

Food Recipe Project SOURCE CODE

Admin Login

<> adminlogin.component.html

```
<link
href="//maxcdn.bootstrapcdn.com/bootstrap/4.1.1/css/bootstrap.min.css"
rel="stylesheet" id="bootstrap-css">
  <script
src="//maxcdn.bootstrapcdn.com/bootstrap/4.1.1/js/bootstrap.min.js"></scri
pt>
  <script
src="//cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script
>
  <!------- Include the above in your HEAD tag ----->

<body>
  <div id="login">
    <h3 class="text-center text-white pt-5">Login form</h3>
    <div class="container">
      <div id="login-row" class="row justify-content-center align-
items-center">
        <div id="login-column" class="col-md-6">
          <div id="login-box" class="col-md-12">

            <h3 class="text-center text-info">Admin
Login</h3>

            <div class="form-group">
              <label for="username" class="text-
info">Username:</label><br>
              <input type="text" name="username"
id="username" [(ngModel)]="username" class="form-control">
            </div>
            <div class="form-group">
              <label for="password" class="text-
info">Password:</label><br>
              <input type="password" name="password"
id="password" [(ngModel)]="password" class="form-control">
            </div>
            <div class="form-group">
              <button (click)=checkLogin() class="btn
btn-info btn-md">
                Login
              </button>
            </div>

          </div>
        </div>
      </div>
    </div>
  </div>
```

```

        </div>
    </div>
</body>
=====

TS adminlogin.component.spec.ts
=====

import { ComponentFixture, TestBed } from '@angular/core/testing';

import { AdminloginComponent } from './adminlogin.component';

describe('AdminloginComponent', () => {
    let component: AdminloginComponent;
    let fixture: ComponentFixture<AdminloginComponent>;

    beforeEach(async () => {
        await TestBed.configureTestingModule({
            declarations: [ AdminloginComponent ]
        })
        .compileComponents();
    });

    beforeEach(() => {
        fixture = TestBed.createComponent(AdminloginComponent);
        component = fixture.componentInstance;
        fixture.detectChanges();
    });

    it('should create', () => {
        expect(component).toBeTruthy();
    });
});
=====

TS adminlogin.component.ts
=====

import { Component, OnInit } from '@angular/core';
import { Router } from '@angular/router';
import { AdminService } from '../admin.service';

@Component({
    selector: 'app-adminlogin',
    templateUrl: './adminlogin.component.html',
    styleUrls: ['./adminlogin.component.css']
})
export class AdminloginComponent implements OnInit {
    username='';
    password='';
    invalidLogin=false;
    constructor(private router:Router,private adminService:AdminService) { }

```

```

ngOnInit(): void {
}
checkLogin(){
  if(this.adminService.authenticate(this.username,this.password)){
this.router.navigate([''])
this.invalidLogin=false;
  }else{
    this.invalidLogin=true;
  }
}
logout(){
  sessionStorage.removeItem('username');
}
}

```

=====

Admin Logout

=====

<> adminlogout.component.html

=====

<p>adminlogout works!</p>

=====

TS adminlogout.component.spec.ts

=====

```
import { ComponentFixture, TestBed } from '@angular/core/testing';
```

```
import { AdminlogoutComponent } from './adminlogout.component';
```

```
describe('AdminlogoutComponent', () => {
  let component: AdminlogoutComponent;
  let fixture: ComponentFixture<AdminlogoutComponent>;
```

```
  beforeEach(async () => {
    await TestBed.configureTestingModule({
      declarations: [ AdminlogoutComponent ]
    })
    .compileComponents();
  });
```

```
  beforeEach(() => {
    fixture = TestBed.createComponent(AdminlogoutComponent);
    component = fixture.componentInstance;
    fixture.detectChanges();
```

```

    });

    it('should create', () => {
      expect(component).toBeTruthy();
    });
  });
});

```

=====

TS adminlogout.component.ts

=====

```

import { Component, OnInit } from '@angular/core';
import { Router } from '@angular/router';
import { AdminService } from '../admin.service';

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

  constructor(private router:Router,private adminService:AdminService) { }

  ngOnInit(): void {
    this.adminService.logout();
    this.router.navigate(['adminlogin']);
  }

}

```

=====

User Login

=====

<> userlogin.component.html

=====

```

<link
href="//maxcdn.bootstrapcdn.com/bootstrap/4.1.1/css/bootstrap.min.css"
rel="stylesheet" id="bootstrap-css">
<script
src="//maxcdn.bootstrapcdn.com/bootstrap/4.1.1/js/bootstrap.min.js"></scri
pt>

```

```

<script
src="//cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script
>
<!------- Include the above in your HEAD tag ----->

<body>
  <div id="login">
    <h3 class="text-center text-white pt-5">Login form</h3>
    <div class="container">
      <div id="login-row" class="row justify-content-center align-
items-center">
        <div id="login-column" class="col-md-6">
          <div id="login-box" class="col-md-12">
            <form (ngSubmit)="checkUserLogin()">
              <h3 class="text-center text-info">Login</h3>
              <div class="form-group">
                <label for="username" class="text-
info">Username:</label><br>
                <input type="text" name="username"
id="username" [(ngModel)]="username" class="form-control">
              </div>
              <div class="form-group">
                <label for="password" class="text-
info">Password:</label><br>
                <input type="text" name="password"
id="password" [(ngModel)]="password" class="form-control">
              </div>
              <div class="form-group">
                <button type="submit" class="btn btn-info
btn-md">Login</button>
              </div>
            </form>
          </div>
        </div>
      </div>
    </div>
  </div>
</body>

```

=====

TS userlogin.component.spec.ts

=====

```

import { ComponentFixture, TestBed } from '@angular/core/testing';

import { UserloginComponent } from './userlogin.component';

describe('UserloginComponent', () => {
  let component: UserloginComponent;
  let fixture: ComponentFixture<UserloginComponent>;

```

```

beforeEach(async () => {
  await TestBed.configureTestingModule({
    declarations: [ UserloginComponent ]
  })
  .compileComponents();
});

beforeEach(() => {
  fixture = TestBed.createComponent(UserloginComponent);
  component = fixture.componentInstance;
  fixture.detectChanges();
});

it('should create', () => {
  expect(component).toBeTruthy();
});
});

```

=====

TS userlogin.component.ts

=====

```

import { Component, OnInit } from '@angular/core';
import { Router } from '@angular/router';
import { UserService } from '../user.service';
import { Users } from '../usermodel';

@Component({
  selector: 'app-userlogin',
  templateUrl: './userlogin.component.html',
  styleUrls: ['./userlogin.component.css']
})
export class UserloginComponent implements OnInit {
  username:string='';
  password:string='';
  invalidLogin = false
  user?:Users;
  constructor(private router:Router,private userService:UserService) { }

  ngOnInit(): void {
  }
  checkUserLogin(){
    if(this.userService.authenticate(this.username,this.password)){
      this.router.navigate(['']);
      this.invalidLogin=false;
      alert("login Successful");
    }else{
      this.invalidLogin=true;
    }
  }
}

```

```
}
```

```
=====
```

User Logout

```
=====
```

```
<> userlogout.component.html
```

```
=====
```

```
<p>userlogout works!</p>
```

```
=====
```

TS userlogout.component.spec.ts

```
=====
```

```
import { ComponentFixture, TestBed } from '@angular/core/testing';
```

```
import { UserlogoutComponent } from './userlogout.component';
```

```
describe('UserlogoutComponent', () => {
```

```
  let component: UserlogoutComponent;
```

```
  let fixture: ComponentFixture<UserlogoutComponent>;
```

```
  beforeEach(async () => {
```

```
    await TestBed.configureTestingModule({
```

```
      declarations: [ UserlogoutComponent ]
```

```
    })
```

```
    .compileComponents();
```

```
  });
```

```
  beforeEach(() => {
```

```
    fixture = TestBed.createComponent(UserlogoutComponent);
```

```
    component = fixture.componentInstance;
```

```
    fixture.detectChanges();
```

```
  });
```

```
  it('should create', () => {
```

```
    expect(component).toBeTruthy();
```

```
  });
```

```
});
```

```
=====
```

TS userlogout.component.ts

```
=====
```

```

import { Component, OnInit } from '@angular/core';
import { Router } from '@angular/router';
import { UserService } from '../user.service';

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

  constructor(private router:Router,private userService:UserService) { }

  ngOnInit(): void {
    this.userService.logout();
    this.router.navigate(['userlogin']);
  }

}

```

=====

User Register

=====

```

<html>
<link
href="//maxcdn.bootstrapcdn.com/bootstrap/4.1.1/css/bootstrap.min.css"
rel="stylesheet" id="bootstrap-css">
<script
src="//maxcdn.bootstrapcdn.com/bootstrap/4.1.1/js/bootstrap.min.js"></scri
pt>
<script
src="//cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script
>

<body>
  <div id="login">
    <h3 class="text-center text-white pt-5">Register form</h3>
    <div class="container">
      <div id="login-row" class="row justify-content-center align-
items-center">
        <div id="login-column" class="col-md-6">
          <div id="login-box" class="col-md-12">
            <div [hidden]="submitted" style="width: 400px;">
              <form (ngSubmit)="registerData()" method="post">
                <h3 class="text-center text-info">Register</h3>
                <div class="form-group">
                  <label for="username" class="text-
info">Name:</label><br>

```



```

        <input type="text" name="name" id="name"
[(ngModel)]="user.name" class="form-control">
    </div>
    <div class="form-group">
        <label for="password" class="text-
info">Email:</label><br>
        <input type="email" name="password"
id="password" [(ngModel)]="user.email" class="form-control">
    </div>
    <div class="form-group">
        <label for="mobile" class="text-
info">Email:</label><br>
        <input type="number" name="mobile" id="password"
[(ngModel)]="user.mobile" class="form-control">
    </div>
    <div class="form-group">
        <label for="username" class="text-
info">Username:</label><br>
        <input type="text" name="username"
id="username" [(ngModel)]="user.username" class="form-control">
    </div>
    <div class="form-group">
        <label for="password" class="text-
info">Password:</label><br>
        <input type="password" name="password"
id="password" [(ngModel)]="user.password" class="form-control">
    </div>
    <div class="form-group">
        <button type="submit" class="btn btn-info
btn-md">Register</button>
    </div>
</form>

</div>
</div>
</div>
</div>
</div>
<div [hidden]="!submitted">
    <h4>You submitted successfully!</h4>
    <!-- <button class="btn btn-success"
(click)="newEmployee()">Add</button> -->
</div>
</body>
</html>
=====

```

TS userregister.component.spec.ts

=====

```
import { ComponentFixture, TestBed } from '@angular/core/testing';
```

```

import { UserregisterComponent } from './userregister.component';

describe('UserregisterComponent', () => {
  let component: UserregisterComponent;
  let fixture: ComponentFixture<UserregisterComponent>;

  beforeEach(async () => {
    await TestBed.configureTestingModule({
      declarations: [ UserregisterComponent ]
    })
    .compileComponents();
  });

  beforeEach(() => {
    fixture = TestBed.createComponent(UserregisterComponent);
    component = fixture.componentInstance;
    fixture.detectChanges();
  });

  it('should create', () => {
    expect(component).toBeTruthy();
  });
});

```

=====

TS userregister.component.ts

=====

```

import { Component, OnInit } from '@angular/core';
import { Router } from '@angular/router';
import { UserService } from '../user.service';
import { Users } from '../usermodel';

@Component({
  selector: 'app-userregister',
  templateUrl: './userregister.component.html',
  styleUrls: ['./userregister.component.css']
})
export class UserregisterComponent implements OnInit {
  user:Users=new Users();
  submitted=false;
  constructor(private router:Router,private userService:UserService) { }

  ngOnInit(): void {
  }
  addItemSave() {
    this.userService.userRegister(this.user).subscribe();
  }
}

```

```

    registerData() {
        this.submitted=true;
        this.addItemSave();
    }
}

```

=====

Search Receipe

=====

<> Searchrecipe.component.html

=====

```

<link
href="//maxcdn.bootstrapcdn.com/bootstrap/4.1.1/css/bootstrap.min.css"
rel="stylesheet" id="bootstrap-css">
  <script
src="//maxcdn.bootstrapcdn.com/bootstrap/4.1.1/js/bootstrap.min.js"></scri
pt>
  <script
src="//cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script
>
  <!-- Include the above in your HEAD tag -->

  <body>
    <div id="login">
      <h3 class="text-center text-white pt-5">Search form</h3>
      <div class="container">
        <div id="login-row" class="row justify-content-center
align-items-center">
          <div id="login-column" class="col-md-6">
            <div id="login-box" class="col-md-12">

              <h3 class="text-center text-info">Serach
Recipe</h3>

              <div class="form-group">
                <label for="username" class="text-
info">Recipe Name:</label><br>
                <input type="text" name="username"
id="itemname" [(ngModel)]="itemname" class="form-control">
              </div>

              <div class="form-group">
                <button (click)=onRecipeSearch()
class="btn btn-info btn-md">
                  Search
                </button>
              </div>

            </div>
          </div>
        </div>
      </div>
    </div>
  </body>
</div>

```

```

        </div>
    </div>
</div>

<div class="panel panel-primary">
  <div class="panel-heading">
    <h2>Search Result</h2>
  </div>
  <div class="panel-body">
    <table class="table table-striped">
      <thead>
        <tr>
          <th>Recipe Id</th>
          <th>Recipe Name</th>
          <th>Recipe Ingredients</th>
          <th>Recipe CookingSteps</th>

          </tr>
        </thead>
        <tbody *ngIf="recipe!=null">

          <tr *ngFor="let item of recipe">

            <td>{{item.id}}</td>
            <td>{{item.itemname}}</td>
            <td>{{item.ingredients}}</td>
            <td>{{item.cookingsteps}}</td>

            <!-- <td><button (click)="reserveTable(item.id)" class="btn btn-
success">Reserve</button></td>-->
          </tr>

        </tbody>
      </table>

    </div>
  </div>
</div>
</body>

```

=====

TS searchrecipe.component.spec.ts

=====

```

import { ComponentFixture, TestBed } from '@angular/core/testing';
import { SearchrecipeComponent } from './searchrecipe.component';
describe('SearchrecipeComponent', () => {

```

```

let component: SearchrecipeComponent;
let fixture: ComponentFixture<SearchrecipeComponent>;

beforeEach(async () => {
  await TestBed.configureTestingModule({
    declarations: [ SearchrecipeComponent ]
  })
  .compileComponents();
});

beforeEach(() => {
  fixture = TestBed.createComponent(SearchrecipeComponent);
  component = fixture.componentInstance;
  fixture.detectChanges();
});

it('should create', () => {
  expect(component).toBeTruthy();
});
});

```

=====

TS searchrecipe.component.ts

=====

```

import { Component, OnInit } from '@angular/core';
import { Router } from '@angular/router';
import { Recipes } from '../recipeModel';
import { UserService } from '../user.service';

@Component({
  selector: 'app-searchrecipe',
  templateUrl: './searchrecipe.component.html',
  styleUrls: ['./searchrecipe.component.css']
})
export class SearchrecipeComponent implements OnInit {
  itemname='';
  recipe?:Recipes[];
  constructor(private router:Router,private userService:UserService) { }

  ngOnInit(): void {
    this.recipeSearch();
  }
  recipeSearch(){
    this.userService.searchRecipe(this.itemname).subscribe(
      data=>{
        this.recipe=data;
      }
    );
  }

  onRecipeSearch() {

```

```

this.recipeSearch();
this.gotoSearchRecipe();
    }
    gotoSearchRecipe() {
        this.router.navigate(['searchrecipe']);
    }
}

```

=====

Add Recipe

=====

<> addrecipe.component.html

=====

<h2>Add Recipes</h2>

```

<div [hidden]="submitted" style="width: 400px;">
    <form (ngSubmit)="onSubmit()" >

        <div class="form-group">
            <label for="name">Recipe Name</label>
            <input type="text" class="form-control" id="itemname" required
[ (ngModel) ]="recipe.itemname" name="itemname">
        </div>

        <div class="form-group">
            <label for="name">Ingredients</label>
            <input type="text" class="form-control" id="ingredients" required
[ (ngModel) ]="recipe.ingredients" name="itemprice">
        </div>

        <div class="form-group">
            <label for="name">Cooking Steps</label>
            <textarea cols="50" rows="10" name="itemdesc"
[ (ngModel) ]="recipe.cookingsteps"></textarea>
        </div>

        <button type="submit" class="btn btn-success">Submit</button>
    </form>
</div>

<div [hidden]="!submitted">
    <h4>You submitted successfully!</h4>
    <!-- <button class="btn btn-success"
(click)="newEmployee()">Add</button> -->
</div>

```

=====

TS addrecipe.component.spec.ts

```

=====

import { ComponentFixture, TestBed } from '@angular/core/testing';

import { AddrecipeComponent } from '../addrecipe.component';

describe('AddrecipeComponent', () => {
  let component: AddrecipeComponent;
  let fixture: ComponentFixture<AddrecipeComponent>;

  beforeEach(async () => {
    await TestBed.configureTestingModule({
      declarations: [ AddrecipeComponent ]
    })
    .compileComponents();
  });

  beforeEach(() => {
    fixture = TestBed.createComponent(AddrecipeComponent);
    component = fixture.componentInstance;
    fixture.detectChanges();
  });

  it('should create', () => {
    expect(component).toBeTruthy();
  });
});

```

=====

TS addrecipe.component.ts

=====

```

import { Component, OnInit } from '@angular/core';
import { Router } from '@angular/router';
import { AdminService } from '../admin.service';
import { Recipes } from '../recipe.model';

@Component({
  selector: 'app-addrecipe',
  templateUrl: '../addrecipe.component.html',
  styleUrls: ['../addrecipe.component.css']
})
export class AddrecipeComponent implements OnInit {
  recipe: Recipes = new Recipes();
  submitted = false;

  constructor(private router: Router, private adminService: AdminService) {}

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

  addRecipe() {
    this.adminService.addRecipe(this.recipe).subscribe(

```

```

        data=>{
            this.recipe=new Recipes;
        }
    };
}
onSubmit() {
    this.submitted=true;
    this.addRecipe();
    this.gotoAddRecipe();
}
gotoAddRecipe() {
    this.router.navigate(['/addrecipe']);
}
}

```

=====

View Recipe

<> viewrecipe.component.html

=====

```

<div class="panel panel-primary">
    <div class="panel-heading">
        <h2>View & Manage Recipes </h2>
    </div>
    <div class="panel-body">
        <table class="table table-striped">
            <thead>
                <tr>
                    <th>Recipe Id</th>
                    <th>Recipe name</th>
                    <th>Ingredients</th>
                    <th>Cooking Steps</th>
                    <th>Status</th>
                    <th>Remove</th>
                    <th>Enable </th>
                    <th>Disable</th>
                </tr>
            </thead>
            <tbody *ngIf="recipe!=null">

                <tr *ngFor="let item of recipe">

                    <td>{{item.id}}</td>
                    <td>{{item.itemname}}</td>
                    <td>{{item.ingredients}}</td>

                    <td><textarea cols="30"
rows="10">{{item.cookingsteps}}</textarea></td>
                    <td>{{item.status}}</td>
                    <td><button (click)="removeItem(item.id)" class="btn btn-
success">Remove</button></td>

```



```

        <td><button (click)="enableItem(item.id)" class="btn btn-
success">Enable</button></td>
        <td><button (click)="disableItem(item.id)" class="btn btn-
success">Disable</button></td>
    </tr>

    </tbody>
</table>

</div>
</div>

```

```
=====
```

```
TS viewrecipe.component.spec.ts
```

```
=====
```

```

import { ComponentFixture, TestBed } from '@angular/core/testing';

import { ViewrecipeComponent } from './viewrecipe.component';

describe('ViewrecipeComponent', () => {
  let component: ViewrecipeComponent;
  let fixture: ComponentFixture<ViewrecipeComponent>;

  beforeEach(async () => {
    await TestBed.configureTestingModule({
      declarations: [ ViewrecipeComponent ]
    })
    .compileComponents();
  });

  beforeEach(() => {
    fixture = TestBed.createComponent(ViewrecipeComponent);
    component = fixture.componentInstance;
    fixture.detectChanges();
  });

  it('should create', () => {
    expect(component).toBeTruthy();
  });
});

```

```
=====
```

```
TS viewrecipe.component.ts
```

```
=====
```

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

```

import { Router } from '@angular/router';
import { AdminService } from '../admin.service';
import { Recipes } from '../recipe.model';

@Component({
  selector: 'app-viewrecipe',
  templateUrl: './viewrecipe.component.html',
  styleUrls: ['./viewrecipe.component.css']
})
export class ViewrecipeComponent implements OnInit {
  recipe?: Recipes[];
  constructor(private router: Router, private adminService: AdminService) { }

  ngOnInit(): void {
    this.getRecipes();
    this.gotoViewrecipe();
  }

  getRecipes() {
    this.adminService.getAllRecipes().subscribe(
      data => {
        this.recipe = data;
      }
    );
  }

  removingItem(id: number) {
    this.adminService.removeRecipeById(id).subscribe();
  }

  removeItem(id: number) {
    this.removingItem(id);
    this.ngOnInit();
    this.gotoViewrecipe();
  }

  setEnableItem(id: number) {
    this.adminService.setEnableRecipe(id).subscribe();
  }

  enableItem(id: number) {
    this.setEnableItem(id);
    this.gotoViewrecipe();
  }

  disableItem(id: number) {
    this.adminService.setDisableRecipe(id).subscribe();
    this.gotoViewrecipe();
  }

  gotoViewrecipe() {
    this.router.navigate(['viewrecipes']);
  }
}

```

=====

TS admin.service.spec.ts

```
import { Component, OnInit } from '@angular/core';
import { Router } from '@angular/router';
import { AdminService } from '../admin.service';
import { Recipes } from '../recipemodel';

@Component({
  selector: 'app-viewrecipe',
  templateUrl: './viewrecipe.component.html',
  styleUrls: ['./viewrecipe.component.css']
})
export class ViewrecipeComponent implements OnInit {
  recipe?:Recipes[];
  constructor(private router:Router,private adminService:AdminService) { }

  ngOnInit(): void {
    this.getRecipes();
    this.gotoViewrecipe();
  }

  getRecipes(){
    this.adminService.getAllRecipes().subscribe(
      data=>{
        this.recipe=data;
      }
    );
  }

  removingItem(id:number){
    this.adminService.removeRecipeById(id).subscribe();
  }

  removeItem(id:number){
    this.removingItem(id)
    this.ngOnInit();
    this.gotoViewrecipe();
  }

  setEnableItem(id: number){
    this.adminService.setEnableRecipe(id).subscribe();
  }

  enableItem(id:number){
    this.setEnableItem(id);
    this.gotoViewrecipe();
  }
}
```

```

        disableItem(id:number){
            this.adminService.setDisableRecipe(id).subscribe();
            this.gotoViewrecipe();
        }

        gotoViewrecipe(){
            this.router.navigate(['viewrecipes']);
        }
    }
}

```

=====

Ts admin.service.ts

```

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

@Injectable({
    providedIn: 'root'
})
export class AdminService {
    invalidLogin=false;
    constructor(private httpClient:HttpClient) { }

    addRecipe(recipe:Object): Observable<Object>{
        return
        this.httpClient.post('http://localhost:1018/whats_cooking/cooking/api/save
        recipe',recipe);
    }

    getAllRecipes(): Observable<any>{
        return
        this.httpClient.get('http://localhost:1018/whats_cooking/cooking/api/viewr
        ecipe');
    }

    removeRecipeById(id: number): Observable<any>{
        return
        this.httpClient.get('http://localhost:1018/whats_cooking/cooking/api/remov
        erecipe?id='+id);
    }

    setEnableRecipe(id: number):Observable<any>{
        return
        this.httpClient.get('http://localhost:1018/whats_cooking/cooking/api/enabl
        e?id='+id);
    }

    setDisableRecipe(id: number):Observable<any>{
        return
        this.httpClient.get('http://localhost:1018/whats_cooking/cooking/api/disab
        le?id='+id);
    }
}

```

```

    }
    authenticate(username:string,password:string){
      if(username ==="Admin" && password ==="Admin"){
        sessionStorage.setItem('username',username);
        return true;
      }else{
        return false;
      }
    }
    isAdminLoggedIn() {
      let user = sessionStorage.getItem('username')
      console.log(!(user === null))
      return !(user === null)
    }
    logout(){
      sessionStorage.removeItem('username');
    }
  }
}

```

=====

TS adminauth-guard.service.spec.ts

```

import { TestBed } from '@angular/core/testing';

import { AdminauthGuardService } from '../adminauth-guard.service';

describe('AdminauthGuardService', () => {
  let service: AdminauthGuardService;

  beforeEach(() => {
    TestBed.configureTestingModule({});
    service = TestBed.inject(AdminauthGuardService);
  });

  it('should be created', () => {
    expect(service).toBeTruthy();
  });
});

```

=====

TS adminauth-guard.service.ts

```

import { Injectable } from '@angular/core';

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

```

```

    constructor() { }
}

```

=====

TS app-routing.module.ts

```

import { NgModule } from '@angular/core';
import { Routes, RouterModule } from '@angular/router';
import { AddrecipeComponent } from '../addrecipe/addrecipe.component';
import { AdminloginComponent } from '../adminlogin/adminlogin.component';
import { AdminlogoutComponent } from
'../adminlogout/adminlogout.component';
import { SearchrecipeComponent } from
'../searchrecipe/searchrecipe.component';
import { UserloginComponent } from '../userlogin/userlogin.component';
import { UserlogoutComponent } from '../userlogout/userlogout.component';
import { UserregisterComponent } from
'../userregister/userregister.component';
import { ViewrecipeComponent } from '../viewrecipe/viewrecipe.component';

const routes: Routes = [
  {path:'',redirectTo:'',pathMatch:'full'},
  {path:'adminlogin',component:AdminloginComponent},
  {path:'addrecipe',component:AddrecipeComponent},
  {path:'viewrecipes',component:ViewrecipeComponent},
  {path:'adminlogout',component:AdminlogoutComponent},
  {path:'userregister',component:UserregisterComponent},
  {path:'userlogin',component:UserloginComponent},
  {path:'searchrecipe',component:SearchrecipeComponent},
  {path:'userlogout',component:UserlogoutComponent}
];

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

```

=====

<> app.component.html

```

<html>
  <body>
<nav class="navbar navbar-expand-sm      bg-success navbar-dark">
  <!-- Links -->
  <ul class="navbar-nav">

    <li class="nav-item">

```

```

        <a *ngIf="!adminLogin.isAdminLoggedIn() &&!userLogin.isUserLoggedIn()"
routerLink="adminlogin" class="nav-link"
routerLinkActive="active">Admin</a>
</li>
<li class="nav-item">
    <a *ngIf="adminLogin.isAdminLoggedIn()" routerLink="" class="nav-link"
routerLinkActive="active">Home</a>
</li>

<li class="nav-item">
    <a *ngIf="adminLogin.isAdminLoggedIn()" routerLink="addrecipe"
class="nav-link" routerLinkActive="active">Add Recipe</a>
</li>

<li class="nav-item">
<a *ngIf="adminLogin.isAdminLoggedIn()" routerLink="viewrecipes"
class="nav-link" routerLinkActive="active">Manage Recipes</a>
</li>
<li class="nav-item">
    <a *ngIf="adminLogin.isAdminLoggedIn()" routerLink="adminlogout"
class="nav-link" routerLinkActive="active">Logout</a>
</li>
<li class="nav-item">
    <a *ngIf="!adminLogin.isAdminLoggedIn() && !userLogin.isUserLoggedIn()"
routerLink="userlogin" class="nav-link" routerLinkActive="active">User</a>
</li>
<li class="nav-item">
    <a *ngIf="!adminLogin.isAdminLoggedIn() && !userLogin.isUserLoggedIn()"
routerLink="userregister" class="nav-link"
routerLinkActive="active">Register</a>
</li>
<li class="nav-item">
    <a *ngIf="!adminLogin.isAdminLoggedIn() && userLogin.isUserLoggedIn()"
routerLink="" class="nav-link" routerLinkActive="active">Home</a>
</li>
<li class="nav-item">
    <a *ngIf="!adminLogin.isAdminLoggedIn() && userLogin.isUserLoggedIn()"
routerLink="searchrecipe" class="nav-link" routerLinkActive="active">Serch
Recipes</a>
</li>

<li class="nav-item">
    <a *ngIf="!adminLogin.isAdminLoggedIn() && userLogin.isUserLoggedIn()"
routerLink="userlogout" class="nav-link"
routerLinkActive="active">logout</a>
</li>
</ul>
</nav>
<div class="container">
    <br>
    <h2 style="text-align: center;">Cooking Client</h2>
    <hr>
    <div class="card">
        <div class="card-body">

```

```

        
        <router-outlet></router-outlet>
      </div>
    </div>
  </div>

<footer class="footer">
  <div class="container">

    </div>
  </footer>
</body>
</html>

```

=====

TS app.component.spec.ts

```

import { TestBed } from '@angular/core/testing';
import { RouterTestingModule } from '@angular/router/testing';
import { AppComponent } from './app.component';

describe('AppComponent', () => {
  beforeEach(async () => {
    await TestBed.configureTestingModule({
      imports: [
        RouterTestingModule
      ],
      declarations: [
        AppComponent
      ],
    }).compileComponents();
  });

  it('should create the app', () => {
    const fixture = TestBed.createComponent(AppComponent);
    const app = fixture.componentInstance;
    expect(app).toBeTruthy();
  });

  it(`should have as title 'cookingclient'`, () => {
    const fixture = TestBed.createComponent(AppComponent);
    const app = fixture.componentInstance;
    expect(app.title).toEqual('cookingclient');
  });

  it('should render title', () => {
    const fixture = TestBed.createComponent(AppComponent);
    fixture.detectChanges();
    const compiled = fixture.nativeElement;
    expect(compiled.querySelector('.content
span').textContent).toContain('cookingclient app is running!');
  });

```



```
});
```

```
=====
```

TS app.component.ts

```
import { Component } from '@angular/core';
import { AdminService } from '../admin.service';
import { UserService } from '../user.service';

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {
  title = 'cookingclient';
  constructor(public userLogin:UserService,public
adminLogin:AdminService){}
}
```

```
=====
```

TS app.module.ts

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

import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';
import { AdminloginComponent } from './adminlogin/adminlogin.component';
import { AdminlogoutComponent } from
'./adminlogout/adminlogout.component';
import { AddrecipeComponent } from './addrecipe/addrecipe.component';
import { ViewrecipeComponent } from './viewrecipe/viewrecipe.component';
import { UserloginComponent } from './userlogin/userlogin.component';
import { UserregisterComponent } from
'./userregister/userregister.component';
import { SearchrecipeComponent } from
'./searchrecipe/searchrecipe.component';
import { UserlogoutComponent } from './userlogout/userlogout.component';
import { FormsModule } from '@angular/forms';
import { HttpClientModule } from '@angular/common/http';

@NgModule({
  declarations: [
    AppComponent,
    AdminloginComponent,
    AdminlogoutComponent,
    AddrecipeComponent,
```

```

        ViewrecipeComponent,
        UserloginComponent,
        UserregisterComponent,
        SearchrecipeComponent,
        UserlogoutComponent
    ],
    imports: [
        BrowserModule,
        AppRoutingModule,
        HttpClientModule,
        FormsModule
    ],
    providers: [],
    bootstrap: [AppComponent]
  })
  export class AppModule { }

```

=====

TS recipemodel.ts

```

export class Recipes{
    id:number=0;
    itemname:string='';
    ingredients:string='';
    cookingsteps:string='';
    status:string='';
}

```

=====

TS user.service.spec.ts

```

import { TestBed } from '@angular/core/testing';

import { UserService } from './user.service';

describe('UserService', () => {
  let service: UserService;

  beforeEach(() => {
    TestBed.configureTestingModule({});
    service = TestBed.inject(UserService);
  });

  it('should be created', () => {
    expect(service).toBeTruthy();
  });
});

```

=====

TS user.service.ts

```
import { HttpClient } from '@angular/common/http';
import { Injectable } from '@angular/core';
import { Observable } from 'rxjs';
import { Users } from '../usermodel';

@Injectable({
  providedIn: 'root'
})
export class UserService {
  users?:Users;
  constructor(private httpClient:HttpClient) { }
  userLogin(username:string,password:string):Observable<any>{
    return
    this.httpClient.get("http://localhost:1029/whats_cooking/cooking/api/login
user?username="+username+"&password="+password);
  }
  userRegister(user:Object): Observable<Object>{
    return
    this.httpClient.post('http://localhost:1029/whats_cooking/cooking/api/save
user',user);
  }
  searchRecipe(itemname:string):Observable<any>{
    return
    this.httpClient.get("http://localhost:1029/whats_cooking/cooking/api/searc
hrecipe?itemname="+itemname);
  }
  getRecipeDetails(id:number):Observable<any>{
    return
    this.httpClient.get("http://localhost:1029/whats_cooking/cooking/api/getre
cipedetails?id="+id);
  }

  authenticate(username:string, password:string) {

    this.userLogin(username,password).subscribe(
      data=>{
        this.users=data;
      }
    );
    if (this.users) {
      console.log(this.users?.username)
      sessionStorage.setItem('name', username)
      console.log("username:"+username);
      return true;
    } else {
      return false;
    }
  }
  isUserLoggedIn() {
```

```

    let name = sessionStorage.getItem('name')
    console.log(!(name === null))
    return !(name === null)
  }

  logOut() {
    sessionStorage.removeItem('name')
  }
}

```

=====

TS userauth-guard.service.spec.ts

```

import { TestBed } from '@angular/core/testing';

import { UserauthGuardService } from './userauth-guard.service';

describe('UserauthGuardService', () => {
  let service: UserauthGuardService;

  beforeEach(() => {
    TestBed.configureTestingModule({});
    service = TestBed.inject(UserauthGuardService);
  });

  it('should be created', () => {
    expect(service).toBeTruthy();
  });
});

```

=====

TS userauth-guard.service.ts

```

import { Injectable } from '@angular/core';

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

  constructor() { }
}

```

=====

TS usermodel.ts

```
export class Users{
  id:number=0;
  name:string='';
  email:string='';
  mobile:string='';
  username:string='';
  password:string='';
  active:boolean=false;
}
```

=====

<> index.html

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>Cookingclient</title>
  <base href="/">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="icon" type="image/x-icon" href="favicon.ico">
</head>
<body>
  <app-root></app-root>
</body>
</html>
```

=====

TS main.ts

```
import { enableProdMode } from '@angular/core';
import { platformBrowserDynamic } from '@angular/platform-browser-
dynamic';

import { AppModule } from './app/app.module';
import { environment } from './environments/environment';

if (environment.production) {
  enableProdMode();
}

platformBrowserDynamic().bootstrapModule(AppModule)
  .catch(err => console.error(err));
```

=====

TS tests.ts

```
// This file is required by karma.conf.js and loads recursively all the
.spec and framework files
```

```
import 'zone.js/dist/zone-testing';
import { getTestBed } from '@angular/core/testing';
import {
  BrowserDynamicTestingModule,
  platformBrowserDynamicTesting
} from '@angular/platform-browser-dynamic/testing';
```

```
declare const require: {
  context(path: string, deep?: boolean, filter?: RegExp): {
    keys(): string[];
    <T>(id: string): T;
  };
};
```

```
// First, initialize the Angular testing environment.
getTestBed().initTestEnvironment(
  BrowserDynamicTestingModule,
  platformBrowserDynamicTesting()
);
// Then we find all the tests.
const context = require.context('./', true, /\.spec\.ts$/);
// And load the modules.
context.keys().map(context);
```