Table of Contents

1.7

1

Angular 5 Examples

How to practice these examples?

1) create the project by using the following command

```
new new example1
```

2) step into the project by using the change directory command

```
cd example1
```

3) start the server

```
ng serve
```

4) open the browser

```
http://localhost:4200
```

5) start developing the code by opening the folder in the vscode editor.

6) start the vs code editor

7)

8) open the example1/src/app folder

9) delete all the preconfigured files in the app folder

10) start type the example code.

11) verify in the browser.

Basic Example

1) app.module.ts

```
import { NgModule          } from '@angular/core';

import { BrowserModule } from "@angular/platform-browser";

import { HelloComponent } from './app.component';

@NgModule({

imports: [BrowserModule], declarations: [HelloComponent], bootstrap:[HelloComponent] })
export class AppModule { }
```

2) app.component.ts

```
import { Component} from '@angular/core';

@Component({

selector: 'app-root', template: 'Basic Example' })

export class HelloComponent{

}
```

Components

Components are the basic build block of the angular application.

In this one we learn about the following concepts

1) property binding

2) event binding.

Interpolation

1) app.module.ts

```
import { NgModule } from '@angular/core'; import { BrowserModule } from
'@angular/platform-browser';

import { AppComponent } from './app.component';

@NgModule({

imports: [BrowserModule], declarations: [AppComponent], bootstrap: [AppComponent] })
export class AppModule { }
```

2) app.component.ts

```
import { Component } from '@angular/core';

@Component({

selector: 'app-root', template: `

<h1>{{message}}</h1> <input type="text" value="{{message}}"/> ` }) export class
AppComponent {

    message: string = 'Angular – Interpolation Example'; }
```

Data Binding

Data binding can be of two types.

1) attribute binding

2) property binding.

Attribute Binding: -

1) app.module.ts

```
import { NgModule } from '@angular/core'; import { BrowserModule } from
'@angular/platform-browser';

import { AppComponent } from './app.component';

@NgModule({

imports: [BrowserModule], declarations: [AppComponent], bootstrap: [AppComponent] })
export class AppModule { }
```

2) app.component.ts

```
import { Component } from '@angular/core';

@Component({

selector: 'app-root', template: `<div><input type="text" [value]="message"/></div>` }) export
class AppComponent {

    message: string = 'Angular – attribute Binding Syntax'; }
```

Property Binding: -

1) app.module.ts

```
import { NgModule } from '@angular/core'; import { BrowserModule } from
'@angular/platform-browser';

import { AppComponent } from './app.component';

@NgModule({

imports: [BrowserModule], declarations: [AppComponent], bootstrap: [AppComponent] })
export class AppModule { }
```

2) app.component.ts

```
import { Component } from '@angular/core';

@Component({

selector: 'app-root', template: `

<div>

   <h1 [textContent]="message"></h1> </div> ` }) export class AppComponent {

   message: string = 'Angular – Property Binding Syntax'; }
```

Event Binding

Ex1:- Listening to the click events

1) app.module.ts

```
import { NgModule } from '@angular/core'; import { BrowserModule } from '@angular/platform-browser';

import { AppComponent } from './app.component';

@NgModule({

imports: [BrowserModule], declarations: [AppComponent], bootstrap: [AppComponent] }) export class AppModule { }
```

2) app.component.ts

```
import { Component } from '@angular/core';

@Component({

selector: 'app-root', template: `<div><h1>{{message}}</h1><button (click)="showMessage()"/></div>` }) export class AppComponent {

message: string = '';

showMessage() {

   this.message = 'Angular – Event Binding'; } }
```

Ex2:- listening to the keyboard events

2 . 3) Event Binding

1) app.module.ts

```
import { NgModule } from '@angular/core'; import { BrowserModule } from '@angular/platform-browser';

import { AppComponent } from './app.component';

@NgModule({

imports: [BrowserModule], declarations: [AppComponent], bootstrap: [AppComponent] })
export class AppModule { }
```

2) app.component.ts

```
import { Component } from '@angular/core';

@Component({

selector: 'event-binding-app', template: `

<div>

<h1>{{message}}</h1> <input type="text" (keypress)="showMessage($event)"/> </div> ` })
export class AppComponent {

message: string = 'Angular – Event Binding';

showMessage(onKeyPressEvent: KeyboardEvent) {

        this.message = (<HTMLInputElement>onKeyPressEvent.target).va lue;

    } }
```

2 . 3) Event Binding

12

Forms

Forms are used to capture the data from the user.

In this one we learn about the following concepts

1) template driven forms

2) reactive forms

3) validations

Template Driven forms

1) app.module.ts

```
import { StudentRegComponent } from './student-reg.component'; import { StudentService } from './student.service'; import { NgModule } from '@angular/core';

import { AppComponent } from './app.component'; import { BrowserModule } from "@angular/platform-browser"; import { FormsModule } from "@angular/forms"; @NgModule({

imports: [BrowserModule,FormsModule], declarations: [AppComponent,StudentRegComponent], bootstrap:[AppComponent] }) export class AppModule { }
```

2) app.component.ts

```
import { Component } from '@angular/core'; @Component({

selector: 'app', templateUrl: 'app.component.html' }) export class AppComponent{ }
```

3) app.component.html

```
<h1>Student Reg</h1> <student-reg></student-reg>
```

4) student-reg.component.html

3 . 1) Template Driven forms

```
<form>
            Id <input type="text" name="id" [(ngModel)]="student.id"/> Name <input type="text"
                                                  name="name" [(ngModel)]="student.nam e"/><br/>
      <button (click)="register();" >Register</button> </form>
```

5)student-reg.component.ts

```
import { StudentService } from './student.service'; import { Student } from './student'; import {
Component, OnInit } from '@angular/core';
@Component({
selector: 'student-reg', templateUrl: 'student-reg.component.html' })
export class StudentRegComponent implements OnInit {
student:Student; constructor() { } ngOnInit() {
     this.student = new Student(); } register(){
      console.log(this.student); } }
```

6)student.ts

```
export class Student{
                                                                    id:number; name:string; }
```

Validations

1) app.module.ts

```
import { StudentRegComponent } from './student-reg.component'; import { StudentService }
from './student.service'; import { NgModule } from '@angular/core';

import { AppComponent } from './app.component'; import { BrowserModule } from
"@angular/platform-browser"; import { FormsModule } from "@angular/forms"; @NgModule({

imports: [BrowserModule,FormsModule], declarations:
[AppComponent,StudentRegComponent], providers: [], bootstrap:[AppComponent] }) export
class AppModule { }
```

2) app.component.ts

```
import { Component, OnInit } from '@angular/core';

@Component({

selector: 'app', templateUrl: 'app.component.html' })

export class AppComponent implements OnInit {

constructor() { }

        ngOnInit() { } }
```

3) app.component.html

```
<h1>Student Reg</h1> <student-reg></student-reg>
```

4) student-reg.component.html

```
<form #studentForm="ngForm">
    Id <input type="text" minlength="3" min="999" name="id" #idC trl="ngModel"
[(ngModel)]="student.id"/>
<span *ngIf="idCtrl.errors&&idCtrl.errors.minlength">
    atleast 3 charcters should enter </span> <span *ngIf="idCtrl.errors&&idCtrl.errors.min">
   maximum value is 999 </span> <br/>
    Name <input type="text" name="name" #nameCtrl="ngModel" [(ng
Model)]="student.name"/><br/>
    <button (click)="register();" [disabled]="studentForm.invali d">Register</button>
</form>
```

5)student.ts

```
export class Student{
                                                            id:number; name:string; }
```

6)student-reg.component.ts

3 . 2) Validations

```
import { StudentService } from './student.service'; import { Student } from './student'; import { Component, OnInit } from '@angular/core';

@Component({

selector: 'student-reg', templateUrl: 'student-reg.component.html' })

export class StudentRegComponent implements OnInit {

student:Student; constructor() { } ngOnInit() {

    this.student = new Student(); } register(){

      console.log(this.student); } }
```

Reactive Forms

1) app.module.ts

```
import { StudentRegComponent } from './student-reg.component'; import { StudentService }
from './student.service'; import { NgModule } from '@angular/core'; import { AppComponent }
from './app.component'; import { BrowserModule } from "@angular/platform-browser"; import
{ ReactiveFormsModule } from "@angular/forms"; @NgModule({

imports: [BrowserModule, ReactiveFormsModule], declarations:
[AppComponent,StudentRegComponent], providers: [], bootstrap:[AppComponent] }) export
class AppModule { }
```

2) app.component.ts

```
import { Component, OnInit } from '@angular/core';

@Component({

selector: 'app', templateUrl: 'app.component.html' })

export class AppComponent { }
```

3) app.component.html

```
<h1>Student Reg</h1> <student-reg></student-reg>
```

4) student-reg.component.html

3 . 3) Reactive Forms

```
<form [formGroup]="studentForm">
        Id <input type="text" name="id" formControlName="id"/> <br/> Name <input type="text"
                                                        name="name" formControlName="name"/> <br/>
        <button (click)="register();" [disabled]="studentForm.invali d">Register</button>
</form>
```

5)student-reg.component.ts

```
import { StudentService } from './student.service'; import { Student } from './student'; import {
Component, OnInit } from '@angular/core';
@Component({
selector: 'student-reg', templateUrl: 'student-reg.component.html' })
export class StudentRegComponent implements OnInit {
studentForm:FormGroup; constructor() { } ngOnInit() {
this.studentForm = new FormGroup({ id:new FormControl(",[]), name:new FormControl(",[])
}); } register(){
        console.log(this.student); } }
```

6)student.ts

3 . 3) Reactive Forms

```
export class Student{

id:number; name:string;
```

3 . 4) Validations

Validations

1) app.module.ts

```
import { StudentRegComponent } from './student-reg.component'; import { StudentService }
from './student.service'; import { NgModule } from '@angular/core';

import { AppComponent } from './app.component'; import { BrowserModule } from
"@angular/platform-browser"; import { ReactiveFormsModule } from "@angular/forms";
@NgModule({

imports: [BrowserModule, ReactiveFormsModule], declarations:
[AppComponent,StudentRegComponent], providers: [], bootstrap:[AppComponent] }) export
class AppModule { }
```

2) app.component.ts

```
import { Component, OnInit } from '@angular/core';

@Component({

selector: 'app', templateUrl: 'app.component.html' })

export class AppComponent implements OnInit {

constructor() { }

    ngOnInit() { } }
```

3) app.component.html

```
<h1>Student Reg</h1> <student-reg></student-reg>
```

4) student-reg.component.html

```
<form [formGroup]="studentForm">
Id <input type="text" name="id" formControlName="id"/> <br/>
       <span *ngIf="studentForm.id.invalid&&tudentForm.id.errors.mi nlength">
       at least 4 charcters should enter </span>
       Name <input type="text" name="name" formControlName="name"/> <br/>
       <span *ngIf="studentForm.name.invalid&&tudentForm.name.error s.required">
       name is required. </span> <br/>
       <button (click)="register();" [disabled]="studentForm.invali d">Register</button>
</form>
```

5)student.ts

```
export class Student{
                                                                              id:number; name:string; }
```

6)student-reg.component.ts

3 . 4) Validations

```
import { StudentService } from './student.service'; import { Student } from './student'; import {
Component, OnInit } from '@angular/core';

@Component({

selector: 'student-reg', templateUrl: 'student-reg.component.html' })

export class StudentRegComponent implements OnInit {

studentForm:FormGroup;

constructor() { }

ngOnInit() {

this.studentForm = new FormGroup({

id:new FormControl(",[Validators.minLength(4)]), name:new
FormControl(",[Validators.required]) }); }

register(){

    console.log(this.student); } }
```

NgFor

NgFor is a structural directive, meaning that it changes the structure of the DOM.

1) app.module.ts

```
import { NgModule } from '@angular/core'; import { BrowserModule } from
"@angular/platform-browser"; import { AppComponent } from './app.component'; import {
StudentListComponent } from "./student-list.component"; @NgModule({

imports: [BrowserModule], declarations: [AppComponent,StudentListComponent], bootstrap:
[AppComponent], }) export class AppModule { }
```

2) app.component.ts

```
import { Component } from '@angular/core'; @Component({

moduleId: module.id, selector: 'project-root', templateUrl: 'app.component.html' }) export class
AppComponent {

message:string; constructor() {

    this.message = "Student List"; } }
```

3) app.component.html

```
<p>{{message}}</p> <student-list></student-list>
```

4) student-list.component.ts

```
import { Component, OnInit } from '@angular/core'; import { Student } from "./student";
@Component({
selector: 'student-list', templateUrl: 'student-list.component.html' }) export class
StudentListComponent{
students:Student[]; constructor() {
            this.students = [new Student(1,"student1"),new Student(2 ,"student2")];
        } }
```

5) student-list.component.html

```
<table>
<tr><td>ID</td><td>Name</td></tr> <tr *ngFor="let student of students">
<td>{{student.id}}</td> <td>{{student.name}}</td> </tr> </table>
```

6) student.ts

```
export class Student{
id:number; name:string; constructor(a,b){
this.id=a; this.name=b; } }
```

4 . 1 .1) ngFor

ngIf

1)app.module.ts

```
import { BrowserModule } from '@angular/platform-browser'; import { NgModule } from
'@angular/core'; import { FormsModule } from '@angular/forms'; import { HttpModule } from
'@angular/http'; import { AppComponent } from './app.component'; @NgModule({

declarations: [

   AppComponent ], imports: [

BrowserModule, FormsModule, HttpModule ], providers: [], bootstrap: [AppComponent] })
export class AppModule { }
```

2)app.component.ts

```
import { Component } from '@angular/core'; import { FormsModule } from '@angular/forms';
@Component({

selector: 'app-root', templateUrl: './app.component.html' }) export class AppComponent {

title: string = 'ngIf Example' ; showMe: boolean; }
```

3)app.component.html

4 . 1 . 2) ngIf

```
<p> {{title}} </p> <div class='panel panel-primary'>
<div class='panel-heading'>
        <p>Simple example of ngIf </p> </div> <div class="panel-body">
<div class="row">
          Show <input type="checkbox" [(ngModel)] ="showMe"/> </div> <div class='row'>
<div *ngIf="showMe">
            ShowMe is checked </div> <div *ngIf="!showMe">
            ShowMe is unchecked </div> </div> </div> </div>
```

Ng If else

1)app.module.ts

```
import { BrowserModule } from '@angular/platform-browser'; import { NgModule } from
'@angular/core'; import { AppComponent } from './app.component'; @NgModule({

imports: [BrowserModule], declarations: [AppComponent], bootstrap: [AppComponent], })
export class AppModule { }
```

2)app.component.ts

```
import { Component, OnInit } from '@angular/core'; @Component({

selector: 'app', templateUrl: 'app.component.html' }) export class AppComponent{

loginPage:boolean = false; public showLogin(){

        this.loginPage = true; } public hideLogin(){

        this.loginPage = false; } login(){

        this.loginPage = false; } }
```

3)app.component.html

4 . 1 . 3) ngIfElse

```
<div *ngIf="loginPage;else elseBlock">

<form>

Email <input type="text" name="email"/>

Password <input type="password" name="password"/> <button
(click)="login();">login</button> </form> <button (click)="hideLogin();">close</button>
</div> <ng-template #elseBlock>Welcome<button (click)="showLogin();">lo
gin</button></ng-template>
```

ngStyle: -

1)app.module.ts

```
import { BrowserModule } from '@angular/platform-browser'; import { NgModule } from
'@angular/core'; import { FormsModule } from '@angular/forms'; import { HttpModule } from
'@angular/http'; import { AppComponent } from './app.component'; @NgModule({

declarations: [

    AppComponent ], imports: [

BrowserModule, FormsModule, HttpModule ], providers: [], bootstrap: [AppComponent] })
export class AppModule { }
```

2)app.component.ts

```
import { Component } from '@angular/core'; import { FormsModule } from '@angular/forms';
@Component({ selector: 'app', templateUrl: './app.component.html', }) export class
AppComponent{

title: string = 'ngStyle Example' ; size: number = 12; color: string= 'red'; styleClass: StyleClass =
new StyleClass(); }
```

4 . 1 . 4) ngStyle

```
class StyleClass {
color: string= 'blue'; 'font-size.%': number= 150; 'font-weight': string= 'bold'; }
```

3)app.component.html

```
<p> {{title}} </p> <div class='panel panel-primary'>
<div class='panel-heading'>
    <p>Simple example of ngIf </p> </div> <div class="panel-body">
<input [(ngModel)]="color" /> <div [ngStyle]="{'color': color}">
    Change my color </div>
 <input [(ngModel)]="size" /> <div [ngStyle]="{'font-size.px': size}">
        Change my size </div>
<div [ngStyle]="styleClass">
    Change my size & Color </div>
    </div> </div>
```

4 . 1 . 5) ngClass

1)app.module.ts

```
import { BrowserModule } from '@angular/platform-browser'; import { NgModule } from
'@angular/core'; import { FormsModule } from '@angular/forms'; import { HttpModule } from
'@angular/http'; import { AppComponent } from './app.component'; @NgModule({
declarations: [
        AppComponent ], imports: [
BrowserModule, FormsModule, HttpModule ], bootstrap: [AppComponent] }) export class
AppModule { }
```

2)app.component.ts

```
import { Component } from '@angular/core'; import { FormsModule } from '@angular/forms';
@Component({ selector: 'app', templateUrl: './app.component.html', }) export class
AppComponent{
     title: string = 'ngClass Example' ; cssStringVar: string= 'red size20'; cssClass: CssClass = new
                                                                                          CssClass(); }
```

4 . 1 . 5) ngClass

```
class CssClass {
                                                   red: boolean= true; size20: boolean= true; }
```

3)app.component.html

```
<p> {{title}} </p> <div class='panel panel-primary'>
<div class='panel-heading'>
    <p>Simple example of ngClass </p> </div> <div class="panel-body">
<div class="row">
<div [ngClass]="'red size20'">
    Red Text with Size 20px : as string </div> </div>
 <!-- This also works.Â•Â•Â• --> <div class="row">
<div ngClass='red size20'>
    Red Text with Size 20px : as string </div> </div> <div class="row">
<div [ngClass]="['red','size20']">
    Red Text with Size 20px : as array </div> </div>
<div class="row">
<div [ngClass]="{'red':true,'size20':true}"> Red Text with Size 20px : as object </div> </div>
       <!-- Getting Data from Component. You can modify the CSS From the
componentÂ•Â•Â• -->
<div class="row">
<div [ngClass]="cssStringVar">
Red Text with Size 20px : from component
```

```
    </div> </div> <div class="row">
<div [ngClass]="cssClass">
                Red Text with Size 20px : from component as obj ect
     </div> </div>
    </div> </div>
```

4 . 1 . 6) ngSwitch

NgSwitch

1)app.component.html

```
<p> {{title}} </p> <div class='panel panel-primary'>
<div class='panel-heading'>
        <p>Simple example of ngSwitch </p> </div> <div class="panel-body">
Input string : <input type='text' [(ngModel)] ="num"/>
<div [ngSwitch]="num">
<div *ngSwitchCase="'1'">One</div> <div *ngSwitchCase="'2'">Two</div> <div
*ngSwitchCase="'3'">Three</div> <div *ngSwitchCase="'4'">Four</div> <div
*ngSwitchCase="'5'">Five</div> <div *ngSwitchDefault>This is Default</div> </div> </div>
</div> <div class='panel panel-primary'>
<div class='panel-heading'>
        <p>Simple example of ngSwitch </p> </div> <div class="panel-body">
<div class='row'>
<div class='col-md-6'>
<select [(ngModel)]="selectedValue">
                                        <option *ngFor="let item of items;" [value]=
"item.name">{{item.name}}</option>
        </select> </div> </div>
<div class='col-md-6'>
<div class='row' [ngSwitch]="selectedValue">
<div *ngSwitchCase="'One'">One is Pressed</div> <div *ngSwitchCase="'Two'">Two is
Selected</div>
```

```
        <div *ngSwitchDefault>This is Default</div> </div> </div> </div> </div>
```

2)app.component.ts

```
import { Component } from '@angular/core'; import { FormsModule } from '@angular/forms';
@Component({
selector: 'app', templateUrl: './app.component.html', }) export class AppComponent {
title: string = 'ngSwitch Example' ; num: number= 0; items: item[] = [{name: 'One', val: 1},
{name: 'Two', val: 2 }, {name: 'Three', val: 3}];
        selectedValue: string= 'One'; } class item {
                                                                  name: string; val: number; }
```

3)app.module.ts

4 . 1 . 6) ngSwitch

```
import { BrowserModule } from '@angular/platform-browser'; import { NgModule } from
'@angular/core'; import { FormsModule } from '@angular/forms'; import { HttpModule } from
'@angular/http'; import { AppComponent } from './app.component'; @NgModule({
declarations: [
    AppComponent ], imports: [
BrowserModule, FormsModule, HttpModule ], providers: [], bootstrap: [AppComponent] })
export class AppModule { }
```

4 . 2 .1) Basic Directives

Basic Example

1)app.component.ts

```
import { Component} from '@angular/core'; @Component({

moduleId: module.id, selector: 'project-app', template: '<ul><li *ngFor="let user of users" highlight>{{us er}}</li></ul>' }) export class AppComponent {

users; constructor() {

    this.users = ["s1","s2","s3"]; } }
```

2)app.module.ts

```
import { NgModule } from '@angular/core'; import { BrowserModule } from "@angular/platform-browser"; import { AppComponent } from './app.component'; import { HighLightDirective } from './app.directive'; @NgModule({

imports: [BrowserModule], declarations: [AppComponent,HighLightDirective], bootstrap: [AppComponent] }) export class AppModule { }
```

3)app.directive.ts

4 . 2 .1) Basic Directives

```
import { Directive, ElementRef, HostListener } from '@angular/co re'; @Directive({
        selector: '[highlight]' }) export class HighLightDirective {
constructor(private element: ElementRef) { } @HostListener('mouseenter') onMouseEnter() {
        this.setAppearance('#aaaaaa', 'pointer'); } @HostListener('mouseleave') onMouseLeave() {
        this.setAppearance(null, null); } setAppearance(color: string, cursor: string) {
let style = this.element.nativeElement.style; style.backgroundColor = color; style.cursor =
cursor; } }
```

Sending data to directives

1)app.component.ts

```
import { Component } from '@angular/core'; @Component({

selector: 'project-app', template: '<ul><li *ngFor="let user of users" highlight="#00
0000">{{user}}</li></ul>' }) export class AppComponent {

users; constructor() {

    this.users = ["s1","s2","s3"]; } }
```

2)app.module.ts

```
import { NgModule } from '@angular/core'; import { BrowserModule } from
"@angular/platform-browser"; import { AppComponent } from './app.component'; import {
HighLightDirective } from './app.directive'; @NgModule({

imports: [BrowserModule], declarations: [AppComponent,HighLightDirective], bootstrap:
[AppComponent] }) export class AppModule { }
```

3)app.directive.ts

4 . 2 . 2) Sending data to directives

```
import { Directive, ElementRef, HostListener } from '@angular/co re'; import { Input} from
'@angular/core'; @Directive({
        selector: '[highlight]' }) export class HighLightDirective {
@Input('highlight') backgroundColor: string; constructor(private element: ElementRef) { }
@HostListener('mouseenter') onMouseEnter() {
        this.setAppearance(this.backgroundColor, 'pointer'); } @HostListener('mouseleave')
onMouseLeave() {
        this.setAppearance(null, null); } setAppearance(color: string, cursor: string) {
let style = this.element.nativeElement.style; style.backgroundColor = color; style.cursor =
cursor; } }
```

Built in pipes

1)app.module.ts

```
import { NgModule } from '@angular/core'; import { BrowserModule } from
"@angular/platform-browser"; import { AppComponent } from './app.component'; import {
StudentListComponent } from "./student-list.component"; @NgModule({

imports: [BrowserModule], declarations: [AppComponent,StudentListComponent], bootstrap:
[AppComponent] }) export class AppModule { }
```

2)app.component.ts

```
import { Component, OnInit } from '@angular/core'; @Component({

selector: 'project', template: `<div class="container-fluid">

<h1>Student Manager</h1> <student-list-view></student-list-view> </div>` }) export class
AppComponent{}
```

3)student-list.template.html

5 . 1) Built in pipes

```
<div class="row">
<div class="col-sm-12">
<button class="btn btn-primary btn-lg">
    Add new student </button> </div> </div> <h2>List of Students</h2> <div class="row"
*ngFor="let student of students">
<div class="col-sm-8">
        <h4>{{student.id}}: {{student.name | lowercase}} : {{student .pocketMoney |
currency:user.currenyFormat }}:
        {{student.jeeScore }} : {{student.attemptDate | date: user.d ateFormat }}</h4>
</div> {{student.pocketMoney}} </div>
<div class="row" *ngFor="let student of students">
<div class="col-sm-8">
        <h4>Score out of 10 : {{student.jeeScore/300 | number: '2.2 -2' }}</h4>
    </div> </div>
<div class="row" *ngFor="let student of students">
<div class="col-sm-8">
      <h4>Percentage : {{student.jeeScore/3000 | percent: '3.1-2' }}</h4> </div> </div>
<div class="row" *ngFor="let student of students">
<div class="col-sm-8">
    <h4>{{student | json }}</h4> </div> </div>
```

4)student-list.component.ts

5 . 1) Built in pipes

```
import {Component} from '@angular/core'; import {Student} from './student'; @Component({

selector: 'student-list-view', templateUrl: 'student-list.template.html' }) export class
StudentListComponent {

user = {'currenyFormat':'INR',dateFormat':'dd/MM/yyyy'}; students = Student.students; }
```

5)student.ts

```
export class Student {

id: number; name: string; pocketMoney: number; jeeScore:number; attemptDate:Date;

static students: Student[] = [

                { id: 1, name: 'student1',pocketMoney:1000,jeeScore:2204 ,attemptDate: new
Date("9/27/2017 11:25")},

                { id: 2, name: 'student2',pocketMoney:5000,jeeScore:2876 ,attemptDate: new
Date("9/27/2016 11:25")},

                { id: 3, name: 'student3',pocketMoney:2500,jeeScore:2600 ,attemptDate: new
Date("9/27/2015 11:25")},

                { id: 4, name: 'student4',pocketMoney:7000,jeeScore:2800 ,attemptDate: new
Date("9/27/2013 11:25")}

        ]; }
```

5 . 2) Slice Pipe and Pagination at client side

Client side pagination with slice pipe

1)app.module.ts

```
import {NgModule} from '@angular/core'; import {BrowserModule} from
'@angular/platform-browser'; import {NumberListComponent} from './number-list.component';
@NgModule({

imports: [BrowserModule], declarations: [NumberListComponent], bootstrap:
[NumberListComponent] }) export class AppModule { }
```

2)number-list.template.html

```
<div class="container-fluid">

<h1>My Numbers</h1> <span *ngFor="let num of numbers | slice:start:end"

class="number"> {{num}} </span> <button (click)="previous();"
[disabled]="start-1<1"><<</butto n>

   <button (click)="next();" [disabled]="end+1>100">>></button> </div>
```

3)number-list.component.ts

```
import {Component} from '@angular/core'; @Component({

selector: 'project', templateUrl: 'number-list.template.html', styles: [`

.number {

display: inline-block; background: #e0e0e0; border-radius: 4px; margin: 4px;
```

```
    padding: 4px 8px; } `] }) export class NumberListComponent {
numbers: number[] = []; start = 1; end = 11; pageSize=10; constructor() {
for (let i = 0; i < 100; i++) {
    this.numbers.push(i); } } toValue(input: string, defValue) {
var value = parseInt(input); if (isNaN(value)) {
    return defValue; } else {
    return value; } } previous(){
this.start = this.start - this.pageSize; this.end = this.end - this.pageSize; } next(){
this.start = this.start + this.pageSize; this.end = this.end + this.pageSize; } }
```

Custom Pipes

1)app.module.ts

```
import { NgModule } from '@angular/core'; import { BrowserModule } from
"@angular/platform-browser"; import { AppComponent } from './app.component'; import {
StudentListComponent } from "./student-list.component"; import {ContentFilterPipe} from
"./content-filter.pipe"; @NgModule({

imports: [BrowserModule], declarations: [AppComponent,StudentListComponent,ContentFilterPipe],

        bootstrap: [AppComponent] }) export class AppModule { }
```

2)app.component.ts

```
import { Component, OnInit } from '@angular/core'; @Component({

moduleId: module.id, selector: 'project', template: `<div class="container-fluid">

<h1>Student Manager</h1> <student-list-view></student-list-view> </div>` }) export class
AppComponent{}
```

3)content-filter.pipe.ts

5 . 3) Custom pipes

```
import {Pipe, PipeTransform} from '@angular/core'; import {Student} from './student';
@Pipe({name: 'contentFilter', pure: false}) export class ContentFilterPipe implements
PipeTransform {

transform(value: Student[], searchFor: string) : Student[] {

if (!searchFor) return value; searchFor = searchFor.toLowerCase(); return value.filter(student =>
student.name.indexOf(searchFo r) >= 0);

    } }
```

4)student-list.component.ts

```
import {Component} from '@angular/core'; import {Student} from './student'; @Component({

selector: 'student-list-view', templateUrl: 'student-list.template.html' }) export class
StudentListComponent {

    students = Student.students; }
```

5)student-list.template.html

```
<div class="row">

<div class="col-sm-12">

Search <div class="col-sm-4 col-sm-offset-5"> <input #searchBox class="form-control input-lg"

placeholder="Search" (keyup)="0" /> </div> </div>
```

5 . 3) Custom pipes

```
<h2>List of Students</h2> <div class="row" *ngFor="let student of students | contentFilter
:searchBox.value">
<div class="col-sm-8">
        <h4>{{student.id}}: {{student.name | lowercase}} : {{student .pocketMoney |
currency:'INR' }}:
                {{student.jeeScore }} : {{student.attemptDate | date: ' short' }}</h4>
    </div> </div>
```

6)student.ts

```
export class Student {
id: number; name: string; pocketMoney: number; jeeScore:number; attemptDate:Date; static
students: Student[] = [ { id: 1, name: 'student1',pocketMoney:1000,jeeScore:2204,att emptDate:
new Date("9/27/2017 11:25")},
        { id: 2, name: 'student2',pocketMoney:5000,jeeScore:2876,att emptDate: new
Date("9/27/2016 11:25")},
        { id: 3, name: 'student3',pocketMoney:2500,jeeScore:2600,att emptDate: new
Date("9/27/2015 11:25")},
        { id: 4, name: 'student4',pocketMoney:7000,jeeScore:2800,att emptDate: new
Date("9/27/2013 11:25")}
        ]; }
```

6 . 1) Basic Service

1)app.component.html

```
<table>

<tr><td>Id</td><td>Name</td></tr> <tr *ngFor="let student of students">

<td>{{student.id}}</td> <td>{{student.name}}</td> </tr> </table>
```

2)app.component.ts

```
import { Component, OnInit } from '@angular/core'; import { Student } from "./student"; import
{ StudentService } from "./student.service"; @Component({

selector: 'project', templateUrl: 'app.component.html' }) export class AppComponent {
students:Student[] ; studentService:StudentService constructor(){

this.studentService= new StudentService(); this.students = this.studentService.getStudents(); } }
```

3)app.module.ts

6 . 1) Basic Service

```
import { NgModule } from '@angular/core'; import { BrowserModule } from
"@angular/platform-browser"; import { AppComponent } from './app.component'; import {
StudentService } from "./student.service"; @NgModule({

imports: [BrowserModule], declarations: [AppComponent], providers:[StudentService],
bootstrap: [AppComponent], }) export class AppModule { }
```

4)student.service.ts

```
import { Injectable } from '@angular/core'; import { Student } from "./student"; @Injectable()
export class StudentService {

getStudents(){

        return [new Student(1,"s1"),new Student(2,"s2")]; } }
```

5)student.ts

```
export class Student{

id:number; name:string; constructor(a,b){

this.id=a; this.name=b; } }
```

6 . 2) Dependency Injection

1)app.component.html

```
<table>
<tr><td>Id</td><td>Name</td></tr> <tr *ngFor="let student of students">
<td>{{student.id}}</td> <td>{{student.name}}</td> </tr> </table>
```

2)app.component.ts

```
import { Component, OnInit } from '@angular/core'; import { Student } from "./student"; import
{ StudentService } from "./student.service"; @Component({
selector: 'project', templateUrl: 'app.component.html' }) export class AppComponent {
students:Student[] ; constructor(private studentService:StudentService){
    this.students = this.studentService.getStudents(); } }
```

3)app.module.ts

```
import { NgModule } from '@angular/core'; import { BrowserModule } from
"@angular/platform-browser"; import { AppComponent } from './app.component'; import {
StudentService } from "./student.service"; @NgModule({
imports: [BrowserModule], declarations: [AppComponent], providers:[StudentService],
bootstrap: [AppComponent], }) export class AppModule { }
```

4)student.service.ts

```
import { Injectable } from '@angular/core'; import { Student } from "./student"; @Injectable()
export class StudentService {
constructor() { } getStudents(){
        return [new Student(1,"s1"),new Student(2,"s2")]; } }
```

5)student.ts

```
export class Student{
id:number; name:string; constructor(a,b){
this.id=a; this.name=b; } }
```

7 . 1) Basic Http Calls for the server side

1)app.component.html

```
<button (click)="load();">Load</button> <table>
<tr><td>Id</td><td>Name</td></tr> <tr *ngFor="let student of students">
<td>{{student.id}}</td> <td>{{student.name}}</td> </tr> </table>
```

2)app.component.ts

```
import { Component, OnInit } from '@angular/core'; import { Student } from "./student"; import
{ StudentService } from "./student.service"; @Component({
selector: 'project', templateUrl: 'app.component.html' }) export class AppComponent {
students:Student[] ; constructor(private studentService:StudentService){
        this.students = []; } load(){
this.studentService.getStudents().subscribe(
        (data)=>this.students=data ); } }
```

3)app.module.ts

```
import { NgModule } from '@angular/core'; import { BrowserModule } from
"@angular/platform-browser"; import { AppComponent } from './app.component'; import {
StudentService } from "./student.service"; import { HttpModule } from "@angular/http";
@NgModule({
imports: [BrowserModule,HttpModule], declarations: [AppComponent],
providers:[StudentService], bootstrap: [AppComponent], }) export class AppModule { }
```

4)student.service.ts

```
import { Injectable } from '@angular/core'; import { Student } from "./student"; import { Http }
from "@angular/http"; import { Observable } from "rxjs"; import 'rxjs/add/operator/map';
@Injectable() export class StudentService {
constructor(private http:Http) { } getStudents(){
            return this.http.get('http://localhost:3000/students').m ap(
      (response) => response.json() ); } }
```

5)student.ts

7 . 1) Basic Http Calls for the server side

```
export class Student{
id:number; name:string; constructor(a,b){
this.id=a; this.name=b; } }
```

7 . 2) With Complete Rest API Calls

Rest API calls

```
import { Injectable } from '@angular/core'; import { Http } from "@angular/http"; import {
Student } from "./student"; import { Observable } from "rxjs"; import "rxjs/add/operator/map";
@Injectable() export class StudentService {
constructor(private http:Http) { } insertStudent(student:Student){
                return this.http.post("http://localhost:3000/students",s tudent).
                        map((response) => response.json()); } getStudents() {
return this.http.get("http://localhost:3000/students").
                        map((response) => response.json()); } getStudentById(id:number) {
            return this.http.get("http://localhost:3000/students/"+id) .
                        map((response) => response.json()); } deleteStudent(student:Student){
            return this.http.delete("http://localhost:3000/students/"+ student.id).
                        map((response) => response.json()); } updateStudent(student:Student){
            return this.http.put("http://localhost:3000/students/"+stu dent.id,student).
                        map((response) => response.json()); } }
```

7 . 2) With Complete Rest API Calls

7 . 3) With Complete Application

1)app.component.html

```
<h1>Student Manager</h1> <student-list></student-list> <student-form></student-form>
<student-update></student-update> <student-delete></student-delete>
```

2)app.component.ts

```
import { Component, OnInit } from '@angular/core'; @Component({

selector: 'app', templateUrl: 'app.component.html' }) export class AppComponent { }
```

3)app.module.ts

```
import { StudentDeleteComponent } from './student-delete.compone nt'; import {
StudentUpdateComponent } from './student-update.compone nt'; import { FormsModule } from
'@angular/forms'; import { StudentFormComponent } from './student-form.component'; import {
BrowserModule } from '@angular/platform-browser'; import { StudentService } from
'./student.service'; import { StudentListComponent } from './student-list.component'; import {
NgModule } from '@angular/core'; import { AppComponent } from './app.component'; import {
HttpModule } from "@angular/http";
```

```
@NgModule({

imports: [BrowserModule,HttpModule,FormsModule], declarations:
[AppComponent,StudentListComponent,StudentForm Component,StudentUpdateComponent,

                    StudentDeleteComponent], providers: [StudentService],
bootstrap:[AppComponent] }) export class AppModule { }
```

4)student-delete.component.html

```
<form>

        Id <input type="text" name="id" [(ngModel)]="student.id" (change)="load();"/><br/>

Name:{{student.name}}<br/> <button (click)="delete();">delete</button> </form>
```

5)student-delete.component.ts

```
import { StudentService } from './student.service'; import { Student } from './student'; import {
Component, OnInit } from '@angular/core'; @Component({

selector: 'student-delete', templateUrl: 'student-delete.component.html' }) export class
StudentDeleteComponent implements OnInit {

student:Student; constructor(private ss:StudentService) { } ngOnInit() {

    this.student = new Student(); }
load(){
this.ss.getStudentById(this.student.id).subscribe(
        (data) => this.student = data ) } delete(){
this.ss.deleteStudent(this.student.id).subscribe(
        (data) => console.log(data) ) } }
```

6)student-form.component.html

```
<form>
        Id <input type="text" name="id" [(ngModel)]="student.id"/><b r/>
        Name <input type="text" name="name" [(ngModel)]="student.nam e"/><br/>
        <button (click)="insert();">Insert</button> </form>
```

)student-form.component.ts

7 . 3) With Complete Application

```
import { StudentService } from './student.service'; import { Student } from './student'; import {
Component, OnInit } from '@angular/core'; @Component({
selector: 'student-form', templateUrl: 'student-form.component.html' }) export class
StudentFormComponent implements OnInit {
student:Student; constructor(private ss:StudentService) { } ngOnInit() {
      this.student = new Student(); } insert(){
this.ss.insertStudent(this.student).subscribe(
        (data) => console.log(data) ) } }
```

7)student-list.component.html

```
<table>
<tr *ngFor="let student of students">
                                                  <td>{{student.id}}</td> <td>{{student.name}}</td> </tr>
</table>
```

8)student-list.component.ts

```
import { StudentService } from './student.service'; import { Student } from './student'; import {
Component, OnInit } from '@angular/core'; @Component({
selector: 'student-list', templateUrl: 'student-list.component.html' }) export class
StudentListComponent implements OnInit {
students:Student[]; constructor(private ss:StudentService) { } ngOnInit() {
this.ss.getStudents().subscribe(
        (data) => this.students = data ) } }
```

9)student-update.component.html

```
<form>
                Id <input type="text" name="id" [(ngModel)]="student.id" /><br/>
                Name <input type="text" name="name" [(ngModel)]="student .name"/><br/>
        <button (click)="update();">Update</button> </form>
```

10)student-update.component.ts

```
import { StudentService } from './student.service'; import { Student } from './student'; import {
Component, OnInit } from '@angular/core'; @Component({

selector: 'student-update', templateUrl: 'student-update.component.html' }) export class
StudentUpdateComponent implements OnInit {

student:Student; constructor(private ss:StudentService) { } ngOnInit() {

      this.student = new Student(); } update(){

this.ss.updateStudent(this.student).subscribe(

        (data) => console.log(data) ) } }
```

11)student.service.ts

7 . 3) With Complete Application

```
import { Student } from './student'; import { Injectable } from '@angular/core'; import { Http }
from "@angular/http"; import { Observable } from "rxjs"; import 'rxjs/add/operator/map';
@Injectable() export class StudentService {
constructor(private http:Http) { } getStudents(){
        return this.http.get("http://localhost:3000/students").m ap(
    (response)=>response.json() ) } getStudentById(id:number){
        return this.http.get("http://localhost:3000/students/"+i d).map(
    (response)=>response.json() ) } insertStudent(student:Student){
        return this.http.post("http://localhost:3000/students",s tudent).map(
    (response)=>response.json() ) } updateStudent(student:Student){
        return this.http.put("http://localhost:3000/students/"+s tudent.id,student).map(
    (response)=>response.json() ) } deleteStudent(id:number){
        return this.http.delete("http://localhost:3000/students/ "+id).map(
    (response)=>response.json() ) } }
```

12)student.ts

```
export class Student{
id:number; name:string;
}
```

8 . 1) Basic

1)app.component.html

```
<h1>Student Manager</h1> <router-outlet></router-outlet>
```

2)app.component.ts

```
import { Component, OnInit } from '@angular/core';
@Component({
selector: 'app', templateUrl: 'app.component.html' })
export class AppComponent implements OnInit {
constructor() { }
        ngOnInit() { } }
```

3)app.module.ts

```
import { routingModule } from './app.route'; import { StudentDeleteComponent } from
'./student-delete.compone nt'; import { StudentUpdateComponent } from
'./student-update.compone nt'; import { FormsModule } from '@angular/forms'; import {
StudentFormComponent } from './student-form.component'; import { BrowserModule } from
'@angular/platform-browser'; import { StudentService } from './student.service'; import {
StudentListComponent } from './student-list.component'; import { NgModule } from
'@angular/core';

import { AppComponent } from './app.component';

import { HttpModule } from "@angular/http";

@NgModule({

        imports: [BrowserModule,HttpModule,FormsModule,routingModule ],

        declarations: [AppComponent,StudentListComponent,StudentForm
Component,StudentUpdateComponent,

                                StudentDeleteComponent], providers: [StudentService],
bootstrap:[AppComponent] }) export class AppModule { }
```

4)app.route.ts

```
import { StudentListComponent } from './student-list.component'; import {
StudentDeleteComponent } from './student-delete.compone nt'; import {
StudentUpdateComponent } from './student-update.compone nt'; import {
StudentFormComponent } from './student-form.component'; import { Routes,RouterModule }
from "@angular/router";

const routes:Routes=[

{

path:'create', component:StudentFormComponent }, {

path:'update/:id', component:StudentUpdateComponent }, {

path:'delete/:id', component:StudentDeleteComponent }, {

path:'list', component:StudentListComponent }, {

path:'', redirectTo:'/list', pathMatch:'full' } ];

export const routingModule = RouterModule.forRoot(routes);
```

5)student-delete.component.html

```
<form>
            Id <input type="text" name="id" [(ngModel)]="student.id" (change)="load();"/><br/>
Name:{{student.name}}<br/>
       <button (click)="delete();">delete</button> </form>
```

6)student-delete.component.ts

```
import { ActivatedRoute,Router } from '@angular/router'; import { StudentService } from
'./student.service'; import { Student } from './student'; import { Component, OnInit } from
'@angular/core';
@Component({
selector: 'student-delete', templateUrl: 'student-delete.component.html' })
export class StudentDeleteComponent implements OnInit {
student:Student;
        constructor(private ss:StudentService,private route:Activate dRoute,private router:Router)
{ }
ngOnInit() {
let id = this.route.snapshot.params["id"]; this.load(id); }
load(id) {
this.ss.getStudentById(id).subscribe(
        (data) => this.student= data ) }
delete(){
this.ss.deleteStudent(this.student.id).subscribe( (data) => this.router.navigate(["/list"]) ) }
}
```

7)student-form.component.html

8 . 1) Basic

```
<form>
        Id <input type="text" name="id" [(ngModel)]="student.id"/><b r/>
        Name <input type="text" name="name" [(ngModel)]="student.nam e"/><br/>
        <button (click)="insert();">Insert</button> </form>
```

8)student-form.component.ts

```
import { StudentService } from './student.service'; import { Student } from './student'; import {
Component, OnInit } from '@angular/core';
@Component({
selector: 'student-form', templateUrl: 'student-form.component.html' })
export class StudentFormComponent implements OnInit {
student:Student;
constructor(private ss:StudentService) { }
ngOnInit() {
    this.student = new Student(); }
insert(){
this.ss.insertStudent(this.student).subscribe(
     (data) => console.log(data) ) }
}
```

9)student-list.component.html

```
<a [routerLink]="['/create']">Register</a> <table>
<tr *ngFor="let student of students">
<td>{{student.id}}</td> <td>{{student.name}}</td> <td><a
[routerLink]="['/update',student.id]">Update</a>< /td>
                <td><a [routerLink]="['/delete',student.id]">Delete</a>< /td>
</tr>
</table>
```

)student-list.component.ts

```
import { StudentService } from './student.service'; import { Student } from './student'; import {
Component, OnInit } from '@angular/core';
@Component({
selector: 'student-list', templateUrl: 'student-list.component.html' })
export class StudentListComponent implements OnInit {
students:Student[];
constructor(private ss:StudentService) { }
ngOnInit() {
this.ss.getStudents().subscribe(
        (data) => this.students = data ) } }
```

10)student-update.component.html

```
<form>
        Id <input type="text" name="id" [(ngModel)]="student.id" /><br/>
        Name <input type="text" name="name" [(ngModel)]="student .name"/><br/>
    <button (click)="update();">Update</button> </form>
```

11)student-update.component.ts

```
import { ActivatedRoute, Router } from '@angular/router'; import { StudentService } from
'./student.service'; import { Student } from './student'; import { Component, OnInit } from
'@angular/core';
@Component({
selector: 'student-update', templateUrl: 'student-update.component.html' })
export class StudentUpdateComponent implements OnInit {
student:Student;
    constructor(private ss:StudentService,private route:Activate dRoute,private router:Router)
{ }
ngOnInit() {
let id = this.route.snapshot.params["id"]; this.load(id); }
load(id) {
this.ss.getStudentById(id).subscribe(
    (data) => this.student= data ) }
update(){
this.ss.updateStudent(this.student).subscribe( (data) => this.router.navigate(["/list"]); ) }
}
```

12)student.service.ts

8 . 1) Basic

```
import { Student } from './student'; import { Injectable } from '@angular/core';
import { Http } from "@angular/http";
import { Observable } from "rxjs";
import 'rxjs/add/operator/map';
@Injectable() export class StudentService {
constructor(private http:Http) { }
getStudents(){
        return this.http.get("http://localhost:3000/students").m ap(
    (response)=>response.json() ) }
getStudentById(id:number){
        return this.http.get("http://localhost:3000/students/"+i d).map(
    (response)=>response.json() ) }
insertStudent(student:Student){
        return this.http.post("http://localhost:3000/students",s tudent).map(
    (response)=>response.json() ) }
updateStudent(student:Student){
        return this.http.put("http://localhost:3000/students/"+s tudent.id,student).map(
(response)=>response.json()
```

```
        ) }
deleteStudent(id:number){
                return this.http.delete("http://localhost:3000/students/ "+id).map(
        (response)=>response.json() ) }load(){ this.ss.getStudentById(this.student.id).subscribe(
(data) => this.student = data ) } delete(){ this.ss.deleteStudent(this.student.id).subscribe( (data)
=> console.log(data) ) } }
}
```

13)student.ts

```
export class Student{
                                                              id:number; name:string; }
```

8. 2) Child Routes

1)app.component.html

```
<a [routerLink]="['home']">Home</a> <a [routerLink]="['product']">Product</a> <a
[routerLink]="['contact']">Contact us</a> <br/> <router-outlet></router-outlet>
```

2)app.component.ts

```
import { Component } from '@angular/core'; @Component({

selector: 'app', templateUrl: './app.component.html', styleUrls: ['./app.component.css'] }) export
class AppComponent { }
```

3)app.module.ts

```
import { BrowserModule } from '@angular/platform-browser'; import { NgModule } from
'@angular/core'; import { FormsModule } from '@angular/forms'; import { HttpModule } from
'@angular/http'; import { RouterModule } from '@angular/router'; import { AppComponent }
from './app.component'; import { HomeComponent} from './home.component' import {
ContactComponent} from './contact.component' import { ProductComponent} from
'./product.component' import { ErrorComponent} from './error.component' import {
ProductDetailComponent} from './product-detail.componen t' import { ProductService } from
'./product.service'; import { appRoutes } from './app.routes';
```

```
@NgModule({
declarations: [
        AppComponent,HomeComponent,ContactComponent,ProductComponent
,ErrorComponent,ProductDetailComponent
], imports: [
        BrowserModule,FormsModule,HttpModule,RouterModule.forRoot(ap pRoutes)
], providers: [ProductService], bootstrap: [AppComponent] }) export class AppModule { }
```

4)app.routes.ts

```
import { Routes } from '@angular/router'; import { HomeComponent} from './home.component'
import { ContactComponent} from './contact.component' import { ProductComponent} from
'./product.component' import { ErrorComponent} from './error.component' import {
ProductDetailComponent} from './product-detail.componen t' export const appRoutes: Routes = [
{ path: 'home', component: HomeComponent }, { path: 'contact', component: ContactComponent
}, { path: 'product', component: ProductComponent,
children: [
    { path: 'detail/:id', component: ProductDetailComponent } ] }, { path: '', redirectTo: 'home',
pathMatch: 'full' }, { path: '**', component: ErrorComponent } ];
```

5)contact.component.ts

```
import {Component} from '@angular/core'; @Component({
template: `<h1>Contact Us</h1>
<p>RK </p> ` }) export class ContactComponent { }
```

6)error.component.ts

```
import {Component} from '@angular/core'; @Component({
template: `<h1>Page not found</h1>
  <p>This is a Error Page</p> ` }) export class ErrorComponent { }
```

7)home.component.ts

```
import {Component} from '@angular/core'; @Component({
template: `<h1>Welcome!</h1>
  <p>This is Home Component </p> ` }) export class HomeComponent { }
```

8)product-detail.component.html

```
Product Details Page <br/> product : {{product.name}} price : {{ product.price}} <p>
    <a (click)="onBack()">Back </a> </p>
```

9)product-detail.component.ts

```
import { Component, OnInit } from '@angular/core'; import { Router,ActivatedRoute } from
'@angular/router'; import { ProductService } from './product.service'; import { Product } from
'./product'; @Component({
    templateUrl: './product-detail.component.html', }) export class ProductDetailComponent{
product:Product; id; constructor(private _Activatedroute:ActivatedRoute,
private _router:Router, private _productService:ProductService){ } onBack(): void {
      this._router.navigate(['product']); } sub;
ngOnInit() {
this.sub=this._Activatedroute.params.subscribe(params => {
this.id = params['id']; let products=this._productService.getProducts();
this.product=products.find(p => p.productID==this.id);
      }); } ngOnDestroy() {
    this.sub.unsubscribe(); } }
```

10)product.component.html

```
Product List <div class='table-responsive'>

<table class='table'>

<thead>

<tr>

<th>ID</th> <th>Name</th> <th>Price</th> </tr> </thead> <tbody>

<tr *ngFor="let product of products;"> <td>{{product.productID}}</td> <td><a
[routerLink]="['detail',product.productID ]">{{product.name}} </a> </td>

        <td>{{product.price}}</td> </tr> </tbody> </table> </div>
<router-outlet></router-outlet>
```

11)product.component.ts

```
import { Component, OnInit } from '@angular/core'; import { ProductService } from
'./product.service'; import { Product } from './product'; @Component({

    templateUrl: './product.component.html', }) export class ProductComponent{

products:Product[]; constructor(private productService:ProductService){ } ngOnInit() {

    this.products=this.productService.getProducts(); } }
```

12)product.service.ts

```
import { Observable } from 'rxjs/Observable'; import {Product} from './Product'; export class ProductService{

public getProducts() {

let products:Product[]; products=[

new Product(1,'Memory Card',500), new Product(2,'Pen Drive',750), new Product(3,'Power Bank',100) ] return products; } public getProduct(id) {

let products:Product[]=this.getProducts(); return products.find(p => p.productID==id); } }
```

13)product.ts

```
export class Product {

        constructor(productID:number, name: string , price: number) {

this.productID=productID; this.name=name; this.price=price; } productID:number ; name: string ; price:number;

}
```