**map.component.ts :**

**import { Component, Input, OnInit, ViewChild, ElementRef } from '@angular/core';**

**import \* as L from 'leaflet';**

**import { GeoLocationService } from 'src/app/Services/geo-location.service';**

**import { GeoSearchControl, OpenStreetMapProvider } from 'leaflet-geosearch';**

**@Component({**

**selector: 'app-map',**

**templateUrl: './map.component.html',**

**styleUrls: ['./map.component.css'],**

**})**

**export class MapComponent implements OnInit {**

**@Input() quoteDescription: string = ''; // Modify input to accept single string**

**searchQuery: string = '';**

**markerIconUrl = './marker-icon.png'; // Updated marker icon URL**

**@ViewChild('map', { static: true }) mapContainer!: ElementRef;**

**private map!: L.Map;**

**private centroid: L.LatLngExpression = [34, 9]; // Adjusted centroid coordinates**

**private searchResult: any;**

**constructor(private geoLocationService: GeoLocationService) { }**

**ngOnInit(): void {**

**this.initMap();**

**this.searchLocation(this.quoteDescription);**

**// this.geoLocationService.location$.subscribe((location) => {**

**// if (location.loaded && !location.error) {**

**// const { lat, lng } = location.coordinates;**

**// console.log('User Location:', lat, lng); // Log user's location**

**// this.map.flyTo([lat, lng], 10);**

**// // Add custom marker for user's location**

**// L.marker([lat, lng], {**

**// icon: L.icon({**

**// iconUrl: this.markerIconUrl, // Use the custom marker icon**

**// iconSize: [40, 40],**

**// iconAnchor: [20, 40], // Adjusted anchor point**

**// popupAnchor: [0, -40], // Adjusted popup anchor point**

**// }),**

**// }).addTo(this.map);**

**// } else {**

**// console.error('Error getting user location:', location.error);**

**// }**

**// });**

**}**

**private searchLocation(description: string): void {**

**const provider = new OpenStreetMapProvider();**

**provider.search({ query: description })**

**.then(results => {**

**if (results.length > 0) {**

**const result = results[0];**

**// Handle search result**

**console.log('Search result:', result);**

**this.map.setView([result.y, result.x], this.map.getZoom());**

**L.marker([result.y, result.x], {**

**icon: L.icon({**

**iconUrl: this.markerIconUrl,**

**iconSize: [40, 40],**

**iconAnchor: [20, 40],**

**popupAnchor: [0, -40],**

**}),**

**}).addTo(this.map);**

**} else {**

**console.log('No results found for:', description);**

**}**

**})**

**.catch(error => {**

**console.error('Error performing search:', error);**

**});**

**}**

**private initMap(): void {**

**this.map = L.map(this.mapContainer.nativeElement, {**

**center: this.centroid,**

**zoom: 6,**

**});**

**const tiles = L.tileLayer('https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png', {**

**maxZoom: 18,**

**minZoom: 6,**

**attribution: '&copy; <a href="https://www.openstreetmap.org/copyright">OpenStreetMap</a> contributors'**

**});**

**const provider = new OpenStreetMapProvider();**

**const searchLabelValue = 'la marsa';**

**const searchControl = new (GeoSearchControl as any)({**

**provider: provider,**

**style: 'bar',**

**autoClose: true,**

**searchLabel: searchLabelValue,**

**showMarker: true,**

**retainZoomLevel: true,**

**animateZoom: true,**

**keepResult: true,**

**updateMap: true,**

**popupFormat: ({ query, result }: { query: string; result: any }) => result.label,**

**maxMarkers: 1,**

**marker: {**

**icon: L.icon({**

**iconUrl: this.markerIconUrl,**

**iconSize: [40, 40],**

**iconAnchor: [20, 40],**

**popupAnchor: [0, -40],**

**}),**

**},**

**}).addTo(this.map);**

**// Trigger search programmatically**

**provider.search({ query: searchLabelValue })**

**.then(results => {**

**if (results.length > 0) {**

**const result = results[0];**

**// Handle search result**

**console.log('Search result:', result);**

**this.map.setView([result.y, result.x], this.map.getZoom());**

**L.marker([result.y, result.x], {**

**icon: L.icon({**

**iconUrl: this.markerIconUrl,**

**iconSize: [40, 40],**

**iconAnchor: [20, 40],**

**popupAnchor: [0, -40],**

**}),**

**}).addTo(this.map);**

**} else {**

**console.log('No results found for:', searchLabelValue);**

**}**

**})**

**.catch(error => {**

**console.error('Error performing search:', error);**

**});**

**tiles.addTo(this.map);**

**}**

**// locateUserPosition(): void {**

**// this.geoLocationService.location$.subscribe((location) => {**

**// if (location.loaded && !location.error) {**

**// const { lat, lng } = location.coordinates;**

**// this.map.flyTo([lat, lng], 10);**

**// } else {**

**// alert('Unable to locate your position.');**

**// }**

**// });**

**// }**

**// async handleSearchButtonClick(): Promise<void> {**

**// if (!this.searchQuery) return;**

**// const provider = new OpenStreetMapProvider();**

**// try {**

**// const results = await provider.search({ query: this.searchQuery });**

**// if (results.length > 0) {**

**// const result = results[0];**

**// console.log('Search result:', result);**

**// }**

**// } catch (error) {**

**// console.error('Error performing search:', error);**

**// }**

**// }**

**}**

**get-quotes.component.html :**

<h1>Quotes</h1>

<table class="table table-striped">

  <thead>

  <tr>

    <th>description</th>

    <th>issuanceDate</th>

    <th>quantity</th>

    <th>totalamount</th>

    <th>unitprice</th>

    <th>Location</th> <!-- Add a new column for the map -->

  </tr>

  </thead>

  <tbody>

  <tr \*ngFor="let quote of quotes ">

    <td>{{quote.description}}</td>

    <td>{{quote.issuanceDate}}</td>

    <td>{{quote.quantity}}</td>

    <td>{{quote.totalamount}}</td>

    <td>{{quote.unitprice}}</td>

    <app-map [quoteDescription]="quote.description"></app-map>

    <td>

      <button class="btn btn-danger" (click)="deleteQuote(quote.quote\_id)">Delete</button>

    </td>

    <td>

      <button class="btn btn-primary" (click)="updateQuote(quote.quote\_id)">Update</button>

    </td>

  </tr>

  </tbody>

  <button class="btn btn-success float-right" (click)="navigateToAddQuote()">Add Quote</button>

</table>