Template driven approach for forms.

HTML code :

<div class="container-fluid fadeIn">

<div class="row">

<div class="col-md-4"> Reviews and Ratings

<form (ngSubmit)="onSubmit(f)" #f="ngForm"> <!--#f is a reference type(u can use any other letter) which is being passed as argument to ngSubmit() method. ngForm here is a angular directives that will let angular know to give us a direct access to the form it automatically creates.-->

<label for="name" style="float:left;font-weight:100;">Name</label>

<input

type="text"

id="name"

name="name"

placeholder="Full Name"

class="form-control"

ngModel>

<label for="comment" style="float:left;font-weight:100;">Comment</label>

<textarea

id="comment"

name="comment"

placeholder="Leave a comment.."

class="form-control"

ngModel

required></textarea><!--required 6ai to make sure that a person types in something-->

<div class="radio" \*ngFor="let rating of ratings">

<label>

<input type="radio"

name="rating"

ngModel

[value]="rating"

required>**{{** rating **}}**

</label>

</div>

<input type="button"

class="btn btn-primary"

type="submit"

[disabled]="!f.valid"> <!--diable the button if the form is not valid-->

</form>

</div>

<hr />

<div class="row" \*ngIf="submitted"><!--this section will display our data-->

<div class="col-md-4">

<h3>Your Data</h3>

<p>Name: **{{** user.name **}}**</p>

<p>Comment: **{{** user.comment **}}** </p>

<p>Rating: **{{** user.rating **}}**</p>

</div>

</div>

<hr />

<div class="col-md-4">Contact Us <p>

<div style="color : mediumvioletred;">

T : 0400000000

<br />

E : restaurant@restaurantName.com.au

</div>

</div>

</div>

</div>

TypeScript code :

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

import { NgForm } from '@angular/forms';

@Component({

selector: 'app-review',

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

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

})

export class ReviewComponent {

@ViewChild('f') reviewForm: NgForm;

//@ViewChild does the same thing as our reference type f would do. this is another approach as to how we can work with forms.while using viewChild rather than passing f as an argument in our html code,we pass it in this file only.

ratings = ['Awesome', 'Very good', 'Good', 'Pretty bad', 'Complete disaster'];

user = {

name : '',

comment : '',

rating : ''

};

submitted = false;

constructor() { }

//onSubmit(form: NgForm)//NgForm is the way to get access to our reference type #f

//{

// console.log(form);

//}

onSubmit() //@ViewChild approach.this method is useful when u have to access the form before u submit the values.

{

this.submitted = true;

this.user.name = this.reviewForm.value.name;

this.user.comment = this.reviewForm.value.comment;

this.user.rating = this.reviewForm.value.rating;

this.reviewForm.reset();/\*to reset all the values \*/

}

}