**Phase 4 :**

**Front end technologies**

**HTML and HTML5**

**CSS and CSS3**

**JS using ES5 and ES6 style**

**Adv JS**

**Function and types of functions**

**ES6 features let, const and var keywords**

**Pre defined object**

**String**

**Math**

**Arrays**

**Bootstrap**

**TypeScript**

**Angular Framework**

**HTML and CSS**

**HTML : Hyper text mark up language which is use to create the web page.**

**Web page is use to display the content on browser in different formats.**

**http://**[**www.google.com**](http://www.google.com)**--> URL**

**req (http/https)---------------------🡪**

**Client Server**

**🡨------res(http/https) HTML/HTML5**

**HTML provided lot of pre-defined tags or elements. Which help to create the web page.**

**HTML is not case sensitive.**

**Syntax**

**<tagName>**

**</tagName>**

**<html>**

**<head>**

**<title>Welcome to My Web Page</title>**

**</head>**

**<body>**

**<p>Welcome to My App</p>**

**</body>**

**</html>**

**IDE : VS Code :**

**6 heading tags start from h1 to h6**

**H1 largest**

**H6 smallest**

**Attribute : attribute is use to describe the properties of tags.**

**Attribute we can write in opening tag in the form of key-value pairs**

**<tag attributeName=”value”></tagName>**

**IN HTML4**

**<!doctype html public url=”sfasfasfasfasf.dtd”/>**

**Document type definition**

**In html5 they remove this rules and they provided few new tags which help to create dynamic web page with help of html.**

**<!doctype html>**

**CSS : Cascading style sheet :**

**CSS provide lot of pre-defined attribute which help to apply different formatting style for web page**

**With the help of css we can do separation of concern.**

**Types of CSS**

1. **Inline CSS**
2. **Internal CSS or Embedded CSS**
3. **External CSS**

**Inline CSS**

**Syntax**

**<tagName style=”property:value;property:value;”></tagName>**

**<p style=”color:red;”>Hello</p>**

**<p>**

**<font color=”red”>Hello</font>**

**</p>**

**Internal CSS or Embedded CSS**

**In between head tag we have to write style tag**

**<style>**

**Selector {property:value}**

**</style>**

**Types of selector**

1. **Universal selector : \* {property:value;}**
2. **Specific tag selector : tagName {property:value}**
3. **Multi specific selector tagName,tagName {property:value}**
4. **Local class selector tagName.className {property:value }**
5. **Global class selector .className{property:value}**
6. **Id selector : #idName{property:value}**
7. **Child selector : parenttagame childtagName {property:value}**

**Class selector Vs Id selector**

**Class is known as group of tags of same type or different types.**

**<p class=”abc” id=”a1”>First Para</p>**

**<p class=”xyz” id=”a2”>Second Para</p>**

**<p class=”abc” id=”a3”>Third Para</p>**

**<p class=”xyz” id=”a4”>Fourth Para</p>**

**External CSS**

**JavaScript : JavaScript was object based interpreter scripting language. Which provide programming features on web page.**

**We will learn JS using ES5 (ECMA ) European Computer Manufacture Association**

**ECMA is a concept and JS is a one of the implementation of ECMA or ES.**

**From ES6 JS also known as object oriented scripting language.**

**<script type=”text/JavaScript”> opening tag**

**</script> closing tag**

**In between head or body tag of html page.**

**Data types :**

**4 types**

1. **Number type**
2. **String type**
3. **Boolean type**
4. **Object reference type**

**In JS to declare the variable we use var keyword (till ES5 JS).**

**Operator**

1. **Arithmetic operator**
2. **Logical operator**
3. **Conditional operator**
4. **Increment and decrement**

**= == ===**

**Typeof**

**31-07-2022**

**If statement**

**If else**

**If else if**

**Switch**

**Loop**

**While loop**

**Do while loop**

**For loop**

**functions**

**function is use to write the set of instruction to perform a specific task.**

**In JS function are divided into two types**

1. **Pre-defined function or built in function**
2. **User-defined function**

**Pre-defined function**

1. **alert(“Msg”): This is a pre defined function which help to display pop up message.**
2. **Prompt() : using this function we can take the value through keyboards.**
3. **parseInt() : it is use to convert string to integer**
4. **parseFloat() : it is use to convert string to float**
5. **eval() : it is use to convert string to number (int or float)**

**user defined function**

**In JS we can declare the function lot of ways.**

1. **Simple syntax In ES5 style**

**syntax**

**function functionName(parameterList) {**

**function body;**

**}**

1. **Function no passing parameter and no return type**
2. **Function passing parameter and no return type**
3. **Function no passing parameter but return type**
4. **Function passing parameter and return type**

**Event : event provide bridge between html and JS.**

**In JS all event start with pre-fix on followed by event name.**

**Like**

**onClick**

**onDblClick**

**onMouseOver**

**onKeyUp**

**onKeyDown**

**onSubmit**

**onFocus**

**onBlur**

**onLoad**

**onUnload**

**onChange**

**etc**

**source of the event all html tags (ie p, div, h1 tag) or forms components like button, radio button check box, window , keyboards etc.**

**listener : they are normal function which help capture the events.**

**DOM : Document Object Model : DOM API : Document Object Model Application Programing interface.**

**Lot of programming like Java, Python, JS etc provided DOM API which help to read, write and update HTML contents dynamically.**

**DOM Hierarchy**

**index.html**

**html**

**head body**

**title p Hello**

**meta h1 Title**

**script div Desc**

**BOM : Browser Object Model**

**Top most hierarchy is BOM**

**Object properties or state**

**Behavior**

**Object properties**

**Behavior**

**Object property**

**Behavior**

**Object**

**window.document.write();**

**or**

**document.write();**

**window.alert(); or alert()**

**ES5 and ES6 JS Features**

**var, let and const keyword we use to declare the variable in JS.**

**Using var we can re-declare same variable once again with same value or different value.**

**var a=10;**

**a=30;**

**var a=20;**

**int a=10;**

**a=20;**

**int a=30; // error in Java**

**using let keyword we can’t do re-declaration.**

**let c=10;**

**c=20;**

**let c=30; // Error in JS**

**using var we can declare global scope**

**using let we can declare local scope like if block or for block etc.**

**06-08-2022**

**Arrow function : arrow function also known as anonymous function.**

**It is equal to lambda expression.**

**In arrow function if we want to write only one statement curly braces not required.**

**By default arrow function return the value without return keyword.**

**Callback function : passing the function name or body or function itself to another function as a parameter is known as callback function.**

**IIFE : Immediate invoke function expression**

**(functionbody)(functioncall);**

**Array : In JavaScript array is known as pre-defined object. which provide set of methods which help do some operation like add, remove, iterate etc. In JavaScript array can store same value as well as different types of values.**

**Syntax to create the array in JavaScript**

1. **Using literal style**
2. **Using new keyword**

**Creating user-defined object.**

**object : any real world entity**

**class : blue print of object or template of object.**

**In JS we can create user-defined object 3 ways**

1. **Literal style : ES5 JS**
2. **Function style : ES5 JS**
3. **Class style : ES6 JS**

**TypeScript : typescript is known as super set of JavaScript which support all ES6 features.**

**Browser doesn’t support typescript directly. So we have to convert typescript to javacript and then we have to include that js file in html page.**

**Typescript support data types.**

**To convert TS to JS we required node js.**

**Node JS : Node JS is run time environment for JS.**

**Before Node JS JavaScript is known as Client side scripting language. But after node JS JavaScript also known as Client side as well as Server side scripting language.**

**Node JS provided lot of predefined module which help to create server side programming language using JS. Like file handling programing, connecting database, creating restful web service using JS.**

**MEAN Stack Mongo DB Express JS Angular Node JS**

**MERN Stack Mongo DB Express JS React JS Node JS**

**Transpiler : converting one format to another format**

**Typescript : it will covert ts to js**

**Babel : es6 to es5**

**Npm : node package manager : this command help us to download external node js modules.**

**To download any external module we have to use the command as**

**npm install –g moduleName**

**npm install –g typescript**

**07-08-2022**

**Modules : modules is like a package in typescript which contains collection of classes, function, interfaces as well as variable when they have same name but different purpose.**

**Using modules we can write our code in different files and using import and export we connect both the files.**

**Angular : Angular is a open source web framework. Which help to develop SPA (Single Page Application). Angular internally follow MVC architecture on view side or on frontend side. Angular is a part of google.**

**Angular JS 1.0 to 1.7 or 1.x version**

**MVC : Model View and Controller**

**To learn angular JS we need knowledge of HTML, CSS and JS using ES5.**

**Angular Framework**

**2.x to 13.x**

**In Angular framework controller is replaced by component. Angular framework is component based architecture framework.**

**To learn Angular Framework we need knowledge of HTML, CSS, JS using ES5, ES6 and TypeScript.**

**Library Vs Framework**

**React JS is a library : Library is light weighted. It doesn’t follow any standard. In MVC react is known as View.**

**Angular is Framework. Angular internally follow standard. The implementation of design pattern is taken care by framework. Framework is heavy and complex compare the library.**

**MPA SPA**

**Index.html home.html**

**Html html**

**Head head**

**Title title**

**Body body**

**Others tags other tags**

**Angular and React JS use component concept. Component control the view or part of the view on web page. Component is use to re-usability of the code. So we can create the component using class or function.**

**React JS we will create the component using function style as well as class style**

**Angular we will create the component using class style.**

**Google provided angular cli (command line interface) which help to create the project with all configuration to run that project.**

**We have to enable ng command. (next generation on web page).**

**Angular**

**To enable ng command we have to install the angular cli with the help of npm command.**

**Npm install @angular/cli –g**

**Or**

**npm install @angular/cli --location=global**

**ng version**

**to create the new project using ng command**

**ng new project-name**

**routing -🡪No**

**styling 🡪 CSS**

**cd project-name (move inside a project folder)**

**code . (short cut to open the project in VS Code)**

**to run the angular project we have to run the command as**

**ng serve**

**it compile the project after compiled 100% open the browser and write url as**

[**http://localhost:4200**](http://localhost:4200) **(by default angular run on port number 4200)**

**src**

**app**

**app.component.html --🡪 Template page**

**app.component.css -🡪 style sheet file**

**app.component.ts ---🡪 component file**

**Angular provided decorator concept.**

**Decorator : decorator is like a annotation in Java. Which provide meta-data information. It is function with extra functionality to make the class or function or variable special.**

**@Component**

**@NgModule**

**@Input**

**@Output**

**@Injectable**

**Etc**

**@Component is a type of decorator to make normal class as component. Which provide set of attribute.**

**selector : “my-tag” This attribute provide user-defined tag name.**

**templateUrl: : This attribute provide the path of html page. Using this attribute we connect to html page**

**styleUrl : : This attribute provide the path of css page. Using this attribute we connect style sheet like link tag.**

**app.module.ts**

**module : collection of variable, function, classes and interfaces.**

@NgModule

**According to angular module is combination of component, services, pipes, etc.**

**declaration : while developing we create more than one component. So all component declaration must be provide in declaration section.**

**Imports : this attribute is use to declare pre-defined or user-defined modules.**

**browserModule.**

**provider : This attribute is use to provide the angular service class details.**

**Bootstrap : This attribute provide parent component details or first component details.**

**app.component.spec.ts : This file use for testing using jasmine and karma tool.**

**Main.ts file : this file provide the parent module details.**

**We will create more than one module and each module contains more than one component.**

**LoginModule AccountModule CustomerModule MainModule**

**1st com**

**2nd com**

**3rd com**

**Angular.json file : This file contains angular configuration details from here we will get the details about main.ts file and index.html file.**

**13-08-2022**

**Data binding :**

**Ng new data-binding**

**Routing 🡪 no**

**Styling –css**

**ng serve –o**

**Or**

**ng serve –open**

**after program compiled 100% it will open this application in our default browser automatically.**

**We can create the component using the cli command as**

**Ng generate component componentname**

**Or**

**ng g c sample**

**Data binding : using data binding we can share the properties values from component to view and vice-versa.**

**Data binding is a bridge between component to view.**

**2 types**

1. **One way data binding** 
   1. **String interpolation : the flow of the application component to view**

**Component -------------------------🡪View**

**This syntax we have to write in html or template page.**

**Syntax**

**{{}}**

**{{variableName}} {{name}}**

**{{expression }} {{6+10}}**

**{{functionName()}} {{display()}}**

**ng g c string-interpolation**

* 1. **Property binding : the flow the application component to view**

**Component -------------------------🡪View**

**This syntax we have to write in html or template page**

**Syntax**

**[]**

**<input type=”text” name=”uname” value=”Raj Deep”/> Raj Deep default value in text field.**

**<input type=”text” value=”fname”/> value itself is a fname.**

**<input type=”text” [value]=”fname”/> then angular check the variable inside a component with name as fname and that fname value display inside a text field.**

**<p [innerText]=”fname”></p>**

**<div [innerText]=”fname”></div>**

**ng g c property-binding**

**we can get the value using string interpolation without dom element but for property binding dom element is required.**

**In string interpolation always value consider as string so it is use to display the value.**

**Property binding we can use other data types rather string.**

* 1. **Event binding : flow of the application view to component**

**View ---------------------------------🡪Component**

**Angular use same event provided by JavaScript only different they remove on pre-fix and all event wrap with ().**

**JavaScript event Angular event**

**onClick (click)**

**onDblClick (dblclick)**

**onMouseOver (mouseover)**

**onSubmit (ngSubmit)**

**using JS event we can all JavaScript function**

**using Angular event we can call Typescript component function without creating the object of component class.**

**ng g c event-binding**

**using event binding with string – interpolation or property binding we can achieve two way data binding.**

**Passing the value from template or view to component**

**Using template reference.**

**<input type=”text” #refName/>**

1. **Two way data binding : if any changes happen on component side it will update automatically on view side and vice-versa.**

**Two way data binding using ngModel attribute.**

**View🡨----------------------------🡪Component**

**Syntax**

**[()]**

**<input type=”text” [(ngModel)]=”uname”/>**

**ngModel is a pre-defined attribute part of FormsModule. So we have to import FormsModule in app.module.ts file in import section.**

**ng g c two-way-binding**

**Types of Directive**

**Directive is use to add the extra behavior to dom or html page.**

**3 types of directive**

1. **Component directive : it is a type of directive which help to create the user-defined tags.**

**@Component({**

**Selector : “my-tag”,**

**template-url:”./myPage.html” // in this page we can**

**}) // write static or dynamic using**

**class MyComponent { // data binding ie 1 way or 2 way.**

**}**

**<my-tag></my-tag>**

1. **Structure directive : using structure directive we can add or remove dom element or tags.**

**\*ngIf**

**\*ngFor**

**We can use if or looping inside js or ts file but we can’t use in html code.**

**But angular allow us to use if and for looping inside html code with help of structure directive**

**ng g c structure-directive**

1. **Attribute directive**

**Attribute directive is use to add dynamic css effects.**

**ngStyle**

**ngClass**

**Syntax to create the model class or interface**

**ng g class classname**

**or**

**ng g interface interfacename**

**14-08-2022**

**ng new angular-forms**

**using template reference we can pass only one value from template to component. But if we want to pass more than one value then we have to create more than one template reference.**

**If we want to pass the more than one value as a container then angular provided angular forms.**

**Angular support two types of forms**

1. **Template Driven Form**

**Flow of this type of form. From template to component. Template ---🡪component**

**This type of forms is very easy to develop and good for small application.**

**The people form html and css background they use this type of form.**

**In This form we have to use two pre-defined attribute ie ngForm and ngModel.**

**Both attribute are part of FormsModule so we have to import FormsModule in app.module.ts file in import section.**

**In this type of form more code on template side and less code in component side.**

**In Template driven form we will create the reference of form using ngForm attribute.**

**Syntax**

**<form #loginRef=”ngForm”>**

**</form>**

1. **Model Driven form or reactive forms.**

**Flow of this type of form. From component to view . Component -----🡪 Template**

**This type of form complex to learn and good for complex forms.**

**To us this type of form you must good in TS and Angular API.**

**In This type of forms we have to use FormGroup and FormControl API in component side. And we have to use formGroup and formControlName attribute in template side. Theses attribute is a part of ReativeFormsModule. So we have to import ReactiveFormsModule in app.module.ts file in import section.**

**In this type of form more code on component side and less code on template side.**

**According to Model driven form if we want to create textfield, password field radio button so we have to create the reference of formcontrol.**

**All form control are grouped in side a fromgroup api.**

**If we write any variable or property and function inside a component those property as well as function we can access within that component or that component ‘ template page.**

**So if we write any business logic simple or complex that logic become local to that component. If we want to use same logic in more than one place we can use angular service.**

**Html component service**

**Tdf-login.html ts**

**Ts (service class)**

**Mdf-login.html ts**

**Angular service mainly divided into two types**

1. **User –defined service**
   1. **Creating object using new keyword**
   2. **Creating object using DI concept.**

**Angular support constructor base DI**

**We can create the service class using angular cli**

**ng g s loginDetails**

1. **Pre-defined or built in service**

**Create Spring boot project with Spring data to Store product details, retrieve product details, delete and update product details.**

**Pid**

**Pname**

**Price**

**url :image URL**

**ng g c tdf-login-page**

**ng g c mdf-login-page**

**ng new angular-service**

**20-08-2022**

**create table product(pid int primary key,pname varchar(10), price float, url blob);**

**first create the spring boot project with four REST API**

1. **Store product**
2. **Delete product**
3. **Update product**
4. **Retrieve product**

**Now create angular project using command as**

**ng new angular-service**

**with routing no**

**styling css**

**ng g c product (create the component)**

**ng g s product (create the service)**

**ng g class product (create the model class)**

**Angular Provided pre-defined API ie HttpClient which help to call REST API develop in any language. So inside user-defined service class we can to do DI for HttpClient.**

**Whenever doing DI for HttpClient we have to import HttpClientModule in app.module.ts file in import section. Because HttpClient API is a part of HttpClientModule.**

**HttpClient provided get, post, put and delete method to call REST API. These method return type is Observable. Observable is a part of RxJS which help to load the data from backend technologies in asynchronous manner. So if we want to load the data we have to use subscribe. Which will take 3 callback parameter**

**1st parameter is use to load the data one by one.**

**2nd parameter will get call if any error generated.**

**3rd parameter get call after all data loaded successfully if no error generate.**

**CORS : Cross Origin Resource Sharing :**

**21-08-2022**

**this.http.get(“url”).subscribe(next,error,()=>{});**

**subscribe method takes 3 callback function as a parameter**

**first parameter is next : which get all automatically to load the data one by one.**

**Second parameter : error it will call if any error generate while before loading, while loading or after loading.**

**Third parameter :completed : it will call after 1st parameter successfully loaded the data.**

**Angular Routing :**

**Using Angular routing we can navigate from one component to another component with or without conditions.**

**Angular mainly use to create the SPA (Single Page Application).**

**Ng new angular-routing**

**Routing 🡪 Yes**

**Styling 🡪 css**