**Lesson21 Angular dynamic forms validation**

**Notes:-**

**1-The Role of the Array is that:-**

**A-If all the all the controls is valid then the whole form array is valid**

**B-If any of the controls is invalid then the whole form array is invalid**

**Solution:-**

**1-Is disabled the ADD Button when the items of the array is invalid and the previous form group will be disabled with all its form controls**

**2-Apply Delete Button to delete the current Form Group**

<div class="form-group">

<div class="col-md-offset-2 col-md-4">

<button type="button" class="btn btn-primary" (click)="addSkillButtonClick()"

//The ADD Button is disabled when the form group is invalid

[disabled]="employeeForm.get('skills').invalid">Add Skill

</button></div></div>

<div class="well">

<div formArrayName="skills" \*ngFor="let skill of employeeForm.get('skills').controls; let i = index">

<hr \*ngIf="i > 0" />

<div [formGroupName]="i">

<div class="form-group">

<label class="col-md-2 control-label" [attr.for]="'txtskillName'+i">skill Name</label>

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

<input [id]="'txtskillName'+i" type="text" class="form-control" formControlName="skillName" (blur)="logValidationErrors()">

<span class="help-block" \*ngIf="skill.get('skillName').errors?.required &&

skill.get('skillName').touched">

{{formErrors.skillName}}</span></div>

//we create Delete Button to apply the delete form group

<div class="col-sm-6" \*ngIf="employeeForm.get('skills').length>1">

<button type="button" class="btn btn-danger btn-sm pull-right" title="Delete Skill" (click)="removeSkillButtonClick(i)">

<span class="glyphicon glyphicon-remove"></span>

</button></div></div>

<div class="form-group">

<label class="col-md-2 control-label" [attr.for]="'txtexperienceInYears'+i">experience In Years</label>

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

<input [id]="'txtexperienceInYears'+i" type="text" class="form-control" formControlName="experienceInYears"

(blur)="logValidationErrors()">

<span class="help-block" \*ngIf="skill.get('experienceInYears').errors?.required &&

skill.get('experienceInYears').touched">

{{formErrors.experienceInYears}}

</span></div></div>

<div class="form-group">

<label class="col-md-2 control-label">Proficiency</label>

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

<label class="radio-inline">

<input type="radio" value="email" formControlName="proficiency">Beginner

</label>

<label class="radio-inline">

<input type="radio" value="phone" formControlName="proficiency">Intermediate

</label>

<label class="radio-inline">

<input type="radio" value="phone" formControlName="proficiency">Advance

</label>

<label class="radio-inline">

<input type="radio" value="phone" formControlName="proficiency">Advance3

</label>

<span class="help-block" \*ngIf="skill.get('proficiency').errors?.required &&

skill.get('proficiency').touched">

{{formErrors.proficiency}}

</span>

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

**//we can make validation as the others on the condition of that there are one form group enabled only**

formErrors = {

'fullName': '',

'emailGroup':'',

'email': '',

'confirmEmail':'',

'phone': '',

'skillName': '',

'experienceInYears': '',

'proficiency': ''

};

// Include required error message for phone form control

validationMessages = {

'fullName': {

'required': 'Full Name is required.',

'minlength': 'Full Name must be greater than 2 characters',

'maxlength': 'Full Name must be less than 10 characters.'},

'emailGroup':{

'matchEmailsMsg': 'email and confirm email must match',},

'email': {

'required': 'Email is required.',

'emailDomain': 'Email domian should be hotmail.com',

'emailDomain1': 'Email domian should be prgaim123.com',},

'confirmEmail': {

'required': 'Confirm Email is required.',},

'phone': {

'required': 'Phone is required.',},

'skillName': {

'required': 'Skill Name is required.',},

'experienceInYears': {

'required': 'Experience is required.',},

'proficiency': {

'required': 'Proficiency is required.',}};

//from the group index it will remove the form group at specific index

removeSkillButtonClick(skillGroupIndex: number): void {

(<FormArray>this.employeeForm.get('skills')).removeAt(skillGroupIndex);}

//when the add button clicked it’s first disable the current form group

addSkillButtonClick(): void {

//we get the Form Array with the name skills

this.FrmArry = this.employeeForm.get('skills') as FormArray;

//we get the index of the form array

let i : number = this.FrmArry.controls.length-1;

//it will disable the specific form group

this.disableDetail(this.FrmArry.controls[i] as FormGroup);

//it will insert the current form group

(<FormArray>this.employeeForm.get('skills')).push(this.AddSkillFormGrouping())}

//this method will disable the form control

disableDetail(frm:FormGroup) {

frm.get('skillName').disable();

frm.get('skillName').markAsUntouched();

frm.get('experienceInYears').disable();

frm.get('experienceInYears').markAsUntouched();

frm.get('proficiency').disable();

frm.get('proficiency').markAsUntouched();}

//it will remove the form group at specific index

removeSkillButtonClick(skillGroupIndex: number): void {

(<FormArray>this.employeeForm.get('skills')).removeAt(skillGroupIndex);}

//it will add new form group in the form array

AddSkillFormGrouping():FormGroup{

return this.fb.group({

skillName: ['', Validators.required],

experienceInYears: ['', Validators.required],

proficiency: ['', Validators.required]

});}