Catalog

app.module.ts

```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';
import {HttpClientModule} from '@angular/common/http'

@NgModule({
  declarations: [
    AppComponent,
  ],
  imports: [
    BrowserModule,
    AppRoutingModule,
    HttpClientModule
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

ng g c components/blog

blog.component.ts

```
import { Component, OnInit } from '@angular/core';
import { HttpClient } from '@angular/common/http';
@Component({
  selector: 'app-blog',
  templateUrl: './blog.component.html',
  styleUrls: ['./blog.component.scss']
})
export class BlogComponent implements OnInit {

  constructor(private http:HttpClient) { }

  ngOnInit(): void {
    this.http.get("https://jsonplaceholder.typicode.com/posts").subscribe((dat
a)=>console.log(data))
  }

}
```

app-routing.module.ts

```
import { NgModule } from '@angular/core';
import { RouterModule, Routes } from '@angular/router';
import { BlogComponent } from './components/blog/blog.component';
const routes: Routes = [
  {
    path:'blog',
    component:BlogComponent
  }
];

@NgModule({
  imports: [RouterModule.forRoot(routes)],
  exports: [RouterModule]
})
export class AppRoutingModule { }
```

app.component.html

```
<a routerLink="blog"> Blog</a>
<router-outlet></router-outlet>
```

blog.component.ts

```
import { Component, OnInit } from '@angular/core';
import { HttpClient } from '@angular/common/http';
@Component({
  selector: 'app-blog',
  templateUrl: './blog.component.html',
  styleUrls: ['./blog.component.scss']
})
export class BlogComponent implements OnInit {
posts;
  constructor(private http:HttpClient) { }

  ngOnInit(): void {
    this.http.get("https://jsonplaceholder.typicode.com/posts").subscribe((dat
a)=>this.displayPost(data))
  }
displayPost(data){
this.posts = data;
}
}
```

blog.component.html

```
<h1>Blog posts</h1>
<div *ngFor="let post of posts">
<h3>{{post.title}}</h3>
<p>{{post.body}}</p>
</div>
```

tsconfig.json

```
"noImplicitAny": false,
```

App.module.ts

```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import { AuthModule } from './auth/auth.module';
import { FormsModule,ReactiveFormsModule  } from '@angular/forms';

import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';


@NgModule({
  declarations: [
    AppComponent
  ],
  imports: [
    BrowserModule,
    AppRoutingModule,
    AuthModule,
    FormsModule,
    ReactiveFormsModule
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

App.component.ts

```
import { Component } from '@angular/core';
import { FormGroup, FormControl, Validators } from '@angular/forms';

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {
  title = 'abhi';
  todaydate: any;
  componentproperty: any;
  emailid: any;
  formdata: any;
  ngOnInit() {
```

```
    this.formdata = new FormGroup({
      emailid: new FormControl("", Validators.compose([
        Validators.required,
        Validators.pattern("[^ @]*@[^ @]*")
      ])),
      passwd: new FormControl("")
    });
  }
  onClickSubmit(data: any) {
    alert("Entered Email id : " + data.emailid);
  }
}
```

App.component.html

```
<app-signup></app-signup>
<app-login></app-login>
<div>
  <form [formGroup]="formdata" (ngSubmit)="onClickSubmit(formdata.value)"> <input
type="text" class="fortextbox"
      name="emailid" placeholder="emailid" formControlName="emailid"> <br />
<input type="password" class="fortextbox"
      name="passwd" placeholder="passwd" formControlName="passwd"> <br /> <input
type="submit"
      [disabled]="!formdata.valid" class="forsubmit" value="Log In"> </form>
</div>
<p> Email entered is : {{emailid}} </p>
```

Auth.module.ts

```
import { NgModule } from '@angular/core';
import { CommonModule } from '@angular/common';
import { LoginComponent } from './login/login.component';
import { SignupComponent } from './signup/signup.component';



@NgModule({
  declarations: [
    LoginComponent,
    SignupComponent
  ],
  imports: [
```

```
    CommonModule
  ],
  exports: [
    LoginComponent,
    SignupComponent
  ]
})
export class AuthModule { }
```

ng g m auth

ng g c auth/login

ng g c auth/signup

auth.module.ts

```
import { NgModule } from '@angular/core';
import { CommonModule } from '@angular/common';
import { LoginComponent } from './login/login.component';
import { SignupComponent } from './signup/signup.component';



@NgModule({
  declarations: [
    LoginComponent,
    SignupComponent
  ],
  imports: [
    CommonModule
  ],
exports:[
LoginComponent,
SignupComponent
]
})

export class AuthModule { }
```

login.component.html

```
<h1>Login Component</h1>
```

signup.component.html

```
<h1>Signup Component</h1>
```

app.module.ts

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

import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';

@NgModule({
  declarations: [
    AppComponent
  ],
  imports: [
    BrowserModule,
    AppRoutingModule,
    AuthModule
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

app.component.html

```
<app-signup></app-signup>
<app-login></app-login>
```

Pipes

app.component.ts

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

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.scss']
})
export class AppComponent {
  title = 'myAngularApp';
  currentdate = new Date();
  js = {name: 'Alex', age: '25', address:{a1: 'Paris', a2: 'France'}};
  courses = ['HTML', 'CSS', 'JavaScript', 'Node js', 'Angular', 'React js',
'PHP'];
}
```

app.component.html

```
    <h1>Uppercase Pipe</h1>
    <b>{{title | uppercase}}</b><br/>

    <h1>Lowercase Pipe</h1>
    <b>{{title | lowercase}}</b>

    <h1>Date pipe</h1>
    <b>{{currentdate | date:'yyyy'}}</b><br/>
    <b>{{currentdate | date:'fullDate'}}</b>

    <h1>Decimal Pipe</h1>
    <b>{{ 278.89 | number: '3.4' }}</b>

    <h1>Currency Pipe</h1>
    <b>{{6589.23 | currency:"USD"}}</b><br/>

    <h1>Json Pipe</h1>
    <b>{{ js | json }}</b>

    <h1>Percent Pipe</h1>
    <b>{{00.7645 | percent}}</b>

    <h1>Slice Pipe</h1>
    <b>{{courses | slice:2:6}}</b>
```

app.module.ts

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

import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';

@NgModule({
  declarations: [
    AppComponent
  ],
  imports: [
    BrowserModule,
    AppRoutingModule,
    FormsModule
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

Decimal pipe

app.component.ts

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

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.scss']
})
export class AppComponent {
  title = 'myAngularApp';
  val=0;
}
```

app.module.ts

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

import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';

@NgModule({
```

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

app.component.html

```
<input type="text" [(ngModel)]="val">
<h1>Decimal Pipe</h1>
<b>{{ val | number: '3.4' }}</b>
```

Terminal Code:

ng g s newservice

newservice.service.ts file

```
//Dependency Injection
import { Injectable } from '@angular/core';

@Injectable({
  providedIn: 'root'
})
export class NewserviceService {
  employees: any[];
  constructor() {
    this.employees = [
      {
        empcode : '001', name : 'Chandan Kumar'
      },
      {
        empcode : '002', name : 'Ajeet Kumar'
      }
    ];

  display () {
    return 'Welcome to ShoSoft Solutions';

  }
}
```

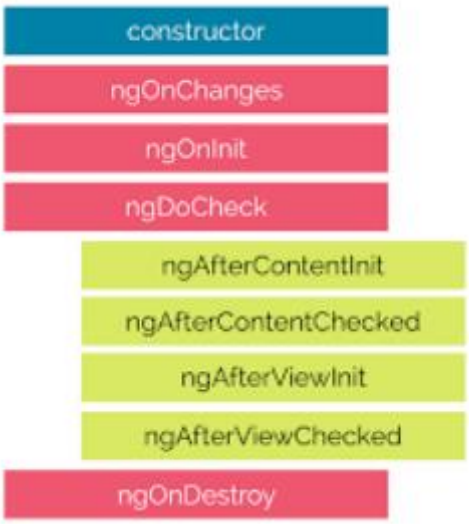
App.component.ts File

```
import { Component } from '@angular/core';
import { NewserviceService } from './newservice.service';

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {


  //For newService
  title = 'ShoSoft Solutions';
  text:string="";
  emp:any[] =[];
  constructor (private _newservice: NewserviceService){}
  ngOnInit () {
    this.emp = this._newservice.employees;
    this.text = this._newservice.display();
  }
}


//ngFor
//ngClass ngStyle ngif
```

App.component.html

```
<h1>
  Welcome to {{ title}}
</h1>
<h2>{{text}}</h2>
<table>
  <tr *ngFor = "let em of emp">
    <td>Emp Code: </td> {{em.empcode}}
    <td>Name : </td>{{em.name}}
  </tr>
```

Angular application goes through an entire set of processes or has a lifecycle right from its initiation to the end of the application. The representation of lifecycle in pictorial representation as follows,



The description of each lifecycle method is as below,

i. **ngOnChanges:** When the value of a data bound property changes, then this method is called.
ii. **ngOnInit:** This is called whenever the initialization of the directive/component after Angular first displays the data-bound properties happens.
iii. **ngDoCheck:** This is for the detection and to act on changes that Angular can't or won't detect on its own.
iv. **ngAfterContentInit:** This is called in response after Angular projects external content into the component's view.
v. **ngAfterContentChecked:** This is called in response after Angular checks the content projected into the component.
vi. **ngAfterViewInit:** This is called in response after Angular initializes the component's views and child views.
vii. **ngAfterViewChecked:** This is called in response after Angular checks the component's views and child views.
viii. **ngOnDestroy:** This is the cleanup phase just before Angular destroys the directive/component.