba(255,255,255,0.08); box-shadow: 0 4px 20px rgba(2,6,23,0.4); }

.logo-glass img{ display:block; border-radius:8px; }

.light .logo-glass{ background: rgba(255,255,255,0.6); border-color: rgba(0,0,0,0.06); box-shadow: 0 4px 16px rgba(0,0,0,0.08); }

/\* welcome section \*/

#welcome{

  background: linear-gradient(180deg, rgba(255,255,255,0.02), transparent);

}

/\* skeletons \*/

.skeleton{ position: relative; overflow: hidden; }

.skeleton::after{ content:""; position:absolute; inset:0; background: linear-gradient(90deg, transparent, rgba(255,255,255,0.08), transparent); transform: translateX(-100%); animation: shimmer 1.2s infinite; }

@keyframes shimmer{ to{ transform: translateX(100%);} }

.skeleton-line{ height:14px; background: rgba(255,255,255,0.08); border-radius:6px; }

.skeleton-title{ height:24px; }

.skeleton-circle{ width:80px; height:80px; border-radius:50%; background: rgba(255,255,255,0.08); }

.skeleton-card{ min-width:120px; height:100px; border-radius:10px; background: rgba(255,255,255,0.06); position:relative; overflow:hidden; }

.skeleton-tile{ height:140px; border-radius:10px; background: rgba(255,255,255,0.06); }

/\* small transition for updates \*/

.transition-fade{ opacity:1; transform:none; transition: opacity 0.45s, transform 0.45s; }

/\* search history \*/

#searchHistory .history-btn{

  margin-right:8px;

  margin-bottom:6px;

  border-radius:20px;

  padding:6px 12px;

  background: rgba(255,255,255,0.05);

  border:1px solid rgba(255,255,255,0.05);

  cursor:pointer;

  color:#eaeaea;

}

/\* hourly scroll \*/

#hourlyForecast{ padding-bottom:8px; }

#hourlyForecast .card{ min-width:120px; margin-right:10px; text-align:center; background: rgba(255,255,255,0.05); border:none; color:#eaeaea; }

.light #hourlyForecast .card{ color:#222; background: rgba(0,0,0,0.04); }

/\* forecast cards \*/

#forecast .card{ background: linear-gradient(180deg, rgba(255,255,255,0.02), rgba(255,255,255,0.01)); border-radius:10px; padding:14px; text-align:center; color:#eaeaea; border:none; }

.light #forecast .card{ color:#222; background: rgba(0,0,0,0.04); }

/\* animation overlay \*/

#weatherAnimation{ position:fixed; top:0; left:0; width:100%; height:100%; pointer-events:none; z-index:2; }

/\* raindrop \*/

.raindrop{ position:absolute; top:-10px; width:2px; height:14px; background:rgba(255,255,255,0.6); animation:fall linear infinite; opacity:0.8; }

@keyframes fall{ to{ transform: translateY(110vh); opacity:0.2; } }

/\* cloud \*/

.cloud{ position:absolute; top:10%; width:120px; height:60px; background:rgba(255,255,255,0.06); border-radius:60px; animation: floatCloud linear infinite; }

@keyframes floatCloud{ from{ transform: translateX(-30vw); } to{ transform: translateX(130vw); } }

/\* chart canvas \*/

canvas{ background: transparent !important; }

/\* Make hero extras and advice more visible \*/

#heroExtra{ color:#ffffff !important; text-shadow: 0 1px 2px rgba(0,0,0,0.5); }

#advice .glass{ color:#ffffff; border-color: rgba(255,255,255,0.2); }

/\* History dropdown \*/

.history-dropdown{ display:flex; flex-wrap:wrap; gap:6px; }

/\* About brighter text \*/

.text-light-strong{ color:#e8edf7 !important; }

/\* Map full-bleed white \*/

.map-card{ background:#fff; border-radius:12px; }

.map-white{ width:100%; height: 420px; border-radius: 10px; }

/\* footer tweaks \*/

footer{ margin-top: 30px; color:#b0b0b0; }

/\* site footer \*/

.site-footer{ background: #0e0f12; color:#b9c0cf; border-top:1px solid rgba(255,255,255,0.08); }

.site-footer a{ color:#d1d5db; text-decoration:none; }

.site-footer a:hover{ color:#ffffff; text-decoration:underline; }

.site-footer .footer-sep{ border-color: rgba(255,255,255,0.08); }

.site-footer .footer-links li{ margin-bottom:6px; }

.light .site-footer{ background:#0e0f12; }

/\* fade-in helper \*/

.fade-in{ animation: fadeIn 0.45s ease both; }

@keyframes fadeIn{ from{ opacity:0; transform:translateY(6px);} to{ opacity:1; transform:translateY(0);} }

/\* responsive \*/

@media (max-width:768px){

  #welcome h1{ font-size:1.8rem; }

  #welcome p{ font-size:0.95rem; }

}

/\* auth overlay \*/

.auth-overlay{ position:fixed; inset:0; background: #000; z-index:1050; overflow:hidden; }

.auth-graphics{ position:absolute; inset:0; pointer-events:none; }

.orb{ position:absolute; width:360px; height:360px; border-radius:50%; filter: blur(80px); opacity:0.35; }

.orb1{ background:#22d3ee; top:-80px; left:-60px; animation: drift1 16s infinite alternate ease-in-out; }

.orb2{ background:#7b5cff; bottom:-120px; right:-100px; animation: drift2 18s infinite alternate ease-in-out; }

.orb3{ background:#f472b6; top:30%; right:20%; animation: drift3 22s infinite alternate ease-in-out; }

.orb4{ background:#34d399; bottom:20%; left:25%; animation: drift4 20s infinite alternate ease-in-out; }

@keyframes drift1{ to{ transform: translate(60px, 40px) scale(1.1);} }

@keyframes drift2{ to{ transform: translate(-50px, -20px) scale(1.05);} }

@keyframes drift3{ to{ transform: translate(30px, -30px) scale(1.08);} }

@keyframes drift4{ to{ transform: translate(-40px, 30px) scale(1.06);} }

.auth-center{ position:relative; z-index:2; width: min(560px, 94vw); margin: 8vh auto; color:#eaeaea; text-align:center; }

.auth-title{ font-weight:700; }

.auth-subtitle{ color:#b0b0b0; }

.auth-card-dark{ background: rgba(255,255,255,0.04); border:1px solid rgba(255,255,255,0.08); border-radius:12px; box-shadow: 0 10px 40px rgba(0,0,0,0.6); text-align:left; }

.auth-card-dark .nav-link{ color:#d1d5db; }

.auth-card-dark .nav-link.active{ background: rgba(255,255,255,0.14); }

.form-control-dark{ background: rgba(255,255,255,0.06); border:1px solid rgba(255,255,255,0.1); color:#fff; }

.form-control-dark::placeholder{ color:#cbd5e1; }

/\* map \*/

#map{ width: 100%; height: 360px; border-radius: 10px; overflow: hidden; }

**JS**

const WEATHER\_API\_KEY = "6b269ed9c2c30ffa5f2d349325ff1b26";

/\* =======================

   Utilities & helpers

======================= \*/

function getRandomUV(){ return Math.floor(Math.random()\*11); }

function getIconFile(name){

  const condition = name.toLowerCase();

  // Check snow first before clouds (since "snow" might contain "cloud")

  if(condition.includes('snow')) return 'snow\_icon.jpg';

  if(condition.includes('rain') || condition.includes('drizzle')) return 'rain\_icon.jpg';

  if(condition.includes('thunderstorm') || condition.includes('storm')) return 'storm\_icon.jpg';

  if(condition.includes('clear') || condition.includes('sunny')) return 'sunny\_icon.jpg';

  if(condition.includes('cloud') || condition.includes('mist') || condition.includes('fog')) return 'cloudy\_icon.jpg';

  return 'storm\_icon.jpg'; // default fallback

}

function getBackgroundFile(name){

  const condition = name.toLowerCase();

  // Check snow first before clouds (since "snow" might contain "cloud")

  if(condition.includes('snow')) return 'snow.jpg';

  if(condition.includes('rain') || condition.includes('drizzle')) return 'rain.jpg';

  if(condition.includes('thunderstorm') || condition.includes('storm')) return 'storm.jpg';

  if(condition.includes('clear') || condition.includes('sunny')) return 'sunny.jpg';

  if(condition.includes('cloud') || condition.includes('mist') || condition.includes('fog')) return 'cloudy.jpg';

  return 'storm.jpg'; // default fallback

}

function formatTemp(t){

  if(unit === 'metric') return `${t} °C`;

  return `${Math.round(t \* 9/5 + 32)} °F`;

}

/\* =======================

   DOM refs & state

======================= \*/

const startSearchBtn = document.getElementById("startSearchBtn");

const cityInputHome = document.getElementById("cityInputHome");

const appContent = document.getElementById("appContent");

const welcome = document.getElementById("welcome");

const themeToggle = document.getElementById('themeToggle');

const signOutBtn = document.getElementById('signOutBtn');

const authOverlay = document.getElementById('authOverlay');

const tabSignup = document.getElementById('tabSignup');

const tabSignin = document.getElementById('tabSignin');

const signupBtn = document.getElementById('signupBtn');

const signinBtn = document.getElementById('signinBtn');

const suName = document.getElementById('suName');

const suEmail = document.getElementById('suEmail');

const suPass = document.getElementById('suPass');

const siEmail = document.getElementById('siEmail');

const siPass = document.getElementById('siPass');

const searchBtn = document.getElementById("searchBtn");

const cityInput = document.getElementById("cityInput");

const unitToggle = document.getElementById("unitToggle");

const geoBtn = document.getElementById("geoBtn");

const searchHistoryDiv = document.getElementById("searchHistory");

const historyToggle = document.getElementById('historyToggle');

let unit = 'metric';

let unitSymbol = '°C';

let tempChart = null;

let comparisonChart = null;

let isLoading = false;

let map = null;

let mapMarker = null;

let lastFetched = { current:null, forecast:null };

/\* =======================

   Animations

======================= \*/

const animLayer = document.getElementById('weatherAnimation');

function clearAnimationLayer(){ animLayer.innerHTML = ''; }

function animateCondition(cond){

  clearAnimationLayer();

  cond = cond.toLowerCase();

  if(cond.includes('rain')){

    for(let i=0;i<60;i++){

      const drop = document.createElement('div');

      drop.className = 'raindrop';

      drop.style.left = Math.random() \* 100 + 'vw';

      drop.style.animationDuration = (0.7 + Math.random() \* 0.9) + 's';

      drop.style.opacity = 0.6 + Math.random() \* 0.4;

      animLayer.appendChild(drop);

    }

  } else if(cond.includes('cloud')){

    for(let i=0;i<6;i++){

      const c = document.createElement('div');

      c.className = 'cloud';

      c.style.top = (5 + Math.random() \* 70) + '%';

      c.style.left = (-30 - Math.random()\*20) + 'vw';

      c.style.width = (120 + Math.random()\*120) + 'px';

      c.style.height = (40 + Math.random()\*40) + 'px';

      c.style.animationDuration = (20 + Math.random()\*20) + 's';

      animLayer.appendChild(c);

    }

  }

}

/\* =======================

   Search history

======================= \*/

function loadSearchHistory(){

  const arr = JSON.parse(localStorage.getItem('ws\_history') || '[]');

  renderSearchHistory(arr);

}

function addToSearchHistory(city){

  if(!city) return;

  let arr = JSON.parse(localStorage.getItem('ws\_history') || '[]');

  arr = arr.filter(c => c !== city);

  arr.unshift(city);

  if(arr.length > 6) arr.pop();

  localStorage.setItem('ws\_history', JSON.stringify(arr));

  renderSearchHistory(arr);

}

function renderSearchHistory(arr){

  searchHistoryDiv.innerHTML = '';

  arr.forEach(city => {

    const btn = document.createElement('button');

    btn.className = 'history-btn';

    btn.textContent = city;

    btn.onclick = () => performSearch(city);

    searchHistoryDiv.appendChild(btn);

  });

}

/\* =======================

   Fetch & render

======================= \*/

async function fetchWeather(city){

  try{

    setLoading(true);

    const res = await fetch(`https://api.openweathermap.org/data/2.5/weather?q=${city}&appid=${WEATHER\_API\_KEY}&units=${unit}`);

    const data = await res.json();

    if(data.cod !== 200) throw new Error(data.message);

    const forecastRes = await fetch(`https://api.openweathermap.org/data/2.5/forecast?q=${city}&appid=${WEATHER\_API\_KEY}&units=${unit}`);

    const forecast = await forecastRes.json();

    return { data, forecast };

  } catch(err){

    console.warn("API fetch failed:", err.message);

    alert("Unable to fetch weather for "+city);

    return null;

  }

  finally{ setLoading(false); }

}

async function performSearch(raw){

  const result = await fetchWeather(raw);

  if(!result) return;

  const data = {

    name: result.data.name,

    sys: { country: result.data.sys.country },

    main: { temp: result.data.main.temp, humidity: result.data.main.humidity },

    weather: result.data.weather,

    wind: { speed: result.data.wind.speed },

    uv: getRandomUV(),

    bg: getBackgroundFile(result.data.weather[0].main),

    icon: getIconFile(result.data.weather[0].main)

  };

  const forecast = { list: result.forecast.list };

  showAppIfHidden();

  renderFullUI(data, forecast);

  addToSearchHistory(data.name);

}

function showAppIfHidden(){

  if(welcome) welcome.classList.add('d-none');

  appContent.classList.remove('d-none');

}

/\* =======================

   Auth

======================= \*/

function isAuthed(){ return !!localStorage.getItem('ws\_user'); }

function requireAuth(){ if(!isAuthed()){ authOverlay.classList.remove('d-none'); } }

function finishAuth(user){ localStorage.setItem('ws\_user', JSON.stringify(user)); authOverlay.classList.add('d-none'); }

function signOut(){ localStorage.removeItem('ws\_user'); location.reload(); }

tabSignup?.addEventListener('click', ()=>{ tabSignup.classList.add('active'); tabSignin.classList.remove('active'); document.getElementById('signupForm').classList.remove('d-none'); document.getElementById('signinForm').classList.add('d-none'); });

tabSignin?.addEventListener('click', ()=>{ tabSignin.classList.add('active'); tabSignup.classList.remove('active'); document.getElementById('signinForm').classList.remove('d-none'); document.getElementById('signupForm').classList.add('d-none'); });

signupBtn?.addEventListener('click', ()=>{

  const name = suName.value.trim(); const email = suEmail.value.trim(); const pass = suPass.value;

  if(!name || !email || !pass){ alert('Please fill all fields'); return; }

  localStorage.setItem('ws\_user\_creds', JSON.stringify({ email, pass, name }));

  finishAuth({ name, email });

});

signinBtn?.addEventListener('click', ()=>{

  const email = siEmail.value.trim(); const pass = siPass.value;

  const creds = JSON.parse(localStorage.getItem('ws\_user\_creds')||'{}');

  if(!creds.email){ alert('No account found, please sign up'); return; }

  if(email===creds.email && pass===creds.pass){ finishAuth({ name: creds.name, email }); }

  else{ alert('Invalid credentials'); }

});

signOutBtn?.addEventListener('click', signOut);

/\* =======================

   Render UI

======================= \*/

function renderFullUI(current, forecast){

  document.getElementById('cityName').innerText = `${current.name}, ${current.sys.country}`;

  document.getElementById('heroIcon').src = `images/${current.icon}`;

  animateTemperature(current.main.temp);

  document.getElementById('heroExtra').innerText = `Wind: ${current.wind.speed} ${unit==='metric'?'m/s':'mph'} | Humidity: ${current.main.humidity}% | UV: ${current.uv}`;

  const adv = document.getElementById('advice');

  let t = current.main.temp;

  let adviceText = t < 10 ? "Cold — wear a warm jacket ❄️" : t < 20 ? "Mild — light sweater 🙂" : "Warm — T-shirt 👕";

  adv.innerHTML = `<div class="glass p-3" style="color:#fff">${adviceText}</div>`;

  document.body.style.backgroundImage = `url('images/${current.bg}')`;

  animateCondition(current.weather[0].main);

  const hourEl = document.getElementById('hourlyForecast');

  hourEl.innerHTML = '';

  forecast.list.slice(0,8).forEach(it => {

    const hour = new Date(it.dt\_txt).getHours();

    const card = document.createElement('div');

    card.className = 'card p-2';

    card.innerHTML = `<h6>${hour}:00</h6>

      <img src="images/${getIconFile(it.weather[0].main)}" width="40" alt="icon">

      <p class="mb-0">${formatTemp(it.main.temp)}</p>`;

    hourEl.appendChild(card);

  });

  const daily = forecast.list.filter((\_,i)=> i%8===0);

  const f = document.getElementById('forecast'); f.innerHTML = '';

  daily.forEach(d => {

    const col = document.createElement('div'); col.className = 'col-md-2';

    const c = document.createElement('div'); c.className = 'card p-2';

    c.innerHTML = `<h6>${new Date(d.dt\_txt).toLocaleDateString()}</h6>

      <img src="images/${getIconFile(d.weather[0].main)}" width="40" alt="icon">

      <p class="mb-0">${formatTemp(d.main.temp)}</p>`;

    col.appendChild(c); f.appendChild(col);

  });

  renderTemperatureChart(daily);

  lastFetched.current = current; lastFetched.forecast = daily;

  const lastUpdatedEl = document.getElementById('lastUpdated');

  if(lastUpdatedEl){ lastUpdatedEl.innerText = new Date().toLocaleString(); }

  const heroContainer = document.getElementById('heroContainer');

  heroContainer.classList.remove('fade-in'); void heroContainer.offsetWidth; heroContainer.classList.add('fade-in');

}

/\* =======================

   Loading UI helpers

======================= \*/

function setLoading(state){

  isLoading = state;

  const hero = document.getElementById('hero');

  const heroSk = document.getElementById('heroSkeleton');

  const hourly = document.getElementById('hourlyForecast');

  const hourlySk = document.getElementById('hourlySkeleton');

  const forecast = document.getElementById('forecast');

  const forecastSk = document.getElementById('forecastSkeleton');

  if(state){

    hero.classList.add('d-none'); heroSk.classList.remove('d-none');

    hourly.classList.add('d-none'); hourlySk.classList.remove('d-none');

    forecast.classList.add('d-none'); forecastSk.classList.remove('d-none');

  }else{

    hero.classList.remove('d-none'); heroSk.classList.add('d-none');

    hourly.classList.remove('d-none'); hourlySk.classList.add('d-none');

    forecast.classList.remove('d-none'); forecastSk.classList.add('d-none');

  }

}

/\* =======================

   Temp count-up animation

======================= \*/

function animateTemperature(targetTemp){

  const el = document.getElementById('heroTemp');

  let start = 0;

  const duration = 600;

  const startTime = performance.now();

  function step(now){

    const progress = Math.min(1, (now - startTime) / duration);

    const current = Math.round(start + (targetTemp - start) \* progress);

    el.textContent = formatTemp(current).replace(/\s/g,' ');

    if(progress < 1) requestAnimationFrame(step);

  }

  requestAnimationFrame(step);

}

/\* =======================

   Charts

======================= \*/

function renderTemperatureChart(daily){

  const labels = daily.map(d => new Date(d.dt\_txt).toLocaleDateString());

  const data = daily.map(d => d.main.temp);

  const ctx = document.getElementById('tempChart').getContext('2d');

  if(tempChart) tempChart.destroy();

  tempChart = new Chart(ctx, {

    type: 'line',

    data: { labels, datasets:[{label:`Temp (${unitSymbol})`,data,borderColor:'#34d399',backgroundColor:'rgba(52,211,153,0.18)',fill:true,pointRadius:6,tension:0.35}] },

    options: { responsive:true, plugins:{ tooltip:{ enabled:true }, legend:{ labels:{ color:'#fff' } } }, scales:{ x:{ ticks:{ color:'#fff' } }, y:{ ticks:{ color:'#fff' } } } }

  });

}

async function renderComparisonChartAPI(aCity, bCity){

  const aData = await fetchWeather(aCity);

  const bData = await fetchWeather(bCity);

  if(!aData || !bData) return;

  const dailyA = aData.forecast.list.filter((\_,i)=>i%8===0).map(d=>d.main.temp);

  const dailyB = bData.forecast.list.filter((\_,i)=>i%8===0).map(d=>d.main.temp);

  const labels = aData.forecast.list.filter((\_,i)=>i%8===0).map(d=>new Date(d.dt\_txt).toLocaleDateString());

  const ctx = document.getElementById('comparisonChart').getContext('2d');

  if(comparisonChart) comparisonChart.destroy();

  comparisonChart = new Chart(ctx, {

    type:'bar',

    data:{ labels, datasets:[{label:aData.data.name,data:dailyA,backgroundColor:'rgba(124,58,237,0.8)'},{label:bData.data.name,data:dailyB,backgroundColor:'rgba(59,130,246,0.8)'}] },

    options:{ responsive:true, plugins:{ legend:{ labels:{ color:'#fff' } }, tooltip:{ enabled:true } }, scales:{ x:{ ticks:{ color:'#fff' } }, y:{ ticks:{ color:'#fff' } } } }

  });

}

/\* =======================

   Coords-based flow

======================= \*/

async function fetchWeatherByCoords(lat, lon){

  try{

    setLoading(true);

    const res = await fetch(`https://api.openweathermap.org/data/2.5/weather?lat=${lat}&lon=${lon}&appid=${WEATHER\_API\_KEY}&units=${unit}`);

    const data = await res.json();

    const forecastRes = await fetch(`https://api.openweathermap.org/data/2.5/forecast?lat=${lat}&lon=${lon}&appid=${WEATHER\_API\_KEY}&units=${unit}`);

    const forecast = await forecastRes.json();

    return { data, forecast };

  }catch(e){ console.warn(e); alert('Unable to fetch by coordinates'); return null; }

  finally{ setLoading(false); }

}

async function useCoords(lat, lon){

  const result = await fetchWeatherByCoords(lat, lon);

  if(!result) return;

  const data = {

    name: result.data.name,

    sys: { country: result.data.sys.country },

    main: { temp: result.data.main.temp, humidity: result.data.main.humidity },

    weather: result.data.weather,

    wind: { speed: result.data.wind.speed },

    uv: getRandomUV(),

    bg: getBackgroundFile(result.data.weather[0].main),

    icon: getIconFile(result.data.weather[0].main)

  };

  const forecast = { list: result.forecast.list };

  showAppIfHidden();

  renderFullUI(data, forecast);

  addToSearchHistory(`${lat.toFixed(2)},${lon.toFixed(2)}`);

}

/\* =======================

   Map & Places

======================= \*/

function initMap(){

  map = L.map('map');

L.tileLayer('https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png', { maxZoom: 19, attribution:'' }).addTo(map);

  const defaultView = [20.5937, 78.9629]; // India center fallback

  map.setView(defaultView, 4);

  map.on('click', (e)=>{

    const { lat, lng } = e.latlng; setLatLonInputs(lat, lng); placeMarker(lat, lng); useCoords(lat, lng);

  });

  if(navigator.geolocation){ navigator.geolocation.getCurrentPosition(pos=>{ const { latitude, longitude } = pos.coords; map.setView([latitude, longitude], 10); placeMarker(latitude, longitude); setLatLonInputs(latitude, longitude); }); }

}

function placeMarker(lat, lon){

  if(mapMarker){ map.removeLayer(mapMarker); }

  mapMarker = L.marker([lat, lon]).addTo(map);

}

function setLatLonInputs(lat, lon){

  document.getElementById('latInput').value = lat.toFixed(6);

  document.getElementById('lonInput').value = lon.toFixed(6);

}

async function geocodePlace(q){

  const res = await fetch(`https://nominatim.openstreetmap.org/search?format=json&q=${encodeURIComponent(q)}`);

  const json = await res.json();

  return json?.[0];

}

/\* =======================

   Event wiring

======================= \*/

startSearchBtn?.addEventListener('click', () => {

  const v = cityInputHome.value.trim();

  if(!v){ alert('Type a city'); return; }

  performSearch(v);

});

// Enter shortcuts

cityInputHome?.addEventListener('keydown', (e)=>{ if(e.key==='Enter'){ startSearchBtn.click(); }});

cityInput.addEventListener('keydown', (e)=>{ if(e.key==='Enter'){ searchBtn.click(); }});

document.getElementById('compareCityA').addEventListener('keydown', (e)=>{ if(e.key==='Enter'){ document.getElementById('compareBtn').click(); }});

document.getElementById('compareCityB').addEventListener('keydown', (e)=>{ if(e.key==='Enter'){ document.getElementById('compareBtn').click(); }});

searchBtn.addEventListener('click', ()=> {

  const v = cityInput.value.trim();

  if(!v){ alert('Type a city'); return; }

  performSearch(v);

});

unitToggle.addEventListener('click', ()=>{

  unit = unit === 'metric' ? 'imperial' : 'metric';

  unitSymbol = unit === 'metric' ? '°C' : '°F';

  unitToggle.innerText = unit === 'metric' ? 'Show °F' : 'Show °C';

  const last = JSON.parse(localStorage.getItem('ws\_history') || '[]')[0];

  if(last) performSearch(last);

});

geoBtn.addEventListener('click', () => {

  if(!navigator.geolocation){ alert('Geolocation not supported'); return; }

  navigator.geolocation.getCurrentPosition(pos => {

    const lat = pos.coords.latitude, lon = pos.coords.longitude;

    // fallback: nearest sample city

    let pick = 'newyork';

    if(lon > 60 && lon < 100 && lat > 6 && lat < 40) pick = 'bangalore';

    else if(lon > -15 && lon < 10) pick = 'london';

    else if(lon > 120 && lon < 150) pick = 'tokyo';

    performSearch(pick);

  }, ()=> alert('Unable to get location'));

});

document.getElementById('compareBtn').addEventListener('click', ()=>{

  const a = document.getElementById('compareCityA').value;

  const b = document.getElementById('compareCityB').value;

  if(!a || !b){ alert('Enter both cities to compare'); return; }

  renderComparisonChartAPI(a,b);

});

// map/places wiring

const placeSearch = document.getElementById('placeSearch');

const useCoordsBtn = document.getElementById('useCoordsBtn');

placeSearch?.addEventListener('keydown', async (e)=>{

  if(e.key==='Enter'){

    const q = placeSearch.value.trim(); if(!q) return;

    const p = await geocodePlace(q);

    if(!p){ alert('Place not found'); return; }

    const lat = parseFloat(p.lat), lon = parseFloat(p.lon);

    setLatLonInputs(lat, lon); map.setView([lat, lon], 11); placeMarker(lat, lon); useCoords(lat, lon);

  }

});

useCoordsBtn?.addEventListener('click', ()=>{

  const lat = parseFloat(document.getElementById('latInput').value);

  const lon = parseFloat(document.getElementById('lonInput').value);

  if(Number.isFinite(lat) && Number.isFinite(lon)) useCoords(lat, lon);

});

/\* =======================

   Theme toggle

======================= \*/

function applyTheme(theme){

  if(theme==='light'){

    document.body.classList.add('light');

    themeToggle.textContent = 'Dark Mode';

  }else{

    document.body.classList.remove('light');

    themeToggle.textContent = 'Light Mode';

  }

}

const savedTheme = localStorage.getItem('ws\_theme') || 'dark';

applyTheme(savedTheme);

themeToggle?.addEventListener('click', ()=>{

  const next = document.body.classList.contains('light') ? 'dark' : 'light';

  localStorage.setItem('ws\_theme', next);

  applyTheme(next);

});

/\* =======================

   Init

======================= \*/

loadSearchHistory();

// footer year

const fy = document.getElementById('footerYear');

if(fy){ fy.textContent = new Date().getFullYear(); }

// initialize features

requireAuth();

if(document.getElementById('map')){ initMap(); }

// toggle history dropdown

historyToggle?.addEventListener('click', ()=>{

  searchHistoryDiv.classList.toggle('d-none');

});

/\* =======================

   Download Report (PDF)

======================= \*/

document.getElementById('downloadBtn')?.addEventListener('click', ()=>{

  if(!lastFetched.current || !lastFetched.forecast){ alert('Search first to generate a report'); return; }

  const { jsPDF } = window.jspdf;

  const doc = new jsPDF();

  const cur = lastFetched.current;

  doc.setFontSize(16);

  doc.text(`Weather Report — ${cur.name}, ${cur.sys.country}`, 14, 18);

  doc.setFontSize(11);

  doc.text(`Temperature: ${formatTemp(cur.main.temp)} | Humidity: ${cur.main.humidity}% | Wind: ${cur.wind.speed} ${unit==='metric'?'m/s':'mph'}`, 14, 28);

  doc.text(`Condition: ${cur.weather?.[0]?.main || ''}`, 14, 36);

  doc.text('5-Day Outlook:', 14, 48);

  let y = 56;

  lastFetched.forecast.forEach(d => {

    const line = `${new Date(d.dt\_txt).toLocaleDateString()}  —  ${formatTemp(d.main.temp)}  (${d.weather?.[0]?.main || ''})`;

    doc.text(line, 14, y);

    y += 8;

  });

  doc.save(`WeatherReport\_${cur.name}.pdf`);

});