```
============================================================
Create Fullstack application using Angular & Spring Boot
============================================================

#### Git Repo : https://github.com/ashokitschool/IES-UI-App.git

Frontend : Angular

Backend : Spring Boot

Database : MySQL

1) Login + Register

2) Create Case Worker

3) View Case Workers

4) Logout


================================
Angular Project Development
================================

################ Step-1 : Create Angular app ################

    $ ng new <app-name>

################ Step-2 : Install Bootstrap in Angular ################

    $ npm install bootstrap@5.3.2

################ Step-3 : Configure Bootstrap  "angular.json" file ###########

       "styles": [
            "./node_modules/bootstrap/dist/css/bootstrap.min.css"
        ],

################  Step-4 : start our angular app ################


################   Step-5 : Create Required Components ################


      $ ng g c login

      $ ng g c register

      $ ng g c side-menu

      $ ng g c dashboard

      $ ng g c create-cw

      $ ng g c view-cw


######################    Step-6 : Create app service   ################

    $ ng generate service app
```

```
################  Step-7 : Configure Routings ################

    {path:'', component: LoginComponent,pathMatch:'full'},
    {path: 'login', component: LoginComponent},
    {path: 'register', component: RegisterComponent},
    {path: 'dashboard', component: DashboardComponent},
    {path: 'create-cw', component: CreateCwComponent},
    {path: 'view-cw', component: ViewCwComponent}


####################    Step-8: Prepare Side Menu   ####################

-----------------side-menu-css-----------------

body {
    background-color: #fbfbfb;
    }
    @media (min-width: 991.98px) {
    main {
    padding-left: 240px;
    }
    }

    /* Sidebar */
    .sidebar {
    position: fixed;
    top: 0;
    bottom: 0;
    left: 0;
    padding: 58px 0 0; /* Height of navbar */
    box-shadow: 0 2px 5px 0 rgb(0 0 0 / 5%), 0 2px 10px 0 rgb(0 0 0 / 5%);
    width: 240px;
    z-index: 600;
    }

    @media (max-width: 991.98px) {
    .sidebar {
    width: 100%;
    }
    }
    .sidebar .active {
    border-radius: 5px;
    box-shadow: 0 2px 5px 0 rgb(0 0 0 / 16%), 0 2px 10px 0 rgb(0 0 0 / 12%);
    }

    .sidebar-sticky {
    position: relative;
    top: 0;
    height: calc(100vh - 48px);
    padding-top: 0.5rem;
    overflow-x: hidden;
    overflow-y: auto; /* Scrollable contents if viewport is shorter than content. */
    }

-------------------------------side-menu-html-------------------------------

<!--Main Navigation-->
<header>
    <!-- Sidebar -->
    <nav id="sidebarMenu" class="collapse d-lg-block sidebar collapse bg-white">
      <div class="position-sticky">
        <div class="list-group list-group-flush mx-3 mt-4">

          <!-- Collapsed content -->
          <ul id="collapseExample1" class="collapse show list-group list-group-flush">
```

```
            <li class="list-group-item py-1">
              <a routerLink="create-cw">CreateCw</a>
            </li>
            <li class="list-group-item py-1">
              <a routerLink="view-cw">ViewCw</a>
            </li>
            <li class="list-group-item py-1">
              <button class="btn btn-primary" (click)="logout()">Logout</button>
              </li>
          </ul>
        </div>
      </div>
    </nav>
    <!-- Sidebar -->

    <!-- Navbar -->
    <nav id="main-navbar" class="navbar navbar-expand-lg navbar-light bg-white fixed-top">
      <!-- Container wrapper -->
      <div class="container-fluid">
        <!-- Toggle button -->
        <button class="navbar-toggler" type="button" data-mdb-toggle="collapse" data-mdb-
target="#sidebarMenu"
          aria-controls="sidebarMenu" aria-expanded="false" aria-label="Toggle navigation">
          <i class="fas fa-bars"></i>
        </button>

        <!-- Brand -->
        <a class="navbar-brand" href="#">
          <img src="../../assets/images/ashokit.png"  height="65" alt="Ashok IT Logo"
            loading="lazy" />
        </a>
        <!-- Search form -->
        <form class="d-none d-md-flex input-group w-auto my-auto">
          <input autocomplete="off" type="search" class="form-control rounded"
            placeholder='Search (ctrl + "/" to focus)' style="min-width: 225px;" />
          <span class="input-group-text border-0"><i class="fas fa-search"></i></span>
        </form>

        <!-- Right links -->
        <ul class="navbar-nav ms-auto d-flex flex-row">
          <!-- Notification dropdown -->
        </ul>
      </div>
      <!-- Container wrapper -->
    </nav>
    <!-- Navbar -->
  </header>
  <!--Main Navigation-->

  <!--Main layout-->
  <main style="margin-top: 58px;">
    <div class="container pt-4"></div>
  </main>
  <!--Main layout-->

  ------------------------------------side-menu-component---------------------------------

  export class SideMenuComponent {

    constructor( private router: Router){

    }

    logout(){
      localStorage.clear();
```

```
      this.router.navigate(['/']);
    }

  }

  ################    Step-9 : app component changes  ################

  export class AppComponent implements DoCheck{
    title = 'app';

    isLoggedIn:Boolean=false;

    constructor(private route:Router,private appserv:AppService){
    }


    ngDoCheck(){
        this.isLoggedIn = this.appserv.validateLogin();
    }

  }

  ################################## Step-10 : app.service.ts file changes  ################

  export class AppService {

    constructor() { }

    validateLogin():Boolean{
      return Boolean(localStorage.getItem("isLoggedIn"));
    }
  }


  #############  Step-11: Invoke side-menu and configure router-outlet in app component #########

  <div class="container">
  <div class="side-nav" *ngIf="isLoggedIn">
  <app-side-menu ></app-side-menu>
  </div>
  <div class="main-content">
   <router-outlet></router-outlet>
   </div>
  </div>


  ################   Step-12 : Login Component Changes ################

  ------------------------------ form binding --------------------

  export class LoginForm {

      email:string ;
      pwd: string;

      constructor(a:string, b:string){
          this.email = a;
          this.pwd = b;
      }
  }

  export class LoginResponse {

     userid:number;
     name:string;
```

```
    validLogin:boolean;

    constructor(a:number, b:string, c:boolean){
     this.userid=a;
     this.name=b;
     this.validLogin=c;
    }
}

----------------------------service changes ----------------------------

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

  private baseUrl='http://localhost:8080';

  constructor(private httpClient: HttpClient) { }

  loginCheck(loginForm:LoginForm):Observable<LoginResponse>{
    return this.httpClient.post<LoginResponse>(`${this.baseUrl}/login`,loginForm,
{responseType:"json"});
  }

}

-----------------------------------------------------------------

export const appConfig: ApplicationConfig = {
  providers: [provideRouter(routes), provideHttpClient()]
};

---------------------- login component -------------------------
export class LoginComponent {

  isLoggedIn:boolean;
  login:LoginForm = new LoginForm("","");
  loginResp:LoginResponse| undefined;
  errMsg:string="";

  constructor(private appSvc:AppService, private router: Router){
    this.isLoggedIn = false;
  }

  loginCheck(){
    this.appSvc.loginCheck(this.login).subscribe(data => {
      this.loginResp = data;
      if(this.loginResp.validLogin){
        localStorage.setItem("isLoggedIn","true");
        this.router.navigate(["dashboard"]);
      }else{
        this.errMsg="Invalid Credentials";
      }
    });
  }

}
--------------------------------- login template ----------------------------------

<div class="errMsg">
    {{errMsg}}
</div>

<form  (submit)="loginCheck()">
```

```
      <table>
          <tr>
              <td></td>
              <td><h1>Login Here</h1></td>
          </tr>
          <tr>
              <td>Enter Email : </td>
              <td><input type="email" [(ngModel)]="login.email" name="email"/></td>
          </tr>
          <tr>
              <td>Enter Pwd : </td>
              <td><input type="password" [(ngModel)]="login.pwd" name="pwd"/></td>
          </tr>
          <tr>
              <td> </td>
              <td><input type="submit" value="Login" class="btn btn-primary"/></td>
          </tr>
      </table>

  </form> <a routerLink="register">Register Here</a>

  ################   Step-14 : Registration Component ################


  ----------------- register form binding -----------------------------

  export class RegisterForm {

      name:string;
      email:string;
      pwd:string;
      phno:string;

      constructor(a:string, b:string, c:string, d:string){
          this.name=a;
          this.email=b;
          this.pwd=c;
          this.phno=d;
      }
  }

  export class RegResponse {

      successMsg:string;
      errorMsg:string;

      constructor(a:string, b:string){
          this.successMsg = a;
          this.errorMsg=b;
      }
  }
  ------------------------- service changes -------------------------------------

    register(reg:RegForm):Observable<RegResponse>{
      return this.httpClient.post<RegResponse>(`${this.baseUrl}/register`,reg,{responseType:"json"});
    }


  ---------------------- register component ----------------------------------

  export class RegisterComponent {

    register:RegForm = new RegForm("","","","");

    regResp:RegResponse | undefined;
```

```
  constructor(private appSvc:AppService){

  }

  registration(){
    this.appSvc.register(this.register)
               .subscribe(data => {
                  this.regResp = data;
                  console.log(this.regResp);
               });
  }

}
------------------------------ register template ---------------------

<div class="succMsg">
    {{regResp?.successMsg}}
</div>

<div class="errMsg">
    {{regResp?.errorMsg}}
</div>


<form (submit)="registration()">
    <table>
        <tr>
            <td></td>
            <td></td>
        </tr>
        <tr>
            <td></td>
            <td><h1>Create Account Here</h1></td>
        </tr>
        <tr>
            <td>Enter Name : </td>
            <td><input type="text"  [(ngModel)]="register.name" name="name"/></td>
        </tr>
        <tr>
            <td>Enter Email : </td>
            <td><input type="email"  [(ngModel)]="register.email" name="email"/></td>
        </tr>
        <tr>
            <td>Enter Pwd : </td>
            <td><input type="pwd"  [(ngModel)]="register.pwd" name="pwd"/></td>
        </tr>
        <tr>
            <td>Enter Phno : </td>
            <td><input type="number" [(ngModel)]="register.phno" name="phno"/></td>
        </tr>
        <tr>
            <td> </td>
            <td><input type="submit" value="Register" class="btn btn-primary" /></td>
        </tr>
    </table>

</form>
    <div>
        Do you account ? <a routerLink="/">Login Here</a>
    </div>

--------------------------------------------------------------------------------
```