**Multi-Dimensional Data Representation: Design a multi-dimensional array to represent geographical data (e.g., latitude, longitude, altitude) for a drone-based mapping system. Implement functions to extract and manipulate data for different regions dynamically.**

<html>

<head>

<title>Drone Mapping System</title>

<style>

body {

font-family: Arial, sans-serif;

}

table {

margin: 20px 0;

border-collapse: collapse;

}

th, td {

border: 1px solid #000;

padding: 10px;

text-align: center;

}

input {

margin: 5px;

}

</style>

</head>

<body>

<h1>Drone-Based Mapping System</h1>

<h2>Add Geographical Data</h2>

<form id="dataForm">

<input type="number" id="latitude" placeholder="Latitude" step="any" required>

<input type="number" id="longitude" placeholder="Longitude" step="any" required>

<input type="number" id="altitude" placeholder="Altitude (meters)" step="any" required>

<button type="submit">Add Data</button>

</form>

<h2>Geographical Data</h2>

<table id="dataTable">

<thead>

<tr>

<th>Latitude</th>

<th>Longitude</th>

<th>Altitude (m)</th>

</tr>

</thead>

<tbody id="dataBody"></tbody>

</table>

<h2>Extract Data for Region</h2>

<form id="regionForm">

<input type="number" id="latMin" placeholder="Min Latitude" step="any" required>

<input type="number" id="latMax" placeholder="Max Latitude" step="any" required>

<input type="number" id="longMin" placeholder="Min Longitude" step="any" required>

<input type="number" id="longMax" placeholder="Max Longitude" step="any" required>

<button type="submit">Extract Data</button>

</form>

<h2>Extracted Data</h2>

<div id="extractedData"></div>

<script>

let geographicalData = []; // Multi-dimensional array to store geographical data

// Function to add geographical data

function addGeographicalData(latitude, longitude, altitude) {

const dataPoint = [latitude, longitude, altitude];

geographicalData.push(dataPoint);

renderData();

}

// Function to render the geographical data table

function renderData() {

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

dataBody.innerHTML = ''; // Clear existing entries

geographicalData.forEach(data => {

const row = document.createElement('tr');

row.innerHTML = `

<td>${data[0]}</td>

<td>${data[1]}</td>

<td>${data[2]}</td>

`;

dataBody.appendChild(row);

});

}

// Function to extract data for a specific region

function extractData(latMin, latMax, longMin, longMax) {

const filteredData = geographicalData.filter(data =>

data[0] >= latMin && data[0] <= latMax &&

data[1] >= longMin && data[1] <= longMax

);

return filteredData;

}

// Function to display extracted data

function displayExtractedData(extracted) {

const extractedDataDiv = document.getElementById('extractedData');

extractedDataDiv.innerHTML = '';

if (extracted.length === 0) {

extractedDataDiv.innerText = 'No data found for the specified region.';

return;

}

const table = document.createElement('table');

const headerRow = document.createElement('tr');

headerRow.innerHTML = `

<th>Latitude</th>

<th>Longitude</th>

<th>Altitude (m)</th>

`;

table.appendChild(headerRow);

extracted.forEach(data => {

const row = document.createElement('tr');

row.innerHTML = `

<td>${data[0]}</td>

<td>${data[1]}</td>

<td>${data[2]}</td>

`;

table.appendChild(row);

});

extractedDataDiv.appendChild(table);

}

// Event listener for adding geographical data

document.getElementById('dataForm').addEventListener('submit', function(event) {

event.preventDefault();

const latitude = parseFloat(document.getElementById('latitude').value);

const longitude = parseFloat(document.getElementById('longitude').value);

const altitude = parseFloat(document.getElementById('altitude').value);

addGeographicalData(latitude, longitude, altitude);

this.reset();

});

// Event listener for extracting data for a specific region

document.getElementById('regionForm').addEventListener('submit', function(event) {

event.preventDefault();

const latMin = parseFloat(document.getElementById('latMin').value);

const latMax = parseFloat(document.getElementById('latMax').value);

const longMin = parseFloat(document.getElementById('longMin').value);

const longMax = parseFloat(document.getElementById('longMax').value);

const extracted = extractData(latMin, latMax, longMin, longMax);

displayExtractedData(extracted);

});

</script>

</body>

</html>