**AUTH GUARDS:**

It contains the 5 interface

1. CanActivate

2. CanActivateChild

3. CanDeactivate

4. Resolve

5. CanLoad

**1. CanActivate**

* Allows us to protect from unauthorized user.
* If all guards returns true navigation continues.
* If any guard returns false navigation cancelled.
* CanActivate interface contains canActivate() method.
* canActivate() contains 2 parameters

1. ActivatedRouteSnapshot
2. RouterStateSnapshot

**2. CanActivateChild:**

* This will use if we have multiple child routes and we want to same functionality for all the child routes.
* This interface implements canActivateChild() method.
* It will return Boolean value,observable<Boolean>,urlTree,Promise
* This route guard will apply for all the routes but it will not apply to the parent route.

**3. CanDeactivate**

* It decides whether a user can navigate away from a route or not.
* CanDeactivate interface implements CanDeactivate() method.
* This method contains 4 parameter

1. Component: Is a generic type.
2. CurrentRoute:ActivatedRouteSnapshot
3. CurrentState:RouterStateSnapshot
4. nextState:it is optional

**4. Resolve:**

* This is implemented to be data provider.
* A data provider class can be used with the router to resolve data during navigation.
* This interface defines the resolve() method.
* It contains 2 parameters

1. Route
2. State

* This method will return Observable and Promise these are generic type.

**5. CanLoad:**

* Interface that a class can implement to be a guard deciding if children can be loaded.
* It implements canLoad () method.
* This method contains 2 parameters

1. route

2. segments

**Name**: Meghana B