**Weather component.html**

<div class="center">

<div class="col-md-6 cloudset">

  <br><h2 class="text-center">Welcome to weather Forecast</h2>

  <br><p class="text-center">Please enter the city name or the Zip code to know the weather details of the city</p>

  <form [formGroup]="addForm">

    <div class="form-group">

      <label for="CountryName">Enter the city Name</label>

      <input formControlName="CountryName" placeholder="Kansas City" [(ngModel)] ="CountryName" name="CountryName" class="form-control" id="CountryName">

    </div>

    <div class="form-group">

      <label for="CountryCode">Enter the zip code</label>

      <input formControlName="CountryCode" placeholder="MO" [(ngModel)] ="CountryCode" name="CountryCode" class="form-control" id="CountryCode">

    </div>

    <div class="row addpadding">

        <div class="col">

            <button id="btnSubmit" class="btn btn-primary form-control btn-block" type="submit" (click)="onSubmit()">Submit</button></div>

        <div class="col">

          <button class="btn btn-danger form-control  btn-block" (click)="onReset()">Clear</button></div>

      </div>

  </form>

</div>

</div>

**Weather Component.ts**

import { Component, OnInit } from '@angular/core';

import { ApiService } from '../api.service';

import {FormBuilder, FormGroup, Validators, EmailValidator} from "@angular/forms";

import { Router } from '@angular/router';

@Component({

  selector: 'app-weather',

  templateUrl: './weather.component.html',

  styleUrls: ['./weather.component.scss']

})

export class WeatherComponent implements OnInit {

    constructor(private formBuilder: FormBuilder,private router: Router,private apiService: ApiService) { }

    addForm: FormGroup;

    details:any;

    submitted = false;

    ngOnInit() {

        this.addForm = this.formBuilder.group({

          CountryName: ['Kansas City', Validators.required],

          CountryCode: ['MO', Validators.required]

        });

      }

      get form()  {

          return this.addForm.controls;

        }

      onSubmit() {

        this.submitted = true;

        const data= JSON.stringify(this.addForm.value);

        this.router.navigate(['weatherDetails', {data:data}]);

      }

      onReset() {

        this.submitted = false;

        this.addForm.reset();

    }

}

**Weather component.css**

.cloudset{

    background-image: url("https://hips.hearstapps.com/hmg-prod.s3.amazonaws.com/images/cold-quotes-1575930075.jpg?crop=1.00xw:0.750xh;0,0.250xh&resize=980:\*");

}

.addpadding{

    padding-top:1px;

}

**Details component.html**

 <div class="row" >

  <div class="col-md-12">

      <div class="weather">

          <div class="current">

              <div class="info col-md-6">

                  <div class="city"><small><small>CITY:</small></small> {{this.details.name}}</div>

                  <div class="temp">{{this.details.main.temp}}&deg; <small>F</small></div>

                  <div class="wind"><small><small>WIND SPEED:</small></small> {{this.details.wind.speed}} km/h</div>

                  <div class="wind"><small><small>PRESSURE:</small></small> {{this.details.main.pressure}} pascals</div>

                  <div class="wind"><small><small>HUMIDITY:</small></small> {{this.details.main.humidity}} g/cc</div>

                  <div class="wind"><small><small>FEELS\_LIKE:</small></small>{{this.details.main.feels\_like}}&deg; <small>F</small></div>

              </div>

              <div class="col-md-6 chartmain" id="chartContainer" style="height: 504px;"></div>

          </div>

      </div>

      <app-carousel [chatCases]="this.chatCases"></app-carousel>

  </div>

</div>

Details component.ts

import { Component, OnInit } from '@angular/core';

import { ActivatedRoute, Router } from '@angular/router';

import { map, filter } from 'rxjs/operators';

import { ApiService } from '../api.service';

import \* as CanvasJS from '../../../../my-app/canvasjs.min'

@Component({

  selector: 'app-details',

  templateUrl: './details.component.html',

  styleUrls: ['./details.component.scss']

})

export class DetailsComponent implements OnInit {

  details: any;

  error: any;

  values:any;

  countryName: any;

  hourelydeatils: any;

  dateDisplay: number;

  momthDisplay: number;

  dps: {

    x: Date;

    y: any;

}[]

  chartdone: boolean;

  errorShown: boolean =false;

  constructor(private \_Activatedroute:ActivatedRoute,private apiService: ApiService,private router: Router) {

  }

  ngOnInit() {

    this.values = this.\_Activatedroute.params.subscribe(params => {

      this.values =JSON.parse(params.data);

     this.getDetails();

    this.hourlyForecast();

      });

  }

  getDetails(){

    this.apiService.weatherDetailsByName(this.values)

    .subscribe( data => {

      this.details=data;

    },(error) =>{

      this.apiService.handleError(error);

    })

    }

  getChartDetails(){

    let chart = new CanvasJS.Chart("chartContainer", {

      animationEnabled: true,

      exportEnabled: true,

      title: {

        text: "Hourly change of temprature "

      },

      axisX:{

        title: "timeline",

        valueFormatString: "DD-MMM : hh TT",

        labelAngle: -50,

        gridThickness: 2

    },

    axisY: {

        title: "Temperature"

    },

      data: [{

        type: "column",

        dataPoints: this.dps

      }]

    });

    chart.render();

    this.chartdone=true;

    }

    chatCases=[];

    hourlyForecast(){

      this.apiService.hourlyForecast(this.values)

      .subscribe( data => {

        this.hourelydeatils=data.list;

        console.log("  this.hourelydeatils",  this.hourelydeatils);

        let cnt=0;

        for (let appViewState of this.hourelydeatils) {

          if (cnt<16) {

            this.chatCases.push(appViewState);

            cnt++;

         //   console.log("this.chatCases",this.chatCases);

           }

       }

       this.dateCalculation();

      },(error) =>{

        this.apiService.handleError(error);

      })

      }

dateCalculation(){

  for (let appViewState of this.chatCases) {

    let d = new Date(appViewState.dt\_txt);

    var months = ["January","February","March","April","May","June","July","August","September","October","November","December"];

    appViewState.date = d.getDate();

    appViewState.month = d.getMonth();

    appViewState.monthName = months[d.getMonth()];

    appViewState.year = d.getFullYear();

    appViewState.hours = d.getHours();

    appViewState.minutes=d.getMinutes();

    appViewState.seconds=d.getSeconds();

    appViewState.temp=appViewState.main.temp;

    console.log("this.chatCases",this.chatCases);

    this.dps = [

      { x: new Date(this.chatCases[0].year, this.chatCases[0].month, this.chatCases[0].date,this.chatCases[0].hours,this.chatCases[0].minutes), y: this.chatCases[0].temp},

      { x: new Date(this.chatCases[1].year, this.chatCases[1].month, this.chatCases[1].date,this.chatCases[1].hours,this.chatCases[1].minutes), y: this.chatCases[1].temp},

      { x: new Date(this.chatCases[2].year, this.chatCases[2].month, this.chatCases[2].date,this.chatCases[2].hours,this.chatCases[2].minutes), y: this.chatCases[2].temp},

      { x: new Date(this.chatCases[3].year, this.chatCases[3].month, this.chatCases[3].date,this.chatCases[3].hours,this.chatCases[3].minutes), y: this.chatCases[3].temp},

      { x: new Date(this.chatCases[4].year, this.chatCases[4].month, this.chatCases[4].date,this.chatCases[4].hours,this.chatCases[4].minutes), y: this.chatCases[4].temp},

      { x: new Date(this.chatCases[5].year, this.chatCases[5].month, this.chatCases[5].date,this.chatCases[5].hours,this.chatCases[5].minutes), y: this.chatCases[5].temp},

      { x: new Date(this.chatCases[6].year, this.chatCases[6].month, this.chatCases[6].date,this.chatCases[6].hours,this.chatCases[6].minutes), y: this.chatCases[6].temp},

      { x: new Date(this.chatCases[7].year, this.chatCases[7].month, this.chatCases[7].date,this.chatCases[7].hours,this.chatCases[7].minutes), y: this.chatCases[7].temp},

      { x: new Date(this.chatCases[8].year, this.chatCases[8].month, this.chatCases[8].date,this.chatCases[8].hours,this.chatCases[8].minutes), y: this.chatCases[8].temp},

      { x: new Date(this.chatCases[9].year, this.chatCases[9].month, this.chatCases[9].date,this.chatCases[9].hours,this.chatCases[9].minutes), y: this.chatCases[9].temp},

      { x: new Date(this.chatCases[10].year, this.chatCases[10].month, this.chatCases[10].date,this.chatCases[10].hours,this.chatCases[10].minutes), y: this.chatCases[11].temp},

      { x: new Date(this.chatCases[11].year, this.chatCases[11].month, this.chatCases[11].date,this.chatCases[11].hours,this.chatCases[11].minutes), y: this.chatCases[12].temp},

      { x: new Date(this.chatCases[12].year, this.chatCases[12].month, this.chatCases[12].date,this.chatCases[12].hours,this.chatCases[12].minutes), y: this.chatCases[13].temp},

      { x: new Date(this.chatCases[13].year, this.chatCases[13].month, this.chatCases[13].date,this.chatCases[13].hours,this.chatCases[13].minutes), y: this.chatCases[14].temp},

      { x: new Date(this.chatCases[14].year, this.chatCases[14].month, this.chatCases[14].date,this.chatCases[14].hours,this.chatCases[14].minutes), y: this.chatCases[15].temp},

];

    this.getChartDetails();

  }

}

}

Details component.css

.weather

{

    display: flex;

    flex-flow: column wrap;

    box-shadow: 0px 1px 10px 0px #cfcfcf;

    overflow: hidden;

}

    .weather .current

    {

        display: flex;

        flex-flow: row wrap;

        color: white;

        padding: 20px;

    }

        .weather .current .info

        {

            background-color: #7aa0c0;

            display: flex;

            flex-flow: column wrap;

            justify-content: space-around;

            flex-grow: 2;

           // background-image: url("https://www.ksnt.com/wp-content/uploads/sites/86/2015/12/sunshine\_36366417\_ver1.0.jpg?w=960&h=540&crop=1");

        }

            .weather .current .info .city

            {

                font-size: 26px;

            }

            .weather .current .info .temp

            {

                font-size: 56px;

            }

            .weather .current .info .wind

            {

                font-size: 24px;

            }

        .weather .current .icon

        {

            text-align: center;

            font-size: 64px;

            flex-grow: 1;

        }

    .weather .future

    {

        display: flex;

        flex-flow: row nowrap;

    }

        .weather .future .day

        {

            flex-grow: 1;

            text-align: center;

            cursor: pointer;

        }

            .weather .future .day:hover

            {

                color: #fff;

                background-color: #F68D2E;

            }

            .weather .future .day h3

            {

                text-transform: uppercase;

            }

            .weather .future .day p

            {

                font-size: 28px;

            }

Carousel component.html

<div class='container-fluid'>

    <div class="row" style="padding: 20px;">

        <div class='col-md-6'>

            <h2>Hourly forecast details </h2>

            <ngb-carousel>

                <ng-template ngbSlide \*ngFor="let user of chatCases">

                    <img src="https://image.freepik.com/free-vector/day-night-weather-forecast-app-realistic-vector-icons-set-isolated-transparent-background-su\_1441-2791.jpg">

                    <div class="carousel-caption">

                        <h2 style="color: white">nature:{{user.weather[0].main}}</h2>

                        <p>sky:{{user.weather[0].description}}</p>

                        <p> pressure: {{user.main.pressure}} pascals</p>

                        <p> humidity: {{user.main.humidity}} g/cc</p>

                        <h2 style="color: white">{{user.date}} {{user.monthName}} {{user.hours}}:00 <span \*ngIf="user.hours < 12">AM</span> <span \*ngIf="user.hours > 12">PM</span></h2>

                    </div>

                </ng-template>

            </ngb-carousel>

        </div>

    </div>

</div>

Carousel component.ts

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

import { NgbCarouselConfig } from '@ng-bootstrap/ng-bootstrap';

@Component({

  selector: 'app-carousel',

  templateUrl: './carousel.component.html',

  styleUrls: ['./carousel.component.css']

})

export class CarouselComponent implements OnInit {

  @Input() chatCases:any;

  constructor(config: NgbCarouselConfig) {

    config.interval = 2000;

    config.wrap = true;

    config.keyboard = false;

    config.pauseOnHover = false;

  }

  ngOnInit() {

  }

}

Carousel component.css

.carousel-control-next-icon, .carousel-control-prev-icon {

    background-color: black;

}