========================
Development Procedure
========================

## Step-1) Create Angular Application

```
$ ng new ies-ui
```

Note: Select "Yes" for Routing because we have to route the requests to multiple components in this application.

## Step-2) Install Bootstrap and Jquery in our angular application

```
$ cd ies-ui
$ npm install bootstrap
$ npm install jquery
```

## Step-3) Configure below styles and scripts in angular.json file

```
"styles": [
"./node_modules/bootstrap/dist/css/bootstrap.css",
"src/styles.css"
],
"scripts": [
"node_modules/jquery/dist/jquery.min.js",
"node_modules/bootstrap/dist/js/bootstrap.min.js"
]
```

## Step- 4) Configure below styles in global src/styles.css file

```
@import "~bootstrap/dist/css/bootstrap.css";
/* You can add global styles to this file, and also import other style files */
.footer{
position: absolute;
bottom: 0;
width: 100%;
height: 70px;
background-color: blue;
text-align: center;
color: white;
}
```

## Step-5) Create Binding Class (account.ts) to map 'front end application' data with 'backend application data'.

```
$ ng generate class account
```

```
export class Account {

userAccId:number=0;
fullName:string="";
email:string="";
mobileNo:number=0;
gender:string="";
dob:string="";
ssn:number=0;
roleId:number=0;
activeSw:string="";

constructor(){}

}
```

Note: Make sure your backend api binding class and front end binding class variables are same.

## Step-6) Import HttpClientModule & FormsModule in app.module.ts file

## Step-7) Create Service class to write business logic

```
$ ng generate service account
```

```
import { HttpClient } from '@angular/common/http';
import { Injectable } from '@angular/core';

import { Observable } from 'rxjs';
import { Account } from './account';

@Injectable({
providedIn: 'root'
})
export class AccountService {

private ADMIN_API_URL="http://localhost:8080";

constructor(private httpClient : HttpClient) { }

createAccount(account:Account):Observable<Object>{
return this.httpClient.post(`${this.ADMIN_API_URL}/account`,
account,
{responseType:"text"}
);
}

getAccounts():Observable<Account[]>{
return this.httpClient.get<Account[]>(`${this.ADMIN_API_URL}/accounts`);
}

getAccountById(id:number):Observable<Account>{
return this.httpClient.get<Account>(`${this.ADMIN_API_URL}/account/${id}`);
}

updateAccountStatus(id:number, status:string):Observable<Object>{
return this.httpClient.put(`${this.ADMIN_API_URL}/account/${id}/${status}`,
null,
{responseType:"text"}
);
}

}
```

## Step-8) Create below Components

```
$ ng g c createaccount
$ ng g c accountlist
$ ng g c accountedit
```

## Step-9) Import FormsModule in app.module.ts file

## Step-10) Configure Routings in app-routing.module.ts file

```
const routes: Routes = [

{path:'accounts', component:AccountlistComponent},
{path: 'edit/:id', component:AccounteditComponent},
{path: 'create-account', component: CreateaccountComponent},
{path: ",redirectTo:'accounts'}

];
```

## Step-11) Add application logic in create account component file

```
import { Component } from '@angular/core';
import { Account } from '../account';
import { AccountService } from '../account.service';
import { Router } from '@angular/router';

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

account:Account = new Account();
msg:string= "";

constructor(private accService:AccountService, private router:Router){}

onSubmit(){
this.saveAccount();
}

saveAccount(){
this.accService.createAccount(this.account).subscribe(data => {
```

```
this.msg = data.toString();
console.log(data);
});
}
}
```

## Step-12) Add presentation logic in create component template file

```
<div class="col-md-6 offset-md-3">
<h2>Create User Account</h2>

<h4>{{msg}}</h4>

<form (ngSubmit)="onSubmit()">

<div class="form-group">
<label >Full Name</label>
<input type="text"
class="form-control"
name="name"
id="name"
[(ngModel)]="account.fullName"
>
</div>
<div class="form-group">
<label >Email</label>
<input type="text"
class="form-control"
name="email"
id="email"
[(ngModel)]="account.email"
>
</div>
<div class="form-group">
<label >Phno</label>
<input type="text"
class="form-control"
name="mobileNo"
id="mobileNo"
[(ngModel)]="account.mobileNo">
</div>
<div class="form-group">
<label >Gender</label>
<input type="text"
class="form-control"
name="gender"
id="gender"
[(ngModel)]="account.gender">
</div>
<div class="form-group">
<label >DOB</label>
<input type="text"
```

```
class="form-control"
name="dob"
id="dob"
[(ngModel)]="account.dob">
</div>

<div class="form-group">
<label >SSN</label>
<input type="text"
class="form-control"
name="ssn"
id="ssn"
[(ngModel)]="account.ssn">
</div>
<div class="form-group">
<label >Role</label>
<select name="demo" id="#" [(ngModel)]="account.roleId" class="form-control" >
<option value="1">Admin</option>
<option value="2">Case Worker</option>
</select>
</div>
<br/>
<div>
<button class="btn btn-primary" type="submit">
Submit
</button>
</div>
</form>
</div>
```

## Step-13) Add below code in account-list component ts file

```
import { Component, OnInit } from '@angular/core';
import { Account } from '../account';
import { AccountService } from '../account.service';
import { Router } from '@angular/router';

@Component({
selector: 'app-accountlist',
templateUrl: './accountlist.component.html',
styleUrls: ['./accountlist.component.css']
})
export class AccountlistComponent implements OnInit {

accounts:Account[] = [];

constructor(private accService:AccountService, private router:Router){}

ngOnInit(): void {
this.getAllAccounts();
}

getAllAccounts(){
```

```
this.accService.getAccounts().subscribe(data => {
this.accounts = data;
console.log(data);
});
}

editAccount(id:number){
this.router.navigate(['/edit', id]);
}

updateAccount(id:number, status:string){
this.accService.updateAccountStatus(id,status).subscribe(data => {
console.log(data);
})

this.getAllAccounts();

this.ngOnInit();
}

}
```

## Step-14) Add Presentation logic for account-list template file

```
<h2>View Accounts</h2>
<table class="table table-striped">
<thead>
<tr>
<th>Id</th>
<th>Name</th>
<th>Email</th>
<th>Phone</th>
<th>Gender</th>
<th>SSN</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr *ngFor="let acc of accounts">
<td>{{acc.userAccId}}</td>
<td>{{acc.fullName}}</td>
<td>{{acc.email}}</td>
<td>{{acc.mobileNo}}</td>
<td>{{acc.gender}}</td>
<td>{{acc.ssn}}</td>
<td>
<button class="btn btn-info" (click)="editAccount(acc.userAccId)">Edit</button>&nbsp;


<ng-container *ngIf="acc.activeSw === 'N';  then activate; else deactivate">
</ng-container>
```

```
<ng-template #activate>
<button class="btn btn-success" (click)="updateAccount(acc.userAccId, 'Y')">Activate</
button>&nbsp;
</ng-template>

<ng-template #deactivate>
<button class="btn btn-danger" (click)="updateAccount(acc.userAccId, 'N')">De-Activate</
button>
</ng-template>


</td>
</tr>
</tbody>
</table>
```

## Step-15) Add application logic in editaccount component ts file

```
import { Component } from '@angular/core';
import { Account } from '../account';
import { AccountService } from '../account.service';
import { ActivatedRoute, Router } from '@angular/router';

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

account:Account=new Account();
id:number=0;
msg:string="";


constructor(private accService:AccountService,
private router:Router,private activeRouter:ActivatedRoute) { }

ngOnInit(): void {
this.getAccount();
}
getAccount(){
this.id=this.activeRouter.snapshot.params['id'];
console.log("UPDATED ID ::"+this.id);
this.accService.getAccountById(this.id).subscribe(
data=>{
console.log("GETTING A CONTACT..");
console.log(data);
this.account=data;
},
error=>{
console.log("SOMETHING WENT WRONG DURING GETTING A CONTACT..");
console.log(error);
}
);
}
updateAccount(){
console.log("UPDATED ..");
this.accService.createAccount(this.account).subscribe(
data=>{
console.log("UPDATING A CONTACT..");
console.log(data);
this.msg = data.toString();
},
error=>{
console.log("SOMETHING WENT WRONG DURING UPDATING A CONTACT..");
console.log(error);
});
}
}
```

## Step-16) Add presentation logic in editaccount template file

```
<div class="col-md-6 offset-md-3">
<h2>Update User Account</h2>


<h4>{{msg}}</h4>

<form (ngSubmit)="updateAccount()">

<div class="form-group">
<label >Full Name</label>
<input type="text"
class="form-control"
name="name"
id="name"
[(ngModel)]="account.fullName"
>
</div>
<div class="form-group">
<label >Email</label>
<input type="text"
class="form-control"
name="email"
id="email"
[(ngModel)]="account.email"
>
</div>
<div class="form-group">
<label >Phno</label>
```

```
<input type="text"
class="form-control"
name="mobileNo"
id="mobileNo"
[(ngModel)]="account.mobileNo">
</div>
<div class="form-group">
<label >Gender</label>
<input type="text"
class="form-control"
name="gender"
id="gender"
[(ngModel)]="account.gender">
</div>
<div class="form-group">
<label >DOB</label>
<input type="text"
class="form-control"
name="dob"
id="dob"
[(ngModel)]="account.dob">
</div>

<div class="form-group">
<label >SSN</label>
<input type="text"
class="form-control"
name="ssn"
id="ssn"
[(ngModel)]="account.ssn">
</div>
<div class="form-group">
<label >Role</label>
<select name="demo" id="#" [(ngModel)]="account.roleId" class="form-control" >
<option value="1">Admin</option>
<option value="2">Case Worker</option>
</select>
</div>
<br>
<div>
<button class="btn btn-primary" type="submit">
Update
</button>
</div>
</form>
</div>
```

## Step-18) Configure Nav Bar & Router Outlet in app.component.html file

```
<!--########### NAV BAR STARTS HERE
##################################################### -->

<nav class="navbar navbar-expand-sm  bg-primary navbar-dark ">
```

```
<div class="container-fluid">
<a class="navbar-brand" href="#">Ashok IT</a>
<button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">
<span class="navbar-toggler-icon"></span>
</button>
<div class="collapse navbar-collapse" id="navbarSupportedContent">
<ul class="navbar-nav me-auto mb-2 mb-lg-0">


<li class="nav-item">
<a routerLink="accounts" routerLinkActive="active" class="nav-link" href="#">View Accounts</a>
</li>
<li class="nav-item">
<a routerLink="create-account" routerLinkActive="active" class="nav-link" href="#">Create Contact</a>
</li>
<li class="nav-item">
<a routerLink="report" routerLinkActive="active" class="nav-link" href="#">Report</a>
</li>
</ul>
<form class="d-flex">
<input class="form-control me-2" type="search" placeholder="Search" aria-label="Search">
<button class="btn btn-outline-success" type="submit">Search</button>
</form>
</div>
</div>
</nav>


<!-- #####################NAV BAR ENDS HERE
############################################### -->


<div class="container">
<router-outlet></router-outlet>
</div>
```

## Step-19) Execute Angular Application using 'ng serve' command.

```
$ ng serve
```

## Step-20) Open the browser and access application using below URL

URL : http://localhost:4200/

```
==============================
Report Module UI Development
==============================
```

1) Create Reports Component

```
$ ng g c reports
```

2) Configure route path for reports component in app.routing.module.ts   file

```
{path:'reports', component:ReportsComponent},
```

3) Create ReportSearchCriteria class

```
$ ng generate class report-search-criteria
```

```
export class ReportSearchCriteria {

planName:string="";
planStatus:string="";
gender:string="";

constructor(){

}
}
```

4) Create ReportResponse class

```
$ ng generate class report-response
```

```
export class ReportResponse {

cid:number=0;
fullName:string="";
email:string="";
mobileNo:number=0;
ssn:number=0;
planName:string="";
planStatus:string="";


constructor(){

}
}
```

5) Create Report service and write logic to make backend call using http client

```
$ ng generate service report
```

```
import { HttpClient } from '@angular/common/http';
import { Injectable } from '@angular/core';
import { ReportSearchCriteria } from './report-search-criteria';
import { Observable } from 'rxjs';
import { ReportResponse } from './report-response';

@Injectable({
providedIn: 'root'
})
export class ReportService {

private REPORT_API_URL="http://localhost:8081";

constructor(private httpClient:HttpClient) { }

getPlanNames() : Observable<any>{
return this.httpClient.get<any>(`${this.REPORT_API_URL}/plan-names`);
}

getPlanStatus(): Observable<any>{
return this.httpClient.get<any>(`${this.REPORT_API_URL}/plan-status`);
}

search(request : ReportSearchCriteria) : Observable<ReportResponse[]>{
return this.httpClient.post<ReportResponse[]>(`${this.REPORT_API_URL}/search`, request);
}

getExcel() {
return this.httpClient.get<any>(`${this.REPORT_API_URL}/excel`, {responseType : 'arraybuffer' as
'json'});
}

getPdf() {
return this.httpClient.get<any>(`${this.REPORT_API_URL}/pdf`, {responseType : 'arraybuffer' as
'json'});
}

}
```

6) Implement functions in component class

```
import { Component, OnInit } from '@angular/core';
import { ReportService } from '../report.service';
import { ReportSearchCriteria } from '../report-search-criteria';
import { ReportResponse } from '../report-response';

@Component({
selector: 'app-reports',
templateUrl: './reports.component.html',
styleUrls: ['./reports.component.css']
})
```

```
export class ReportsComponent implements OnInit{

constructor(private reportService : ReportService) { }

public planNames: string[] | undefined ;
public planStatuses : any;

public selectedPlan = "select";
public selectedStatus = "select";

searchRequest : ReportSearchCriteria = new ReportSearchCriteria();
searchResponse : ReportResponse[] = [];

ngOnInit(): void {
this.getPlanNames();
this.getPlanStatus();
}

getPlanNames(){
this.reportService.getPlanNames().subscribe(data => {
this.planNames = data;
});
}

getPlanStatus(){
this.reportService.getPlanStatus().subscribe(data => {
this.planStatuses = data;
});
}

search(){
this.searchRequest.planName = this.selectedPlan;
this.searchRequest.planStatus = this.selectedStatus;
this.reportService.search(this.searchRequest).subscribe(data => {
this.searchResponse = data;
});
}

onSubmit() {
this.search();
}

exportToExcel() {
this.reportService.getExcel().subscribe(data => {
let file = new Blob([data], { type: 'application/vnd.openxmlformats-
officedocument.spreadsheetml.sheet' });
var fileURL = URL.createObjectURL(file);
window.open(fileURL);
});
}

exportToPdf() {
this.reportService.getPdf().subscribe(data => {
let file = new Blob([data], { type: 'application/pdf' });
var fileURL = URL.createObjectURL(file);
window.open(fileURL);
});

}

}
```

7) design report page in template file

```
<div class="row">
<div class="col-8 offset-2">
<div class="card">
<div class="card-header text-center">
<h3>Insurance Report</h3>
</div>
<div class="card-body">
<form (ngSubmit)="onSubmit()">
<div class="row">
<div class="col-3">
<select class="form-select" name="planName" [(ngModel)]="selectedPlan">
<option selected Value="select" disabled>Select a Plan Name</option>
<option Value="">All Plans</option>
<option value="{{planName}}" *ngFor="let planName of planNames">{{planName}}</option>
</select>
</div>
<div class="col-3">
<select class="form-select" name="planStatus" [(ngModel)]="selectedStatus">
<option selected value="select" disabled>Select a Plan Status</option>
<option Value="">All Status</option>
<option value="{{planStatus}}" *ngFor="let planStatus of planStatuses">{{planStatus}}</option>
</select>
</div>
<div class="col-3">
<button type="submit" class="btn btn-primary"><i class="fa fa-search" aria-hidden="true"></i>
Search</button>
</div>
</div>
</form>
<div>
<div class="row">
<div class="col-10 offset-1">
<table class="table">
<thead>
<tr>
<th>ID</th>
<th>Plan Name</th>
<th>Plan Status</th>
<th>Holder Name</th>
<th>SSN</th>
</tr>
</thead>
```

```
<tbody>
<tr *ngFor="let response of searchResponse">
<td>{{response.cid}}</td>
<td>{{response.planName}}</td>
<td>{{response.planStatus}}</td>
<td>{{response.fullName}}</td>
<td>{{response.ssn}}</td>
</tr>
</tbody>
</table>
</div>

</div>

</div>
</div>
<div class="card-footer">
<div class="row">
<div class="col-4 offset-8">
<div class="row">
<div class="col-5 offset-2">
<!-- (click)="exportToExcel()" -->
<!-- <a target="_blank"> -->
<button class="btn btn-success" (click)="exportToExcel()">
Export <i class="fa fa-file-excel-o" aria-hidden="true"></i></button>
<!-- </a> -->
</div>
<div class="col-5">
<button class="btn btn-danger" (click)="exportToPdf()">Export <i class="fa fa-file-pdf-o" aria-
hidden="true"></i></button>
<!-- </a> -->
</div>
<!-- <a class="btn btn-success" (click)="exportToExcel()"> <i class="fa fa-plus-square"></i>
Excel</a> -->
</div>
</div>
</div>
</div>
</div>
</div>
</div>

=======================================================================================
===================================
```