

App.routes.ts ➔

```
import { Routes } from '@angular/router';
import { LoginComponent } from '../components/login/login.component';
import { DashboardComponent } from
'../components/dashboard/dashboard.component';
import { authGuard } from '../auth.guard';
import { ProfileComponent } from '../components/profile/profile.component';
import { SettingsComponent } from '../components/settings/settings.component';
import { InboxComponent } from '../components/inbox/inbox.component';

export const routes: Routes = [
  {
    path: '',
    redirectTo: 'login',
    pathMatch: 'full'
  },
  {
    path: 'login',
    component: LoginComponent
  },
  {
    path: 'dashboard',
    component: DashboardComponent,
    canActivate: [authGuard],
    children: [
      {
        path: 'profile',
        component: ProfileComponent
      },
      {
        path: 'settings',
        component: SettingsComponent
      },
      {
        path: 'inbox',
        component: InboxComponent
      }
    ]
  },
  {
    path: '**',
    redirectTo: 'login'
  }
];
```

Login.component.ts →

```
import { Component, OnInit } from '@angular/core';
import { FormsModule } from '@angular/forms';
import { MatCardModule } from '@angular/material/card';
import { Router } from '@angular/router';
import { AuthService } from '../../services/auth.service';
import { MatButtonModule } from '@angular/material/button';
import { MatFormFieldModule } from '@angular/material/form-field';
import { MatInputModule } from '@angular/material/input';
import { MatProgressSpinnerModule } from '@angular/material/progress-spinner';
import { LoadingService } from '../../services/loading.service';
import { CommonModule } from '@angular/common';
import { Observable } from 'rxjs';

@Component({
  selector: 'app-login',
  standalone: true,
  imports:
[MatCardModule,CommonModule,MatProgressSpinnerModule,MatFormFieldModule,MatInputModule,FormsModule,MatButtonModule],
  templateUrl: './login.component.html',
  styleUrls: ['./login.component.css']
})
export class LoginComponent implements OnInit{
  username:string = '';
  password:string = '';
  timeOut!:Observable<boolean>;

  constructor(private loader:LoadingService,private route: Router,private
authUser: AuthService){ }
  ngOnInit(): void {
    this.timeOut = this.loader.loading$;
  }
  authenticateUser(){
    this.loader.show();
    let authentication = this.authUser.authUser(this.username,this.password)

    setTimeout(() => {
      if(authentication){
        this.route.navigate(['/dashboard'])
        console.log('success')
      }else{
        alert('user not found!!');
        console.log('fail')
      }
      this.loader.hide();
    }, 2000);
  }
}
```

```

    }

}

```

Login.component.html ➔

```

<h1>Login Form</h1>
<mat-card>
  <mat-card-subtitle>
    <i>Username: Alan</i><br>
    <i>password: 123</i>
  </mat-card-subtitle>
  <mat-card-content>
    <div class="form">
      <mat-form-field>
        <mat-label>Username</mat-label>
        <input matInput [(ngModel)]="username" type="text">
      </mat-form-field>
      <br>
      <mat-form-field>
        <mat-label>Password</mat-label>
        <input matInput [(ngModel)]="password" type="password">
      </mat-form-field>
      <br>
      <button mat-raised-button
(click)="authenticateUser()">Login</button>
    </div>
    @if(timeOut | async){
      <div class="spinner">
        <mat-spinner></mat-spinner>
      </div>
    }
  </mat-card-content>
</mat-card>

```

Login.component.css ➔

```

.form{
  margin: 40px;
}
mat-form-field{
  width: 40vw;
}

```

```

mat-card{
  margin-left: 20vw;
  margin-right: 20vw;
}
.spinner{
  position: fixed;
  top: 0;
  left: 0;
  width: 100%;
  height: 100%;
  background: rgba(0, 0, 0, 0.5);
  display: flex;
  justify-content: center;
  align-items: center;
}

```

auth.guard.ts➔

```

import { inject } from '@angular/core';
import { CanActivateFn, Router } from '@angular/router';
import { LoadingService } from '../services/loading.service';

export const authGuard: CanActivateFn = (route, state) => {
  let userFound = localStorage.getItem('loggedUser')
  const router = inject(Router);
  const loaderService = inject(LoadingService);

  return new Promise<boolean>((resolve)=>{
    if(userFound){

      resolve(true);
      //return true
    }
    else{
      setTimeout(()=> {
        router.navigate(['/login']);
        resolve(false)
      },0)

      // return false
    }
  })
};

```

Auth.service.ts ➔

```

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

```

```

@Injectable({
  providedIn: 'root'
})
export class AuthService {
  userFound = false;
  users = [
    {
      username : "Alan",
      password : '123'
    },
    {
      username : "Amal",
      password : '123'
    }
  ]
  constructor() { }

  authUser(username:any,password:any){
    for(let i=0; i<this.users.length; i++){
      if(username == this.users[i].username &&
password==this.users[i].password){
        localStorage.setItem('loggedUser',username)
        this.userFound = true;
        return true
      }
    }
    return false
  }
}

```

Loading.service.ts ➔

```

import { Injectable } from '@angular/core';
import { BehaviorSubject, Observable } from 'rxjs';

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

  private loading = new BehaviorSubject<boolean>(false);
  public loading$:Observable<boolean> = this.loading.asObservable();

  constructor() { }

```

```
show(){  
  this.loading.next(true);  
}  
hide(){  
  this.loading.next(false);  
}  
}
```