Steps to intall angular

[Angular - Introduction to modules](https://angular.io/guide/architecture-modules)

* check if node.js installed
* cmd: node –v
* install anagular cli
* cmd: npm install -g @angular/cli
* to check verion
* cmd:ng –version
* to create new angular application
* cmd:ng new myFirstApp
* to create component :
* ng generate component first
* ng g c first
* //here first is component name
* ng g c first –is -skipTests
* //to ignore scss file and creating inline css
* ng g c first –it -skipTests
* //to creating inline template
* ng g pipe pipes/custom
* npm install bootstrap
* //to install services
* ng generate service hero => will create HeroService
* //to install routes
* ng generate module app-routing --flat --module=app

angular input output decorators : [https://angular.io/guide/inputs-outputs](https://angular.io/guide/inputs-outputshttps:/angular.io/guide/inputs-outputs)

angular forms: [Angular 8 - Template-Driven Forms Validation Example | Jason Watmore's Blog](https://jasonwatmore.com/post/2019/06/15/angular-8-template-driven-forms-validation-example)

angular io: [Angular - Building a template-driven form](https://angular.io/guide/forms)

reactive forms: [Angular 10 - Dynamic Reactive Forms Example | Jason Watmore's Blog](https://jasonwatmore.com/post/2020/09/18/angular-10-dynamic-reactive-forms-example)

<form [formGroup]="registrationForm" (ngSubmit)="onSubmit()">

<div class="form-group">

<label>UserName</label>

<input type="text" formControlName="name" class="form-control" >

<div class="error" class="text-danger" \*ngIf="registrationForm.controls['name'].hasError('required') && registrationForm.controls['name'].touched">name is required</div>

</div>

<div class="form-group">

<label>password</label>

<input type="password" formControlName="password" class="form-control" >

<div class="error" class="text-danger" \*ngIf="registrationForm.controls['password'].hasError('required') && registrationForm.controls['password'].touched">password is required</div>

</div>

<div class="form-group">

<label>confirm password</label>

<input type="password" formControlName="confirmPassword" class="form-control" >

<div class="error" class="text-danger" \*ngIf="registrationForm.controls['confirmPassword'].hasError('required') && registrationForm.controls['confirmPassword'].touched">Confirm is required</div>

registrationForm=new FormGroup(

{name: new FormControl('',[Validators.required, Validators.pattern('^[a-zA-Z ]\*$')]),

password: new FormControl('',Validators.required),

confirmPassword:new FormControl('' ,Validators.required)});