**Day 2 :**

**16 June – 2024**

We want to pass the value (dynamic value from ) view to components

1. Template reference.
2. Angular forms

Using this we can pass group of value using reference name.

* 1. Template drive form
     1. View to component flow
     2. We use ngForm and ngModel attribute in view side.
     3. Easy to use.
     4. Good for html background
     5. For simple form.

In template driven form we can create form reference using command as

<form #loginRef=”ngForm”>

</form>

ngForm and ngModel attribute part of FormsModule. So while using these attribute we need to import in app.module.ts file.

* 1. Model driven form or reactive forms.
     1. Component to view
     2. formGroup formControlName attribute in view side and FormGroup and FormControl in ts side
     3. complex to use
     4. good for ts or angular background.
     5. Complex form.

ng g c angular-forms child of app component

ng g c template-reference these three are child of angular-forms

ng g c tdf-login

ng g c mdf-login

Angular forms

Angular service

Angular Service :

If we write any business logic (simple or complex ) that logic is local to that components.

But if we need same logic for more than one component we can write that code in separate class that class is known as service class.

Html (template) component.ts service.ts

Tdf-login.html ---🡪 checkUser()

checkUser()

Mdf-login.html -🡪 checkUser()

Checking emailid and password

Angular service mainly divided into two types.

1. User defined service
   1. Create user defined service class explicitly using new keyword. (normal class)
   2. Create user defined service class object using DI (Dependency Injection)

as well as IOC(Inversion of control).

IOC : Inversion of control : it is a concept or design pattern.

In place creating or maintaining any resources explicitly allow to create by container or engine. If container create it will maintain properly. Base upon our requirement we need to pull it and use it and leave it.

DI : Dependency Injection : Implementation of IOC.

1. Setter base DI
2. Constructor base DI

1. Pre-defined service

HttpClient : it is pre defined api provided by angular which help to call rest api develop using angular language like axios() or fetch().

Axios or fetch() return type is Promise

We need to use then and catch. If resolve then call else catch block.

HttpClient return type is Observable.

Observable is part of rxjs (reactive programming).

First inside component or user defined service class we need to do DI for HttpClient. HttpClient is part HttpClientModule. So we need to import this module in app.module.ts

Then we can do DI for HttpClient inside constructor

this.http.get(“url”).subscribe({

next 🡪 load the data one by one

error 🡪 this is use to handle error

complete🡪 after loaded successfully third parameter get called.

})

fake component ng g c fake it is use to create the component

fake service ng g s fake it is use to create the service

fake model ng g class fake this command create normal class

which is use to map json data.

app.component.html

app.module.ts

fake.component.html

fake.component.ts

fake.service.ts