A Guard Route is part of Angular in which it protects the route from being visited, and you would need to be authenticated to access the route. (Fain & Moiseev, 2019) Upon logging in it will then direct you through the guarded route, or if authentication fails it will direct you back into the login page. There are many different types of guard routes in Angular, such as canActivate, canActivateChild, canDeactivate, Resolve and canLoad. (Fain & Moiseev, 2019)

Guard routes are helpful with many different aspects. Such as the ability to navigate to the route if they user was able to authenticate and get authorization. It is able to give a reminder to users about changes that are not saved when they attempt leave the route. Is able to only allow the navigation once the user is authorized. (Fain & Moiseev, 2019)

The different interfaces such as canActivate will grant or deny route navigation. canActivateChil will arbitrate the child route of navigation. Resolve will make sure the data that is required will be retrieved before navigation starts to the route. (Fain & Moiseev, 2019) canLoad authorizes or denies modules that are lazy-loading. (Fain & Moiseev, 2019) And lastly canDeactivate is an interface that will authorize or deny navigation that is directed away from the route. (Fain & Moiseev, 2019)

An example of canActivate is:

Const routes: Routes = [

{path: ‘tool’,

Component: ToolDetailComponent,

canActivate: [LoginGuard]}

];

Notice how the canActivate is inside the curly brackets under path. It will also return true or false depending on the correct credentials or not which will then authorize the user to continue to the route. (Fain & Moiseev, 2019)

Reference:

Fain, Y., & Moiseev, A. (2019). *Angular Development with TypeScript* (2nd ed.). Manning.