```
<!DOCTYPE html>
<html lang="en">
<head>
 <meta charset="UTF-8" />
 <title>Shortest Path Navigation System</title>

 <!-- Leaflet -->
 <link rel="stylesheet" href="https://unpkg.com/leaflet/dist/leaflet.css"/>
 <script src="https://unpkg.com/leaflet/dist/leaflet.js"></script>

 <style>
  :root {
   --bg: radial-gradient(1200px 600px at 10% 10%, #0b1220, #020617);
   --panel: linear-gradient(180deg, #0f172a, #020617);
   --text: #e5e7eb;
   --accent: #38bdf8;
   --muted: #94a3b8;
  }
  body.light {
   --bg: linear-gradient(135deg, #f8fafc, #e2e8f0);
   --panel: #ffffff;
   --text: #020617;
   --accent: #0284c7;
   --muted: #475569;
  }
  * { box-sizing: border-box; }
  body {
   margin: 0;
   min-height: 100vh;
   font-family: Inter, Arial, sans-serif;
   background: var(--bg);
```

```
  color: var(--text);
}
.shell {
  width: 100vw;
  height: 100vh;
  display: grid;
  grid-template-columns: 360px 1fr;
  gap: 12px;
  padding: 12px;
}
.panel {
  background: var(--panel);
  border-radius: 16px;
  padding: 14px;
  box-shadow: 0 20px 40px rgba(56,189,248,.15),
              inset 0 0 0 1px rgba(148,163,184,.12);
}
.controls { display: flex; flex-direction: column; gap: 12px; }
h2 { margin: 0; text-align: center; color: var(--accent); }
.section {
  background: rgba(2,6,23,.6);
  border: 1px solid rgba(148,163,184,.12);
  border-radius: 12px;
  padding: 12px;
}
body.light .section { background: #f8fafc; }
.grid { display: grid; grid-template-columns: 1fr 1fr 1fr; gap: 8px; }
input, button {
  width: 100%;
  padding: 10px 12px;
  border-radius: 10px;
```

```
  border: 1px solid rgba(148,163,184,.2);
  background: #020617;
  color: var(--text);
  outline: none;
}
body.light input, body.light button {
  background: #f1f5f9;
  color: #020617;
}
.toolbar { display: flex; gap: 8px; margin-top: 8px; }
button.primary {
  background: linear-gradient(135deg, #38bdf8, #22d3ee);
  color: #020617;
  font-weight: 700;
  border: none;
  cursor: pointer;
}
.output {
  background: #020617;
  border-radius: 10px;
  padding: 10px;
  min-height: 60px;
  color: #a5f3fc;
  border: 1px dashed rgba(148,163,184,.25);
  font-size: 14px;
}
.cities {
  max-height: 140px;
  overflow: auto;
  border: 1px solid rgba(148,163,184,.12);
  border-radius: 10px;
```

```
    padding: 8px;
    font-size: 13px;
    color: var(--muted);
   }
   #map { width: 100%; height: calc(100vh - 32px); border-radius: 16px; }
  </style>
</head>
<body>

<div class="shell">
  <div class="panel controls">
   <h2>Shortest Path Navigation</h2>

   <div class="section">
    <div class="grid">
     <input id="c1" placeholder="City/State 1" />
     <input id="c2" placeholder="City/State 2" />
     <input id="dist" placeholder="Distance" />
    </div>
    <div class="toolbar">
     <button class="primary" onclick="addRoad()">Add Road</button>
     <button onclick="clearAll()">Clear</button>
    </div>
   </div>

   <div class="section">
    <div class="grid">
     <input id="src" placeholder="Source" />
     <input id="dest" placeholder="Destination" />
     <button class="primary" onclick="findPath()">Find Path</button>
    </div>
```

```
      <div class="toolbar">
        <button onclick="showCities()">Show Nodes</button>
        <button onclick="toggleTheme()">Theme</button>
      </div>
    </div>

    <div class="output" id="output">Add roads to start.</div>
    <div class="cities" id="citiesBox"></div>
  </div>

  <div class="panel">
    <div id="map"></div>
  </div>
</div>

<script>
  const cityIndex = new Map();
  const indexToCity = [];
  const graph = [];
  let markers = [];
  let polylines = [];
  let map;

  // All Indian States + UTs (capital coordinates)
  const cityCoords = {
    "andhra pradesh": [16.5062, 80.6480],     // Amaravati
    "arunachal pradesh": [27.0844, 93.6053],  // Itanagar
    "assam": [26.1445, 91.7362],              // Dispur
    "bihar": [25.5941, 85.1376],              // Patna
    "chhattisgarh": [21.2514, 81.6296],       // Raipur
    "goa": [15.4909, 73.8278],                // Panaji
```

"gujarat": [23.2156, 72.6369], // Gandhinagar

"haryana": [30.7333, 76.7794], // Chandigarh

"himachal pradesh": [31.1048, 77.1734], // Shimla

"jharkhand": [23.3441, 85.3096], // Ranchi

"karnataka": [12.9716, 77.5946], // Bengaluru

"kerala": [8.5241, 76.9366], // Thiruvananthapuram

"madhya pradesh": [23.2599, 77.4126], // Bhopal

"maharashtra": [19.0760, 72.8777], // Mumbai

"manipur": [24.8170, 93.9368], // Imphal

"meghalaya": [25.5788, 91.8933], // Shillong

"mizoram": [23.7271, 92.7176], // Aizawl

"nagaland": [25.6747, 94.1100], // Kohima

"odisha": [20.2961, 85.8245], // Bhubaneswar

"punjab": [30.7333, 76.7794], // Chandigarh

"rajasthan": [26.9124, 75.7873], // Jaipur

"sikkim": [27.3389, 88.6065], // Gangtok

"tamil nadu": [13.0827, 80.2707], // Chennai

"telangana": [17.3850, 78.4867], // Hyderabad

"tripura": [23.8315, 91.2868], // Agartala

"uttar pradesh": [26.8467, 80.9462], // Lucknow

"uttarakhand": [30.3165, 78.0322], // Dehradun

"west bengal": [22.5726, 88.3639], // Kolkata

"delhi": [28.6139, 77.2090],

"jammu and kashmir": [34.0837, 74.7973],

"ladakh": [34.1526, 77.5771],

"chandigarh": [30.7333, 76.7794],

"puducherry": [11.9416, 79.8083],

"andaman and nicobar islands": [11.6234, 92.7265],

"dadra and nagar haveli and daman and diu": [20.3974, 72.8328],

"lakshadweep": [10.5667, 72.6417]

```
};

function initMap() {
  map = L.map('map').setView([22.5, 78.9], 5);
  L.tileLayer('https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png', {
    attribution: '&copy; OpenStreetMap contributors'
  }).addTo(map);
}
initMap();

function toggleTheme() { document.body.classList.toggle("light"); }
function cap(s) { return s.replace(/\b\w/g, c => c.toUpperCase()); }
function output(msg) { document.getElementById("output").innerHTML = msg; }

function getCityIndex(name) {
  name = name.toLowerCase();
  if (!cityIndex.has(name)) {
    cityIndex.set(name, indexToCity.length);
    indexToCity.push(name);
    graph.push([]);
    const [lat, lng] = cityCoords[name] || [22.5, 78.9];
    const marker = L.marker([lat, lng]).addTo(map).bindPopup(cap(name));
    markers.push(marker);
  }
  return cityIndex.get(name);
}

function addRoad() {
  const c1 = document.getElementById("c1").value.trim();
  const c2 = document.getElementById("c2").value.trim();
  const w = parseInt(document.getElementById("dist").value, 10);
```

```
  if (!c1 || !c2 || isNaN(w) || w < 0) return output("Enter valid nodes & distance.");
  const u = getCityIndex(c1);
  const v = getCityIndex(c2);
  graph[u].push([v, w]);
  graph[v].push([u, w]);
  output(`Road added: ${cap(c1)} ↔ ${cap(c2)} (${w})`);
}

function dijkstra(src) {
  const n = graph.length;
  const dist = Array(n).fill(Infinity);
  const parent = Array(n).fill(-1);
  const used = Array(n).fill(false);
  dist[src] = 0;
  const pq = [[0, src]];
  while (pq.length) {
    pq.sort((a,b) => a[0] - b[0]);
    const [d, u] = pq.shift();
    if (used[u]) continue;
    used[u] = true;
    for (const [v, w] of graph[u]) {
      if (dist[u] + w < dist[v]) {
        dist[v] = dist[u] + w;
        parent[v] = u;
        pq.push([dist[v], v]);
      }
    }
  }
  return { dist, parent };
}
```

```
function findPath() {
  const srcCity = document.getElementById("src").value.trim().toLowerCase();
  const destCity = document.getElementById("dest").value.trim().toLowerCase();
  if (!cityIndex.has(srcCity) || !cityIndex.has(destCity)) return output("Node not found.");
  const src = cityIndex.get(srcCity);
  const dest = cityIndex.get(destCity);
  const { dist, parent } = dijkstra(src);
  if (!isFinite(dist[dest])) return output("No path exists.");
  let path = [];
  for (let v = dest; v !== -1; v = parent[v]) path.push(v);
  path.reverse();

  polylines.forEach(p => map.removeLayer(p));
  polylines = [];
  for (let i = 0; i < path.length - 1; i++) {
    const p1 = markers[path[i]].getLatLng();
    const p2 = markers[path[i+1]].getLatLng();
    polylines.push(L.polyline([p1, p2], { weight: 4 }).addTo(map));
  }
  output(`Distance: ${dist[dest]}<br>Path: ${path.map(i => cap(indexToCity[i])).join(" → ")}`);
}

function showCities() {
  if (!indexToCity.length) return output("No nodes yet.");
  document.getElementById("citiesBox").innerHTML = indexToCity.map(cap).join(", ");
}

function clearAll() {
  cityIndex.clear();
  indexToCity.length = 0;
  graph.length = 0;
```

```
    markers.forEach(m => map.removeLayer(m));
    polylines.forEach(p => map.removeLayer(p));
    markers = [];
    polylines = [];
    document.getElementById("citiesBox").innerHTML = "";
    output("Cleared all data.");
  }
</script>

</body>
</html>
```