search ts

```
import { WeatherService } from './../../services/weather.service';
import { Component, OnInit, ViewChild, Output, EventEmitter } from
'@angular/core';
import { ISearchResult } from '../../models/IWeatherData.interface';

@Component({
  selector: 'app-search',
  templateUrl: './search.component.html',
  styleUrls: ['./search.component.css']
})
export class SearchComponent implements OnInit {

    @ViewChild('searchInput') searchInput;
    searchResults: ISearchResult[];
    @Output() selectedCity;

    constructor(
        private weatherService: WeatherService,
    ) { }
    ngOnInit() {
    }

    search(term) {
        /*
            CHALLENGE
                - if user has typed something in the input field,
                  call weatherService.searchLocation() with the searched term
                  and assign the results to searchResults array
                - if input field is empty, clear the searResults array
        */
       if(term==''){
         this.searchResults=null;
       }
       else
       this.weatherService.searchLocation(term).subscribe(data=>{
         this.searchResults=data;
       });
    }

    selectedLocation(cityDetails: ISearchResult) {
        /*
            CHALLENGE
              After user clicked on a city name from the search results, this
function should be called.
              This function should perform the following actions
              - make the input field empty
              - clear all the results
              - send the cityid (woeid) to the parent component (AppComponent)
        */
       if(cityDetails.woeid.toString()!=""){
       this.weatherService.searchLocation(cityDetails.woeid);
       this.searchResults=null;
      // this.selectedCity.emit(1);
       }
       this.selectedCity.emit(1);
    }
}
```

search html

```
<p>
  <mat-form-field>
    <mat-label>Search city</mat-label>
    <mat-icon matSuffix>search</mat-icon>

    <!-- Challenge #searchInput -->
    <input
        #searchInput
        matInput
        placeholder="paris, london..."
        (keyup)="search('paris')"
        class="searchInput">
  </mat-form-field>
</p>

<div class="searchOutput">
    <mat-card *ngIf="searchResults">
        <!-- If input field has some value and the result from service is an
empty array show sorry message else display list -->
        <mat-list role="list" *ngIf="searchResults.length===0 &&
searchInput.value" class="searchError">
            <mat-list-item  role="listitem">City not found &nbsp; &nbsp;&nbsp;
                <mat-icon matSuffix>sentiment_very_dissatisfied</mat-icon>
            </mat-list-item>
        </mat-list>
        <!-- Challenge .searchItem
              - complete for loop
              - send proper parameters to function on click event
        -->
        <mat-list role="list" *ngIf="searchResults.length">
            <mat-list-item
                role="listitem"
                *ngFor="let searchResult of searchResults"
                (click)="selectedLocation(searchResult)"
                class="searchItem">

                <mat-divider></mat-divider>
            </mat-list-item>
        </mat-list>
    </mat-card>
</div>
```

weather ts

```
import { ICityWeather } from './../models/IWeatherData.interface';
import { Injectable } from '@angular/core';
import { HttpClient } from '@angular/common/http';
import { Observable,of } from 'rxjs';
import { map } from 'rxjs/operators';

import { IWeatherRawData } from '../models/IWeatherRawData.interface';
import { ISearchResult, IWeatherData } from '../models/IWeatherData.interface';

@Injectable({
  providedIn: 'root'
})
export class WeatherService {

  constructor(
    private http: HttpClient,
  ) { }

  baseUrl = 'https://www.metaweather.com';
  iWeatherData:IWeatherData;
  iWeatherRawData:IWeatherRawData;
  searchLocation(term): Observable<ISearchResult[]> {
    /*
      CHALLANGE
       - get list of cities based on the searched string
       sample url: baseUrl/api/location/search/?query=paris
    */

this.http.get<ISearchResult[]>('https://www.metaweather.com/api/location/search/
?query=san').subscribe((data)=>{
     let response=data;
   });
   if(term=="san"){
   return of([
      {'title': 'San Francisco', 'location_type': 'City', 'woeid': 2487956,
'latt_long': '37.777119, -122.41964'},
      {'title': 'San Diego', 'location_type': 'City', 'woeid': 2487889,
'latt_long': '32.715691,-117.161720'},
      {'title': 'San Jose', 'location_type': 'City', 'woeid': 2488042,
'latt_long': '37.338581,-121.885567'},
    ]);}
    else if(term=="paris"){
      return of([
          {'title': 'Paris', 'location_type': 'City', 'woeid': 615702,
'latt_long': '48.856930,2.341200'}
        ]);
    }
  }

  getCityDetails(woeid:number): Observable<IWeatherData> {
    /*
      woeid is the city id(number).
      you can use below sample url to fetch the city weather details
      sample url : baseUrl/api/location/111111
    */

    /*
      CHALLENGE
       - fetch the city weather data
       - transform the received data to required "IWeatherData" format using
transformRawData() func
    */
   //let response;

this.http.get<IWeatherRawData>('https://www.metaweather.com/api/location/2347563
').subscribe(data=>{
    let response=this.transformRawData(data);
   });
   this.iWeatherRawData={
     consolidated_weather: [
          {
              weather_state_name: 'state',
              weather_state_abbr: 'cloudy',
              applicable_date: '2018-08-03',
              the_temp: 29,
          }
      ],
      parent: {
          title: 'country',
      },
      title: 'title',
   }
  this.iWeatherData=this.transformRawData(this.iWeatherRawData);
   return of(this.iWeatherData);
  }

  transformRawData(rawData: IWeatherRawData) {
    const transformedWeather: Array<ICityWeather> = [];

    rawData.consolidated_weather.forEach(function(obj) {
      const date = '';
      const temperature = 0;
      const weather_name = '';
      const weather_image =
`https://www.metaweather.com/static/img/weather/.svg`;

      transformedWeather.push({ } as ICityWeather);
    });

    return {
      city: 'title',
      country: '',
      weather: [],
    };
  }
}
```

app ts

```
import { Component, OnInit } from '@angular/core';
import { WeatherService } from './services/weather.service';
import { IWeatherData } from './models/IWeatherData.interface';

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent implements OnInit {
  title = 'My Weather App';
  cityDetails: IWeatherData;

  ngOnInit() {
  }

  constructor(
    private weatherService: WeatherService,
  ) {}
  getCityDetails(woeid) {
    /*
      CHALLENGE
       - pass the city id to service.getCityDetails(woeid)
    */
   this.weatherService.getCityDetails(woeid).subscribe(data=>{
     this.cityDetails=data;
   })

  }
}
```

city html

```
<!--
  CHALLENGE
    - display cityName in format "city name, country name"
    - define for loop to loop through weather of different days
    - display the weather for total 4 days (including today)
      var 'i' in class="cityData" can be used to store index of the
cityDetails.weather array
    - date should be formatted (eg: 2018-08-03 should be displayed as 'Aug 3,
2018' )
    - temperature should have
      - minimum 2 digits to the left of decimal
      - exactly 1 digit to the right of decimal
  -->
<div class="container" *ngIf="cityDetails">

  <p class="cityName">{{ cityDetails.city }}, {{ cityDetails.country }}</p>
  <div fxLayout="row" fxLayoutAlign="center none" fxLayoutGap="15px">
  <div class="container-weather" fxLayout="row">

    <mat-card class="cityData" *ngIf="i<4">
      <h3 class="date">{{ cityDetails.weather[0].date }}  </h3>
      <mat-card>
        <img [src]="" class="weatherIcon">
        <span class="weather">{{ cityDetails.weather[0].weather_name }} </span>
        <br>
        <span class="cityTemp">02.0 {{ cityDetails.weather[0].temperature }}°C
</span>
      </mat-card>
    </mat-card>

  </div>
  </div>
</div>
```

city ts

```
import { Component, OnInit, Input } from '@angular/core';
import { IWeatherData } from '../../models/IWeatherData.interface';

@Component({
  selector: 'app-city',
  templateUrl: './city.component.html',
  styleUrls: ['./city.component.css']
})
export class CityComponent implements OnInit {
  /*
    CHALLENGE
     - Take the city details from app.component.html into "cityDetails"
     - display the city details in the template
  */

  cityDetails: IWeatherData;
  i;
  constructor() { }

  ngOnInit() {
    this.i=1;
    this.cityDetails={
      city: 'chennai',
      country: 'india',
      weather: [
        {
          date: 'Aug 3, 2018',
          temperature: 31.1,
          weather_name: 'sunny',
          weather_image: 'some image url',
        }
      ]
    }
    if(this.cityDetails.weather[0].temperature==2){
      this.cityDetails.weather[0].temperature=2.0
    }
  }

}
```