```
Project: Build an Angular app where the admin can view and manage employees.
1:Creating the app using the command "ng new Employee
______
Step 2:Editing index.html file.
<!doctype html>
<html lang="en">
<head>
 <meta charset="utf-8">
 <title>EventManagement</title>
 <base href="/">
 <meta name="viewport" content="width=device-width, initial-scale=1">
 <link rel="icon" type="image/x-icon" href="favicon.ico">
 <link rel="stylesheet"</pre>
href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.2/dist/css/bootstrap.min.css"
integrity="sha384-
x0olHFLEh07PJGoPkLv1IbcEPTNtaed2xpHsD9ESMhqIYd0nLMwNLD69Npy4HI+N"
crossorigin="anonymous">
</head>
<body>
 <app-root></app-root>
 <script
src="https://cdn.jsdelivr.net/npm/jquery@3.5.1/dist/jquery.slim.min.js"
integrity="sha384-DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRk
fj" crossorigin="anonymous"></script>
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@4.6.2/dist/js/bootstrap.bundle.min.j
s" integrity="sha384-
Fy6S3B9q64WdZWQUiU+q4/2Lc9npb8tCaSX9FK7E8HnRr0Jz8D60P9d05Vg3Q9ct"
crossorigin="anonymous"></script>
<script src="https://unpkg.com/ionicons@4.5.10-0/dist/ionicons.js"></script>
</body>
</html>
3:Creating User model Class
export class UserClass{
   id:number=0;
   fullname:string="";
   mobile:string="";
   email:string="";
   password:string=""
       -----Sten
4:Editing the app.module.ts file which is auto generated while creating the app.
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';
import { LoginComponent } from './login/login.component';
import { CrudOperationComponent } from './crud-operation/crud-
operation.component';
import {    FormsModule } from '@angular/forms';
import {    ReactiveFormsModule } from '@angular/forms';
```

```
import {HttpClientModule} from '@angular/common/http';
@NgModule({
  declarations: [
    AppComponent,
    LoginComponent,
    CrudOperationComponent,
  imports: [
    BrowserModule,
    AppRoutingModule,
    FormsModule,
    HttpClientModule,
    ReactiveFormsModule
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }
Step 5:Editing app-routing.ts file
import { NgModule } from '@angular/core';
import { RouterModule, Routes } from '@angular/router';
import { CrudOperationComponent } from './crud-operation/crud-
operation.component';
import { LoginComponent } from './login/login.component';
const routes: Routes = [
  {path: '', redirectTo: 'login', pathMatch: 'full'},
  {path:'login',component:LoginComponent},
{path:'ManageUser',component:CrudOperationComponent},
@NgModule({
  imports: [RouterModule.forRoot(routes)],
  exports: [RouterModule]
})
export class AppRoutingModule { }
Step 6:Editing app.component.ts file
<router-outlet></router-outlet>
Step 7:Adding login component in the application
1.)login.component.ts
import { HttpClient } from '@angular/common/http';
import { Component, OnInit } from '@angular/core';
import {FormGroup,FormBuilder} from '@angular/forms'
import { Route, Router } from '@angular/router';
@Component({
   selector: 'app-login',
  templateUrl: './login.component.html',
  styleUrls: ['./login.component.css']
})
```

```
export class LoginComponent implements OnInit {
  public loginForm !: FormGroup
  constructor(private formBuilder:FormBuilder,private
HttpClient:HttpClient,private route:Router) { }
  ngOnInit(): void {
    this.loginForm=this.formBuilder.group({
      email:[''],
      password:['']
    })
  }
  login(){
    this.HttpClient.get<any>("http://localhost:3000/users")
    .subscribe(res=>{
      const user=res.find((a:any)=>{
        return a.email===this.loginForm.value.email &&
a.password===this.loginForm.value.password
      });
      if(user){
        alert("Login successful");
        this.loginForm.reset();
        this.route.navigate(["ManageUser"])
      }
      else{
        alert("Incorrect Credentials")
   })
  }
}
2.)login.component.html
<div class="container">
    <div class="row">
        <div class="col-md-6">
            <div class="card">
                <div class="text-center">
                    <h1>Login</h1>
                    <h6>Please enter Email & password !!</h6>
                <form [formGroup]="loginForm" (ngSubmit)="login()">
                    <div class="form-group">
                      <label for="exampleInputEmail1">Email address</label>
                      <input formControlName="email" type="email" class="form-</pre>
control" id="exampleInputEmail1" aria-describedby="emailHelp" placeholder="Enter
email">
                      <small id="emailHelp" class="form-text text-muted ">We'll
never share your email with anyone else.</small>
                    </div>
                    <div class="form-group">
                      <label for="exampleInputPassword1">Password</label>
                      <input formControlName="password" type="password"</pre>
class="form-control" id="exampleInputPassword1" placeholder="Password">
                    <button type="submit" class="btn btn-primary</pre>
">Login</button>
                  </form>
```

```
</div>
    </div>
</div>
3.)login.component.css
.card{
   margin-top: 10px;
   border: none;
   width: 500px;
   padding: 40px;
   position: absolute;
   top: 50%;
    left: 55%;
   background:linear-gradient(90deg, rgb(107, 107, 231) 0%,rgb(36, 36, 97)
100%);
   color:white;
   font-size: 16px;
}
a{
   margin-top: 20px;
   color: rgb(237, 69, 237);
   font-size: 20px;
   font-weight: 500;
}
h1{
    font-size: 32px;
   font-weight: 500;
   color: white;
.btn{
   align-items: center;
   text-align: center;
   margin-left: 25%;
   width: 200px;
   font-weight: 500;
   background-color: rgb(7, 43, 249);
   border: none;
   color: white;
}
Step 8:Adding crud-operation component
1.)crud-operation.component.ts
import { Component, OnInit } from '@angular/core';
import { ListService } from '../list.service';
import { UserClass } from '../UserClass';
import {FormGroup,FormBuilder} from '@angular/forms'
import { HttpClient } from '@angular/common/http';
import { Router } from '@angular/router';
@Component({
  selector: 'app-crud-operation',
  templateUrl: './crud-operation.component.html',
  styleUrls: ['./crud-operation.component.css']
export class CrudOperationComponent implements OnInit {
```

```
constructor(private service:ListService,private
formBuilder:FormBuilder ,private http:HttpClient,private router:Router) { }
  public signupForm!:FormGroup
  users:UserClass[];
  usersObj:UserClass=new UserClass()
  btnUpdateShow:boolean = false;
  btnSaveShow:boolean = true;
    ngOnInit(): void {
      this.service.getAllUsers().subscribe(result=>this.users=result);
      this.signupForm=this.formBuilder.group({
        fullname:[''],
        email:[''],
password:[''],
        mobile:['<sup>1</sup>]
      })
    AddUser(){
      this.SaveShowBtn();
      this.http.post<any>("http://localhost:3000/users",this.signupForm.value)
      .subscribe(res=>{
        alert("User added successfully");
        this.signupForm.reset();
        this.router.navigate(['ManageUser'])
        window.location.reload()
      },err=>{
        alert("Something went wrong")
      })
DeleteUser(row :any){
  this.service.DeleteUserDetails(row.id).subscribe(res=>{
    alert("User Deleted")
    window.location.reload();
  })
}
UpdateUser(row :any){
this.signupForm.controls['fullname'].setValue(row.value);
this.signupForm.controls['email'].setValue(row.value);
this.signupForm.controls['password'].setValue(row.value);
this.signupForm.controls['mobile'].setValue(row.value);
this.usersObj.id=row.id
this.UpdateShowBtn();
UpdateUserDetails(){
  this.usersObj.fullname=this.signupForm.value.fullname
  this.usersObj.email=this.signupForm.value.email
  this.usersObj.password=this.signupForm.value.password
  this.usersObj.mobile=this.signupForm.value.mobile
this.service.UpdateUserDetailing(this.usersObj,this.usersObj.id).subscribe(res=>
  alert("Data updated successfully")
```

```
window.location.reload();
})
UpdateShowBtn()
 {
   this.btnUpdateShow = true;
   this.btnSaveShow = false;
 SaveShowBtn()
   this.btnUpdateShow = false;
   this.btnSaveShow = true;
 }
}
2)crud-operation.component.html
<div class="container">
   <div class="row">
   <h2 style="color: white;">Event Management</h2>
   <button type="button" class="btn btn-primary" data-toggle="modal" data-</pre>
target="#exampleModal"> Add User</button>
</div>
   <thead>
   Id
   Name
   UserName
   Email
   Action
   </thead>
   {{u.id}}
   {{u.fullname}}
   {{u.mobile}}
   {u.email}}
   <button class="btn btn-success me-2" type="button"</pre>
(click)="UpdateUser(u)" data-toggle="modal"
data-target="#exampleModal">Edit</button>
```

```
<button class="btn btn-danger" (click)="DeleteUser(u)">Delete</button>
    <!-- Modal -->
      <div class="modal fade" id="exampleModal" tabindex="-1" role="dialog"</pre>
aria-labelledby="exampleModalLabel" aria-hidden="true">
        <div class="modal-dialog" role="document">
          <div class="modal-content">
            <div class="modal-header">
              <h5 class="modal-title" id="exampleModalLabel">User Details</h5>
              <button type="button" class="close" data-dismiss="modal" aria-</pre>
label="Close">
                <span aria-hidden="true">&times;</span>
              </button>
            </div>
            <div class="modal-body">
                <form [formGroup]="signupForm" (ngSubmit)="AddUser()">
                    <div class="form-group">
                      <label for="exampleInputEmail1">Full Name</label>
                      <input formControlName="fullname" type="text" class="form-</pre>
control" id="exampleInputEmail1" aria-describedby="emailHelp" placeholder="Enter
vour name">
                         </div>
                    <div class="form-group">
                      <label for="exampleInputPassword1">Mobile Number</label>
                      <input formControlName="mobile" type="text" class="form-</pre>
control" id="exampleInputPassword1" placeholder="Enter phone number">
                    </div>
                    <div class="form-group">
                        <label for="exampleInputEmail1">Email</label>
                        <input formControlName="email" type="email" class="form-</pre>
control" id="exampleInputEmail1" aria-describedby="emailHelp" placeholder="Enter
email">
                           </div>
                           <div class="form-group">
                            <label for="exampleInputEmail1">Password</label>
                            <input formControlName="password" type="password"</pre>
class="form-control" id="exampleInputEmail1" aria-describedby="emailHelp"
placeholder="Enter password">
                               </div>
        </form>
    </div>
 <div class="modal-footer">
              <button type="button" class="btn btn-secondary" data-</pre>
dismiss="modal">Close</button>
              <button *ngIf="btnSaveShow" type="submit" id="add" class="btn btn-</pre>
primary" >Add User</button>
              <button *ngIf="btnUpdateShow" (click)="UpdateUserDetails()"</pre>
class="btn btn-primary" >Update</button>
            </div>
          </div>
        </div>
      </div>
```

```
</div>
```

```
3)crud-operation.component.css
..........
background: linear-gradient(90deg, blue 0%, rgb(33, 33, 87) 100%);
display: flex;
justify-content: space-between;
margin: 3px;
}
.row button{
   width: 200px;
   height: 3rem;
   margin: 30px;
   background-color: green;
   color: white;
   padding:2px;
border: none;
transition: 1s;
}
.row button:hover{
   background-color: rgb(54, 90, 54);
}
.btn{
   margin: 10px;
.container{
   margin-top: 30px;
.row h2{
   margin: 20px;
```