CS5551 Advanced Software Engineering

Lab Assignment 2 Report

Name: Anh T Nguyen

ID: 19

Email: lndhf@mail.umkc.edu

Github link: <https://github.com/anhnguyent93/CS5551/tree/master/Lab2>

# Requirements

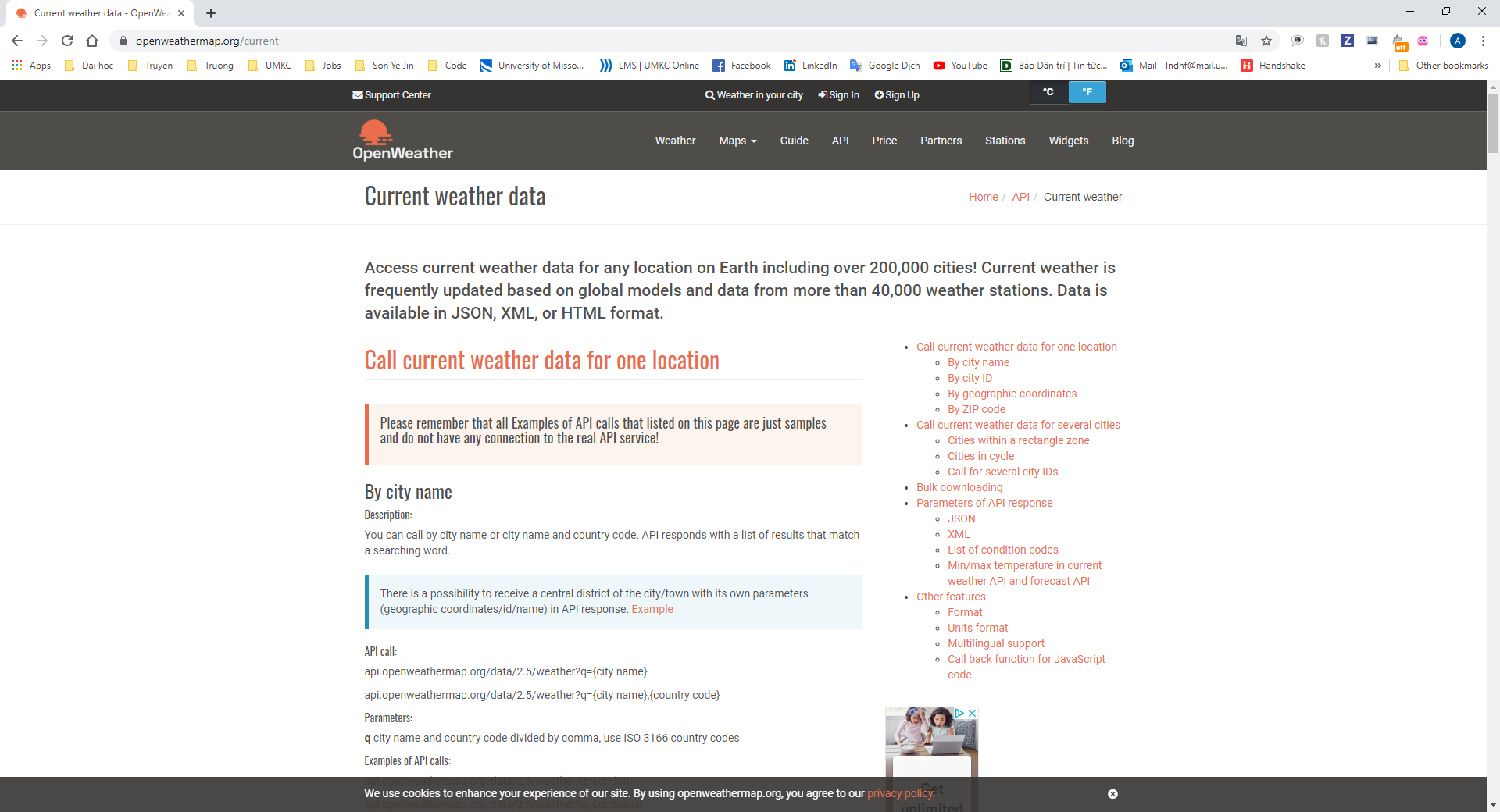
# Create a single page application:

1. Have a text field that takes city name
2. Pass the value and display weather details
3. Display the hourly forecast

# API

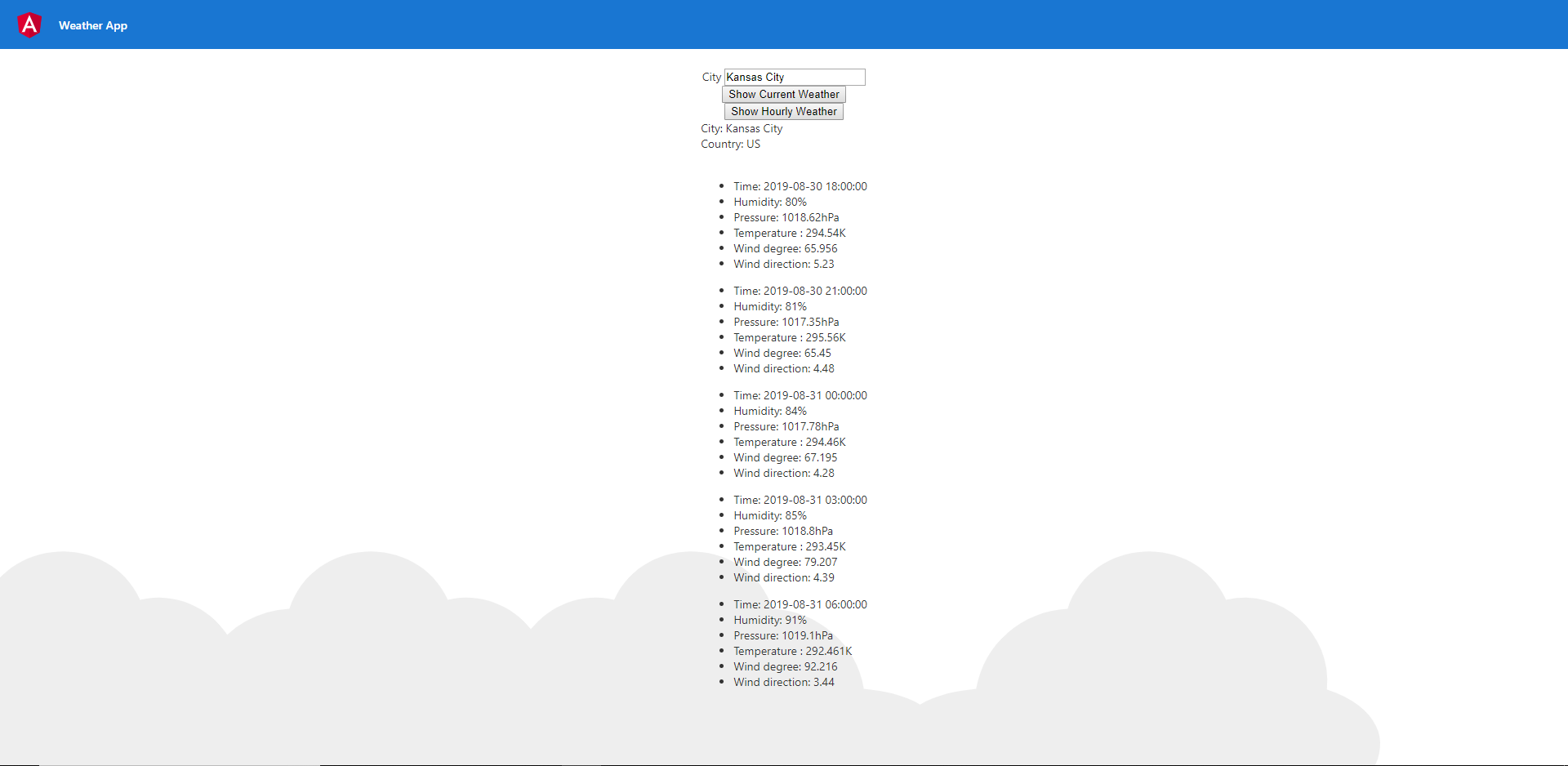
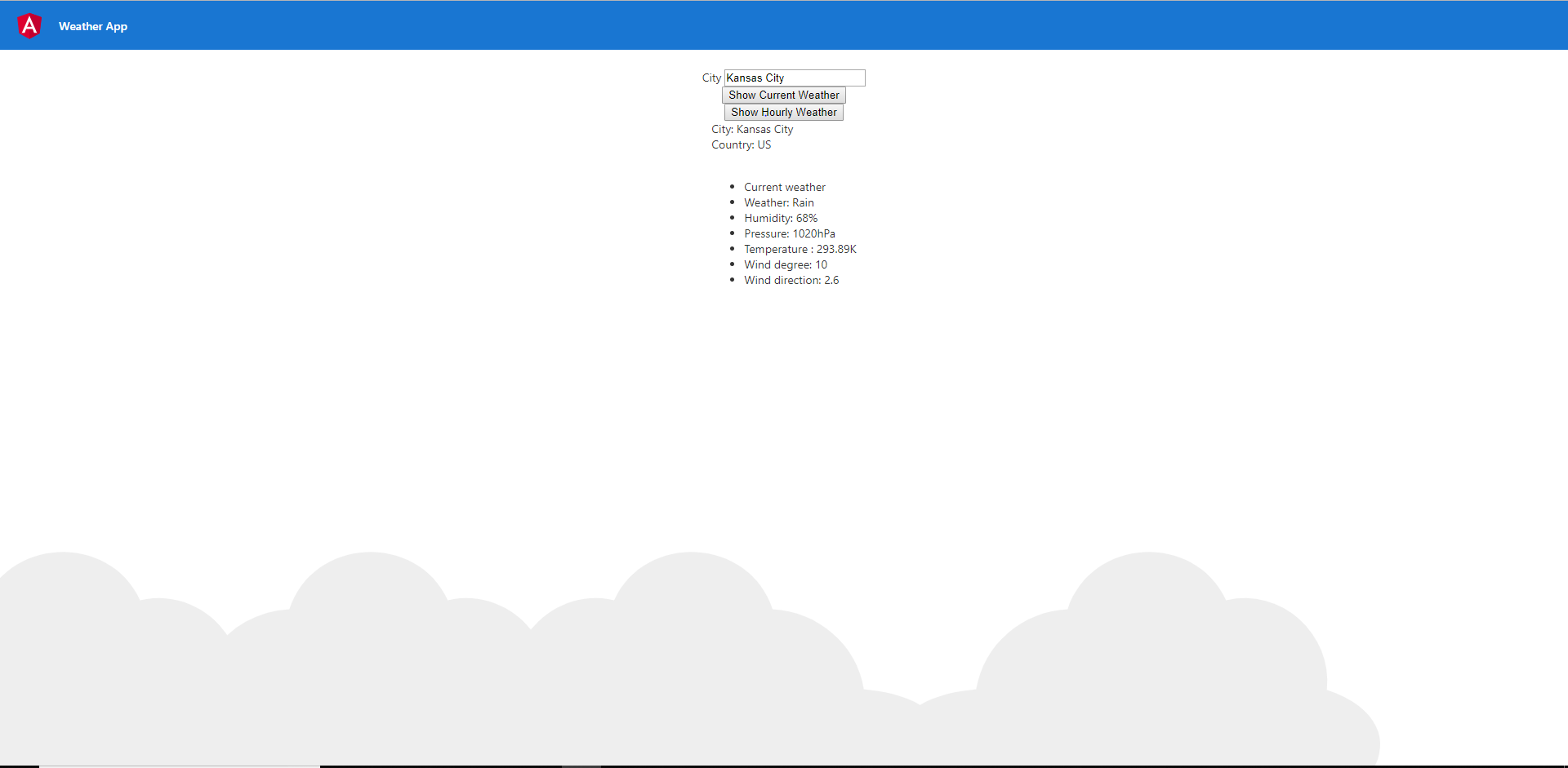
I used the API from [https://openweathermap.org](https://openweathermap.org/) to get the weather data.

I signed up and used the free plan to get current weather and 3 hour forecast details.

****

# Result

Enter the city name and click one of buttons to display the weather details.

****

# Code

## app.module.ts

import { BrowserModule } from '@angular/platform-browser';  
import { NgModule } from '@angular/core';  
import { FormsModule } from '@angular/forms'; // <-- NgModel lives here  
import { HttpClientModule } from '@angular/common/http';  
import { AppComponent } from './app.component';  
  
@NgModule({  
 declarations: [  
 AppComponent  
 ],  
 imports: [  
 BrowserModule,  
 FormsModule,  
 HttpClientModule,  
 ],  
 providers: [],  
 bootstrap: [AppComponent]  
})  
export class AppModule { }

## app.component.ts

import { Component } from '@angular/core';  
import {HttpClient} from '@angular/common/http';  
  
@Component({  
 selector: 'app-root',  
 templateUrl: './app.component.html',  
 styleUrls: ['./app.component.css']  
})  
export class AppComponent {  
 title = 'Lab 2';  
 city = 'Kansas City';  
 data: any;  
 currentUrl = 'https://api.openweathermap.org/data/2.5/weather';  
 hourlyUrl = 'https://api.openweathermap.org/data/2.5/forecast';  
 isCurrent = true;  
  
 constructor(  
 private httpClient: HttpClient  
 ) {  
 }  
  
 getCurrentWeather() {  
 this.data = null  
 this.httpClient.get(this.currentUrl, {params: {  
 q: this.city,  
 appid: '9fbf36dac24bc96ff5bf83a41bfea5b0'  
 }  
 }).subscribe(data => {  
 this.data = data;  
 this.isCurrent = true;  
 console.log(data);  
 });  
 }  
  
 getHourlyWeather() {  
 this.data = null  
 this.httpClient.get(this.hourlyUrl, {params: {  
 q: this.city,  
 appid: '9fbf36dac24bc96ff5bf83a41bfea5b0'  
 }  
 }).subscribe(data => {  
 this.data = data;  
 this.isCurrent = false;  
 console.log(data);  
 });  
 }  
}

## app.component.html

<style>  
 :host {  
 font-family: -apple-system, BlinkMacSystemFont, "Segoe UI", Roboto, Helvetica, Arial, sans-serif, "Apple Color Emoji", "Segoe UI Emoji", "Segoe UI Symbol";  
 font-size: 14px;  
 color: #333;  
 box-sizing: border-box;  
 -webkit-font-smoothing: antialiased;  
 -moz-osx-font-smoothing: grayscale;  
 }  
  
 h1,  
 h2,  
 h3,  
 h4,  
 h5,  
 h6 {  
 margin: 8px 0;  
 }  
  
 p {  
 margin: 0;  
 }  
  
 .spacer {  
 flex: 1;  
 }  
  
 .toolbar {  
 height: 60px;  
 margin: -8px;  
 display: flex;  
 align-items: center;  
 background-color: #1976d2;  
 color: white;  
 font-weight: 600;  
 }  
  
 .toolbar img {  
 margin: 0 16px;  
 }  
  
 .toolbar #twitter-logo {  
 height: 40px;  
 margin: 0 16px;  
 }  
  
 .toolbar #twitter-logo:hover {  
 opacity: 0.8;  
 }  
  
 .content {  
 display: flex;  
 margin: 32px auto;  
 padding: 0 16px;  
 max-width: 960px;  
 flex-direction: column;  
 align-items: center;  
 }  
  
 svg#clouds {  
 position: fixed;  
 bottom: -160px;  
 left: -230px;  
 z-index: -10;  
 width: 1920px;  
 }  
</style>  
  
<!-- Toolbar -->  
<div class="toolbar" role="banner">  
 <img  
 width="40"  
 alt="Angular Logo"  
 src="data:image/svg+xml;base64,PHN2ZyB4bWxucz0iaHR0cDovL3d3dy53My5vcmcvMjAwMC9zdmciIHZpZXdCb3g9IjAgMCAyNTAgMjUwIj4KICAgIDxwYXRoIGZpbGw9IiNERDAwMzEiIGQ9Ik0xMjUgMzBMMzEuOSA2My4ybDE0LjIgMTIzLjFMMTI1IDIzMGw3OC45LTQzLjcgMTQuMi0xMjMuMXoiIC8+CiAgICA8cGF0aCBmaWxsPSIjQzMwMDJGIiBkPSJNMTI1IDMwdjIyLjItLjFWMjMwbDc4LjktNDMuNyAxNC4yLTEyMy4xTDEyNSAzMHoiIC8+CiAgICA8cGF0aCAgZmlsbD0iI0ZGRkZGRiIgZD0iTTEyNSA1Mi4xTDY2LjggMTgyLjZoMjEuN2wxMS43LTI5LjJoNDkuNGwxMS43IDI5LjJIMTgzTDEyNSA1Mi4xem0xNyA4My4zaC0zNGwxNy00MC45IDE3IDQwLjl6IiAvPgogIDwvc3ZnPg=="  
 />  
 <span>Weather App</span>  
 <div class="spacer"></div>  
</div>  
  
<div class="content" role="main">  
  
 <div>  
 <label>  
 City  
 </label>  
 <input type="text" [(ngModel)]="city"/>  
 </div>  
  
 <button (click)="getCurrentWeather()">  
 Show Current Weather  
 </button>  
  
 <button (click)="getHourlyWeather()">  
 Show Hourly Weather  
 </button>  
  
 <div *\*ngIf*="!isCurrent">  
 <div *\*ngIf*="data; else notAvailable">  
 <div>City: {{data.city.name}}</div>  
 <div>Country: {{data.city.country}}</div>  
 <br/>  
 <div *\*ngFor*="let item of data.list; let i = index">  
 <ul *\*ngIf*="i<5">  
 <li>Time: {{item.dt\_txt}}</li>  
 <li>Humidity: {{item.main.humidity}}%</li>  
 <li>Pressure: {{item.main.pressure}}hPa</li>  
 <li>Temperature : {{item.main.temp}}K</li>  
 <li>Wind degree: {{item.wind.deg}}</li>  
 <li>Wind direction: {{item.wind.speed}}</li>  
 </ul>  
 </div>  
 </div>  
  
 <ng-template #notAvailable>  
 <p> Not data available <p>  
 </ng-template>  
 </div>  
  
 <div *\*ngIf*="isCurrent">  
 <div *\*ngIf*="data; else notAvailable">  
 <div>City: {{data.name}}</div>  
 <div>Country: {{data.sys.country}}</div>  
 <br/>  
 <ul>  
 <li>Current weather</li>  
 <li>Weather: {{data.weather[0].main}}</li>  
 <li>Humidity: {{data.main.humidity}}%</li>  
 <li>Pressure: {{data.main.pressure}}hPa</li>  
 <li>Temperature : {{data.main.temp}}K</li>  
 <li>Wind degree: {{data.wind.deg}}</li>  
 <li>Wind direction: {{data.wind.speed}}</li>  
 </ul>  
 </div>  
  
 <ng-template #notAvailable>  
 <p> Not data available <p>  
 </ng-template>  
 </div>  
  
 <svg id="clouds" alt="Gray Clouds Background" xmlns="http://www.w3.org/2000/svg" width="2611.084" height="485.677" viewBox="0 0 2611.084 485.677">  
 <path id="Path\_39" data-name="Path 39" d="M2379.709,863.793c10-93-77-171-168-149-52-114-225-105-264,15-75,3-140,59-152,133-30,2.83-66.725,9.829-93.5,26.25-26.771-16.421-63.5-23.42-93.5-26.25-12-74-77-130-152-133-39-120-212-129-264-15-54.084-13.075-106.753,9.173-138.488,48.9-31.734-39.726-84.4-61.974-138.487-48.9-52-114-225-105-264,15a162.027,162.027,0,0,0-103.147,43.044c-30.633-45.365-87.1-72.091-145.206-58.044-52-114-225-105-264,15-75,3-140,59-152,133-53,5-127,23-130,83-2,42,35,72,70,86,49,20,106,18,157,5a165.625,165.625,0,0,0,120,0c47,94,178,113,251,33,61.112,8.015,113.854-5.72,150.492-29.764a165.62,165.62,0,0,0,110.861-3.236c47,94,178,113,251,33,31.385,4.116,60.563,2.495,86.487-3.311,25.924,5.806,55.1,7.427,86.488,3.311,73,80,204,61,251-33a165.625,165.625,0,0,0,120,0c51,13,108,15,157-5a147.188,147.188,0,0,0,33.5-18.694,147.217,147.217,0,0,0,33.5,18.694c49,20,106,18,157,5a165.625,165.625,0,0,0,120,0c47,94,178,113,251,33C2446.709,1093.793,2554.709,922.793,2379.709,863.793Z" transform="translate(142.69 -634.312)" fill="#eee"/>  
 </svg>  
  
</div>