which basically imports some script that are required for running angular this is required because angular Framework uses features of JavaScript that are not available in the current version of JavaScript supported by most browsers available out there so this policy will the gap between the features of JavaScript that angular needs and the features supported by the current browser

**editconfig :** we can specify a set of rules in this file which every developer needs to follow while developing the application.

**gitInnore :** you can specify the files and folders here which you want to exclude from your git repository.

**Angular.json** : this is one of the most important files and it contains all the configuration of a project so it contains configuration like what is the project name what is the folder name what is the source folder name and other configurations.

**Package.json** : this is a standard configuration file which every node project has in this file you can see we have some configurations like name of the project, version of the project and other configuration then you will also see we have this dependencies. It also contains dependencies and devdependencies.

**Tsconfig.json :** this file contains a bunch of settings for your typescript compiler the types and compiler looks at this settings and based on these settings it is going to compile your types in code into JavaScript that the browser can understand

App.module.ts

bootstrap: [AppComponent] : here we are telling angulaer that it should know about this app component when application starts.

where the script files are added in index.html :-

the script files are injected by angular cli automatically in the index HTML webpage ok so whenever we run this NG serve command is NG serve command first reveals the angular project then it will create JavaScript bundles and it will add them automatically to this index.htm file so in the final file the script files will be present like you can see here so this script files injected to this index.htm by by angular cli

work of ng serve command :-

NG serve command will first recompile the angular project it will generate some bundles it will inject those one bundles in the index.html file and then it will restart the live development server so you can see the angular project has been compiled successfully these bundles have been generated and they have been injected to the index of HTML file and then it also restarted the live development server ok so this is how an angular project gets executed

================================================================================

selector: 'app-root' : used as HTML tag

selector: '[app-root]' : now we can use it as HTML attribute.

selector: '.app-root' : now we can use it as css class.

================================================================================

What is data binding : way to communicate between a component and its corresponding view template.

What is directives?

directives are simply an instruction to the dom. in the term we use directives to tell them what to add to the web page and what not we also instruct them using directives on how to display the HTML element.

Components are one kind of directive. components a kind of one such instruction in the Dom using components we instruct them what to add in the web page.

Example of some custom directive :-

changeDivGreen : to change the background color.

How to make directive?

@Directive({

Seclector: [‘changeDirective’]

})

Export class changeDirective{

}

Structureal directive : structural directive manipulates the Dom by adding or removing web page elements. ngFor,ngIf

Attribute directive : attribute directive does not add or remove elements from the web page it simply changes they look at the behaviour of the HTML element. NgStyle , ngClass.

NgFor directive is used to repet a portionof HTML template once per each item from an itterable list.

Custom Property binding :

We can pass data from parent component to child component using @Input decorator. It called as custom property binding because here we bind the custom properties of child component class with the property or method of parent component class.

Template reference variable : it is a variable which refer a dom element or a component or a directive.

We can also say that it is a local reference variable which is referring a dom element.

Example :-

<input type=”text” #ggg (keyup)=”0”> //its mean that ggg is referring the whole input element. Now with the help of ggg we can access all the properties like value , id of input element.

<p> {{ggg.value}} </p>

@ViewChild(‘ ‘) : it assign a property with a reference to an html element or to a component or a directive.

Example :-

@ViewChild(‘dobInput’) dateOfbirth : ElementRef

NativeElement : it is a property of ElementRef which actually stores the Html element.

ng-content :-

1. The ng-content is used when we want to insert the content dynamically inside the child component from the parent component that helps to increase component reusability.
2. Using ng-content we can pass content inside the component selector and when angular parses that content that appears at the place of ng-content.

Angular Service :

Angular service is a typescript class without any decorator in Angular and we can use that type script class across multiple components in our angular application.

Why we need service class ?

1. Easy to debug and test.
2. Services provides reusability of code.
3. With services we can communicate across different component weather these component has parent child relationship or not.

What is Dependency Injection?

Dependency injection is a technique in which a class receives its dependencies from external sources rather than creating them itself

Provider array :-

we learn that when we want angular to inject an instance of a service to our component class or directive we use providers array to tell angular what type of instance it needs to provide us.

Hierarchical Injection : If we provide a service on one component, the angular fremwork will create and inject an instance of that service for that component and all its child component.

What is Observable?

We use observables to perform asynchronous operations and handle asynchronous data. Observable strems the data , it sends the data data in the form of packets.

Observable is provided by RxJS(Reactive Extension library for javaScript).

What is RxJS?

It is library of Js that allow us to work with async data stream.

Observable vs Promise:-

a promise returns us all the data at once it provides us a single data

but an observable streams the data. it provides us multiple values in the form of packets. When some of the data will be available it will send the data and then it will again get the rest of the data and it will send it with the response.

A promise promises you some data it will certainly give you that data even if there is no code using that data .

but in case of an observable the observable will only provide the data if there is someone to use that data if there is no one which is using that data in that case the observable will not send that data.

What is subscribe method?

The subscribe() method takes an observer object as a parameter.

import { Observable } from 'rxjs';

const observable = new Observable(observer => {

observer.next(1);

observer.next(2);

observer.complete();

});

observable.subscribe(

value => console.log(value),

error => console.log(error),

() => console.log('complete')

);

Subscribe takes 3 methods as parameters each are functions:

* next: For each item being emitted by the observable perform this function
* error: If somewhere in the stream an error is found, do this method
* complete: Once all items are complete from the stream, do this method

What is routing :-

Routing allow us to navigate from one part of out application to another part of our application or from one view to another view.

<router-outlet> </router-outlet> :-

It is a directive that tell to angular where in our page we want to display while routing.

[RouterLink] : It is a directive when applied to an element in a template, it creates a link to that element that initiates navigation on a route

[RouterLinkActive] : router link active is a directive for adding or removing classes from an HTML element that is bound to routerlink using this directive we can toggle CSS classes for active router links based on the current router state and the main use case of this directive is to highlight which route is currently active.

[RouterLinkActiveOption] : “{excat:true}” :-

This directive will make the router link active directory to be applied only if the route URL matches exactly to the current URL.

**Absolute and Relative route path :-**

**routerLink=”/about” :- (Absolute path)**

when we specify a slash before a path it uses the absolute path that means that path gets appended to the root URL and what is the route URL it is localhost 4200 so this part will be appended to this route URL so it will be like localhost 4200 slash about.

Instad of / if we use “./about” now it will consider as relative path.

**routerLink = “about” :- (Relative path)**

when we use a relative path that path gets appended to the currently active route.

**This.route.navigate([‘home’])**

**This.route.navigateUrl(‘home’)**

Above two functions helps to navigate between routes programmatically.

Here you could see that the path which we gives in above two function we have’nt used ‘/’ before them so you can though that it is relative path. But this is not a relative path that we have provided in the url , it is absolute path. Because this above two function always takes absolute path as a argument.

**To make it relative path we need to do following steps :-**

This.route.navigate([‘home’],{relativeTo : this.activeRoute});

{path: 'update-employee/:id/:name', component: UpdateEmployeeComponent},

:id/:name – this id or name will act as a placeholder , the value of the id can change dynamically in the runtime.

this.id = this.route.snapshot.params['id'];

this.id = this.route.snapshot.paramMap.get('id');

These two helps to retrieve value from url.

What is Query Parameters?

Quary parameters are optional parameters that you pass to a route. These query parameters are added at the end of the url separated by “?”.

Example :

**Localhost:4200/products?id=12345&&Name=iphone**

**Path parameter(Route Parameter) vs Quary parameters**

Path params are used to identify a specific resource, while query params are used to filter and sort the data. Path params are typically used when retrieving a single resource, while query params are used when retrieving multiple resources.

Aslo a major difference is quary parameters always takes value in “key value” pair.

**How to sent a Quary Parameter in the router URL?**

[quaryParams]=”{edit=true}” //through HTML code

This.router.navigate([‘course’,this.courseId],{quaryParams:{edit:true}}) //through programmatically

What is Fragment :-

a fragment in a route so if fragment in a route is nothing but a link which jumps to a section or a content in HTML page which contains the ID mentioned in the fragment . it is denoted by ‘#’

Angular pipes :-

Pipes in angular are used to transform data before displaying it in the view. Angular Pipes takes data as input and formats or transform the data to display it in the template.

Example of pipes :-

Uppercase , lowercase , date , async etc