Angular

1.Changing DateOfBirth into Blue theme.

2.To make week number off then need to showWeekNumbers:false”­.

example:

this.datePickerConfig = Object.assign({}, { containerClass: "theme-dark-blue" ,showWeekNumbers:false});

3.To maintained min or max date .

Example:

constructor() {

this.datePickerConfig = Object.assign({}, {

containerClass: "theme-dark-blue" ,

showWeekNumbers:false,

minDate:new Date(2018, 0, 1),

maxDate:new Date(2018,11,31)});

}

4.to make date format need to update in ts

constructor() {

this.datePickerConfig = Object.assign({}, {

containerClass: "theme-dark-blue" ,

showWeekNumbers:false,

minDate:new Date(2018, 0, 1),

maxDate:new Date(2018,11,31),

dateInputFormat:'DD/MM/YYYY'

});

}

5.To get by default date need to update in create.ts file.

dateOfBirth: Date= new Date(2018,11,30);

6.to get DateOfBirth normal size not so big as like above all then need to update in create.html file.

Example:

<div class="row">

<div class="form-group col-md-5">

<label for="dateOfBirth">Date of Birth</label>

<input type="text" [(ngModel)]="dateOfBirth" [bsConfig]="datePickerConfig" name="dateOfBirth" class="form-control" bsDatepicker>

</div>

</div>

7.to make DateOfBirth place either left,right,Middel have to change.

<div class="row">

<div class="form-group col-md-5">

<label for="dateOfBirth">Date of Birth</label>

<input type="text" [(ngModel)]="dateOfBirth" [bsConfig]="datePickerConfig"

name="dateOfBirth" class="form-control" bsDatepicker placement="right">

</div>

</div>

Note:

The DatePickerConfig object can be used to configure

Min-date

Max-date

Show/hide Week Numbers

Date Format

Etc...

Lecture:13

**Angular ngif directive**

**Export class create-employee.component{**

previewPhoto= false;

togglePhotoPreview(){

this.previewPhoto= !this.previewPhoto;

}

**}**

<div class="form-group">

<label for="photoPath">Photo Path</label>

<input type="text" class="form-control" id="photoPath" [(ngModel)]="photoPath" name="photoPath">

</div>

<div class="form-group">

<button class="btn btm-primary" type="button" (click)="togglePhotoPreview()">

{{previewPhoto?"Hide":"Show"}} Preview</button>

</div>

<div class="form-group">

<img [src]="photoPath" height="200" width="200" \*ngIf="previewPhoto" />

</div>

1. **Every time clicking show preview button then submit alys executing to fix this issue need to write type=”button” in create.html.**

**Example->**

</div>

<div class="form-group">

<button class="btn btm-primary" type="button" (click)="togglePhotoPreview()" >Show Preview</button>

</div>

2.now some enhacement if photo has been showed then need one button which will behave as a Hide Preview or Show Vise-Versa...

Example->

<div class="form-group">

<button class="btn btm-primary" type="button" (click)="togglePhotoPreview()">

{{previewPhoto?"Hide":"Show"}} Preview</button>

</div>

Lecture 14.

Angular disable browser validation

1. By default Angular 4and later version disable brower native Validation.

How to Enable Browser validation.

2.to make enable

<form #employeeForm="ngForm" ngNativeValidate (ngSubmit)="saveEmployee(employeeForm)">

<div class="panel panel-primary">

<div class="panel-heading">

<h3 class="panel-title">Create Employee</h3>

</div>

<div class="panel-body">

<div class="form-group">

<label for="fullName">Full Name</label>

<input id="fullName" required [(ngModel)]="fullName" name="Fullname" type="text" class="form-control" placeholder="Enter your Fullname"

/>

</div>

**Lecture #14.** **Angular form validation**

**Angular** form validation properties->

**1.Touched**

**2.Un-Touched**

**3.Pristine**

**4.Dirty**

**5.Valid**

**6.Invalid**

**HTML5 validation Attributes**

**1.required**

**2.maxLength**

**3.pattern**

**4.min**

**5.max**

**To check validation properties we will do some changes**

<div class="panel-body">

<div class="form-group">

<label for="fullName">Full Name</label>

<input id="fullName" required [(ngModel)]="fullName" name="Fullname" type="text" class="form-control" placeholder="Enter your Fullname"

#fullNameControl="ngModel" />

</div>

<div>

touched ->{{fullNameControl.touched}}<br>

un-touched ->{{fullNameControl.untouched}}

</div>

Note->

1.To include the html5 validation attribute such as requied for example:

<input id="fullName" type=”text”/>

2.Export ngModel directive to a local templte variable

<input id=”fullName” required #fullNameControler=”ngModel” type=”text”/>

3.use the local template variable to access the validation properties (touched,untouched,dirty,valid,etc).

<div>touch{{ fullNameControler .touched}}</div>

<div>untouch{{ fullNameControler .untouched}}</div>

<div>dirty{{ fullNameControler .dirty}}</div>

4.to check entire for validation properties

Export ngForm directive to a local template reference variable.

<form #employeeForm="ngForm" ngNativeValidate (ngSubmit)="saveEmployee(employeeForm)">

5.use the template reference variable to access the validation properties at form lavel (touched,valid,dirty).

<div>touch{{ employeeForm.touched}}</div>

<div>untouch{{ employeeForm.untouched}}</div>

<div>dirty{{ employeeForm.dirty}}</div>

**Lecture #15.** **Displaying angular form validation error messages**

**BootStarp classes for styling validation error messages.**

**->has-error**

**->control-label**

**->help-block**

<div class="form-group" [class.has-error]="fullNameControl.invalid && fullNameControl.touched" [class.has-success]="fullNameControl.valid">

<label for="fullName" class="control-label">Full Name</label>

<input id="fullName" required [(ngModel)]="fullName" name="Fullname" type="text" class="form-control" placeholder="Enter your Fullname"

#fullNameControl="ngModel" />

<span class="help-block" \*ngIf="fullNameControl.invalid && fullNameControl.touched" >FullName Required</span>

</div>

**Note->**

**To make from submit button disable.**

<button type="submit" class="btn btn-primary" [disabled]="employeeForm.invalid">Save</button>