

Search Box With auto Complete feature for location and Getting the Co-ordinates of the Pickup and drop location using MapBox

Autocomplete.html :

```
<mat-form-field class="example-full-width" appearance="outline">
  <input matInput [formControl]="myControl" [matAutocomplete]="auto" />
  <mat-autocomplete #auto="matAutocomplete" [displayWith]="displayFn"
(optionSelected)="onOptionSelected($event.option.value)">
    <mat-option
      *ngFor="let option of filteredOptions | async"
      [value]="option"
      >{{ option.place_name }}</mat-option
    </mat-option>
  </mat-autocomplete>
</mat-form-field>
```

autocomplete.component.ts

```
import { AsyncPipe, CommonModule } from '@angular/common';
import { Component, EventEmitter, Output } from '@angular/core';
import { FormControl, FormsModule, ReactiveFormsModule } from
 '@angular/forms';
import { MatAutocompleteModule } from '@angular/material/autocomplete';
import { MatFormFieldModule } from '@angular/material/form-field';
import { MatInputModule } from '@angular/material/input';
import { Feature, MapboxService } from
 '../.../core/services/mapbox.service';
import { debounce, debounceTime, map, Observable, of, startWith,
switchMap, timer } from 'rxjs';

@Component({
  selector: 'app-autocomplete',
  standalone: true,
  imports:
[MatFormFieldModule, MatInputModule, MatAutocompleteModule, AsyncPipe, Reactiv
eFormsModule, CommonModule ],
  templateUrl: './autocomplete.component.html',
  styleUrls: ['./autocomplete.component.css']
})
```

```

export class AutocompleteComponent {
  myControl = new FormControl<string>('');
  options: Feature[] = [];
  filteredOptions!: Observable<Feature[]>;

  @Output() coordinatesSelected = new EventEmitter<[number, number]>();

  constructor(private mapboxService : MapboxService){

  }
  ngOnInit(): void {
    this.filteredOptions = this.myControl.valueChanges.pipe(
      startWith(''), // emits the default value
      map(value => this._filter(value || ''))
    );
    // .pipe(debounce(()=>timer(1000)))
    this.myControl.valueChanges.subscribe(searchTerm =>{
      // console.log(searchTerm)
      if (searchTerm && searchTerm.length > 0) {
        this.mapboxService
          .search_word(searchTerm)
          .subscribe((features: Feature[]) => {
            this.options = features
          });
      } else {
        this.options = [];
      }
    });
  }

  private _filter(value : string) : Feature[] {
    const filterValue = value.toString().toLowerCase() || '';

    return this.options.filter(option =>
      option.place_name.toLowerCase().includes(filterValue)
    );
  }

  displayFn(option : Feature){
    return option ? option.place_name : '';
  }
}

```

```

    }

    onOptionSelected(option: Feature) {
      if (option && option.geometry && option.geometry.coordinates) {
        this.coordinatesSelected.emit(option.geometry.coordinates);
      }
    }
  }
}

```

Home.component.html :

```

<div class="container">
  <div class="booking-section">
    <div class="form-header">
      <p style="font-size: 30px; line-height: 10px"><U>Book a cab</U></p>
    </div>
    <div class="form-container">
      <mat-label>Pickup Location</mat-label>
      <!-- inp 1 -->
      <app-autocomplete
(coordinatesSelected)="onPickupCoordinates($event)" />
      <mat-label>Drop Location</mat-label>
      <!-- Inp 2 -->
      <app-autocomplete
(coordinatesSelected)="onDropCoordinates($event)" />
      <button mat-fab class="btn" (click)="onCheckExpense()">Check
Expense</button>
    </div>
  </div>
  <div class="img-section">
    
  </div>
</div>

```

Home.component.ts :

```

import { Component, OnDestroy, OnInit } from '@angular/core';
import { MatButtonModule } from '@angular/material/button';
import { MatFormFieldModule } from '@angular/material/form-field';
import { Subscription, throwError } from 'rxjs';

```

```

import { AutocompleteComponent } from
'../autocomplete/autocomplete.component';
import { FareService } from '../services/fare.service';
import { Router } from '@angular/router';
import { fare } from '../core/models/fareData';

@Component({
  selector: 'app-home',
  standalone: true,
  imports: [MatFormFieldModule, AutocompleteComponent, MatButtonModule],
  templateUrl: './home.component.html',
  styleUrls: ['./home.component.css']
})
export class HomeComponent implements OnDestroy{
  fareSubscription : Subscription | undefined;
  pickupCoordinates: [number, number] | null = null;
  dropCoordinates: [number, number] | null = null;

  constructor(private fareService : FareService, private router : Router){

  }

  onPickupCoordinates(coordinates: [number, number]) {
    this.pickupCoordinates = coordinates;
  }

  onDropCoordinates(coordinates: [number, number]) {
    this.dropCoordinates = coordinates;
  }

  onCheckExpense() {
    console.log(this.pickupCoordinates)
    console.log(this.dropCoordinates)
    if(this.pickupCoordinates != null && this.dropCoordinates != null){
      const pickupCoordinatesStr = this.pickupCoordinates[1] + "," +
this.pickupCoordinates[0];
      const dropCoordinatesStr = this.dropCoordinates[1] + "," +
this.dropCoordinates[0];

```

```

        this.fareSubscription =
this.fareService.checkExpense(pickupCoordinatesStr,dropCoordinatesStr).sub
scribe((res : fare)=>{
    console.log(res);
    if(res){
        this.router.navigate(['/customer/fare'],{state : {fareData :
res}}})
    }
    })
    }else{
        throwError(()=> "Please Enter Pickup And Drop Location")
    }
}

ngOnDestroy(): void {
    this.fareSubscription?.unsubscribe()
}
}

```

Map With route Service using MapBox :

Install : `npm i mapbox-gl`

Add Style In angular.json : `"node_modules/mapbox-gl/dist/mapbox-gl.css"`

.ts file :

```

import { HttpClient } from '@angular/common/http';
import { Component, OnInit } from '@angular/core';
import * as mapboxgl from 'mapbox-gl';

@Component({
  selector: 'app-map',
  templateUrl: './test.component.html',
  styleUrls: ['./test.component.css'],
})
export class TestComponent implements OnInit {
  map!: mapboxgl.Map ;
  directionsService =
'https://api.mapbox.com/directions/v5/mapbox/driving/';
  style = 'mapbox://styles/mapbox/streets-v11';
  lat = 37.75;
  lng = -122.41;
}

```

```

mapboxToken: string =

'pk.eyJ1Ijoia3Jpc2huYWtoYWRRrZS1nZHRjIiwiYSI6ImNsemF5NXlwbzBza2cyanM4eGpoc2hjNTUifQ.O4YVe1XQM3GcGkFa3HeXvQ';

sourceLat = 19.075784;
sourceLng = 72.9952364;

destLat = 19.025773;
destLng = 73.0591845;

constructor(private http : HttpClient){

}

ngOnInit(): void {
    this.initializeMap();
}

initializeMap(): void {
    this.map = new mapboxgl.Map({
        accessToken: this.mapboxToken,
        container: 'map',
        style: this.style,
        zoom: 13,
        center: [this.sourceLng, this.sourceLat],
    });

    this.map.on('style.load', ()=>{
        this.addRoute(
            [this.sourceLng, this.sourceLat],
            [this.destLng, this.destLat]
        );
    })

    const nav = new mapboxgl.NavigationControl();
    this.map.addControl(nav, 'top-right');

    // Optional: Add a marker for the source location
    new mapboxgl.Marker()

```

```

        .setLngLat([this.sourceLng, this.sourceLat])
        .addTo(this.map);

        new mapboxgl.Marker({color : 'red'})
        .setLngLat([this.destLng, this.destLat])
        .addTo(this.map);
    }
    addRoute(start: [number, number], end: [number, number]) {
        const url =
`$${this.directionsService}${start.join(',')};${end.join(',')}?geometries=geojson&access_token=${this.mapboxToken}`;

        this.http.get(url).subscribe((data: any) => {
            const route = data.routes[0].geometry;

            this.map.addSource('route', {
                type: 'geojson',
                data: {
                    type: 'Feature',
                    geometry: route,
                    properties : {}
                }
            });

            this.map.addLayer({
                id: 'route',
                type: 'line',
                source: 'route',
                layout: {
                    'line-join': 'round',
                    'line-cap': 'round'
                },
                paint: {
                    'line-color': '#3887be',
                    'line-width': 5
                }
            });
        });
    }
}

```

.html file :

```
<div id="map" style="width: 100%; height: 50%;"></div>
```