**Angular Notes**

-Angular is a front-end development framework. which is used to create single-page applications which won't refresh when we travel from page to page

* To install angular at the first time

npm install -g @angular/cli

* Create the first project go to the required folder and then

Ng new “project name”

* To run a newly created application which is on port 4200

Ng serve.

* Package.json -it contains details of your project and which script runs your project and all libraries included in this file only
* When we first create an application everything is inside the src/app

The app is a component. Component is a piece of code that we can reuse.

* When we have to save images we can save them into assets

**-Interpolation**

* When you want to display dynamic data in html page
* We use this interpolation symbol {{}}
* We can use the function also in interpolation to get data

<h1>{{getValue()}}</h1>

**-Property Binding([value])**

when we do value from ts to html Boolean value is not getting properly using interpolation so always use property binding **-**

**-Event binding ( (click) )**

we can have many events like click,keyup, keydown , blear

**-Two-way Data Binding**

-same time update and display a property this is a two-way binding

-use have to use[(ngModel)]=”property\_Name” and import formModule

**-IMP Command**

* We can create module component classes and interfaces with this commands
* Ng g m modulename;
* Ng g c module\_name/component\_name

-when you want to build you use ng build it will build and create a new folder dist that you have to deploy

-through selector, we can reuse component

**--I**nline style we can achieve by providing a flag while creating a component so that when we have less modification in that file we can go with this approach

Ng g c component\_name –inline-style

Ng g c component\_name –inline-template

-here you have to provide everything in that component.ts file not in css and templeate file .

--VERY IMP

-suppose if I create new Module and I have to use the component inside this module to app module so what I have to do

1)first register this module in appModule first imports this module

2)int the new module you have to export the module so that it will get reflected in app module.

**--Function Call**

Create function in .ts file like

getName(){

alert( “my name is shubham”);

}

-in html we can create a button and use(click)=”function()” to call that function by button click event;

--how to get value from text box to .ts file and display to html

We will create one function in .ts file and assign the value getting from input text box

<input type=”text” (keyup)=”getName()” placeholder=” Enter Name” name=”name” />

--the way of style we use either declare with class and give color or what ever property want to give .

-else we can use global style for heading in global css

-we can use internal style and inline style as well.

**-Property Binding**

-suppose I want to take value from .ts file I can use interpolation as well as property binding to set value but property binding will work with Boolean as well where interpolation wont work with Boolean value.

Suppose I am having value in ts file

Name:string=”shubham”;

-I want this value in my text box at html

<input type =”text” name=”user-name” value={{name}} />

OR

<input type =”text” name=”user-name” [value]=name />

-suppose I am having Boolean at ts it wont read properly it read as string not Boolean value in case of interpolation If I want to change something on button it fail

disable:bookean=false;

<input type =”text” name=”user-name” value={{name}} disabled={{disable}}/>

<input type =”text” name=”user-name” [value]=name [disabled]=disable />

**-NG template we will use with if else condition**

**-Ng switch**

<div [ngSwitch]="color">

    <h1 \*ngSwitchCase="'red'" style="color: red;">Red Color</h1>

    <h1 \*ngSwitchCase="'green'" style="color:green">green Color</h1>

    <h1 \*ngSwitchCase="'blue'" style="color:blue">blue Color</h1>

    <h1 \*ngSwitchCase="'pink'"style="color:pink">pink Color</h1>

    <h1 \*ngSwitchDefault=>Color Not matching</h1>

</div>

**-NgFor loop and nested loop in angular**

**--How to create nav bar in angular**

**Angular form**

**Template based form and reactive form**

**-first we have to import form module inside your module**

**-use form tag and in that tag use # formName and ngForm with (ngSubmit)=”fun()”**

**-and every filed use ngModel**

**Toggle Element In Angular**

-use if and in .ts file we use this.isSubmit = ! this.isSubmit so it will toggle.

**How to install Bootstrap in your project**

ng add @ng-bootstrap/ng-bootstrap

**How to Install Marterial UI**   
ng add @angular/material

-Two way Binding

-same time update and display property called as two way Binding

-import form Module in that module.ts file

-in text field use [(ngModel)]=”property name” use same name in .ts file.

**-Component communication**

**-There are many ways they can communicate mainly**

**1)@Input() Decorator**

**2)@output() Decorator**

**3)services**

**1)Input/Output**

**-for this we require relationship parent child relation**

**Im having user module in that module I have two componants**

**List-user and filter-user**

**List User is parent so it will pass data to filter-user.**

**And at filter-user .ts file we will use @Input() decorater**

**Here we pass data via[anyname]=”same name in parent.ts”**

**Parent -> child data transfer through compoanant**

**Child to Parent data Transfer using @Output and new EventEmitter .emit method**

**-in case of child to parent communication we need to use @Output at child side only and transfer data through click event by creating method in child and when you click at child through eventEmitter .emit method we can pass the data**

**-in parent where we are calling child we have to capture with same name like**

**(updatedList)=anyMethodname($event);**

**While passing from child to parent we use**

**(same name in child )=methodname($event);**

**-Angular Component Via Services**

**-create service inside service folder and in app.module.ts import httpClientModule**

import {HttpClientModule} from '@angular/common/http';

**and import HttpClientModule**

**-after this go to your service where u need to import HttpClient and inject in constructor**

**Import {HttpClient} from ‘@angular/common/http’;**

**Constructor(private http:HttpClient){}**

**Directive**

* **It will provide additional feature to element**

**Routing(VVIMP)**