Details about Security

The security implementation used in this project is very basic, which is “canActivate route” from Angular Authentication service. It is a type of router guard.

We may want some routes to only be accessible once the user has logged in or accepted Terms & Conditions. We can use route guards to check these conditions and control access to routes.

Route guards can also control whether a user can leave a certain route. For example, say the user has typed information into a form on the page, but has not submitted the form. If they were to leave the page, they would lose the information. We may want to prompt the user if the user attempts to leave the route without submitting or saving the information.

Guards return either true if the user can access a route or false if they can’t. They can also return an Observable or Promise that later on resolves to a Boolean in case the guard can’t answer the question straight away, for example it might need to call an API. Angular will keep the user waiting until the guard returns true or false

When canActivate returns true, the user can activate the route. When canActivate returns false, the user cannot access the route. We are allowing access when the user is logged in.

canActivate can also be used to notify the user that they can't access that part of the application or redirect them to the login page.