\_services -> here are all services

guards -> here is a guard that does not allow us to change the route if we are not registered respectively if there is no user in the session storage

includes/layouts -> here is main layout of the application

models -> here are all models

modules -> here are all modules with main components (here are rovers, register page etc...)

shared/components -> Here are all components which we used as building blocks and here is components module which imports and exports all this components so we can use them where we need them

app-routing.module.ts -> here is main routes with lazy loading in it

environment.ts -> here is variable API\_KEY, i registered so you can use my key, but when you change key, here you change it, and everything should work just fine, because you are limited on 50 requests per day, so you must register for more.

Additionally:

In the application we have two global components which are loading and toast, and we controll them by their services, toast.service.ts and loading.service.ts, with the help of toast messages we can tell user if so mething went wrong or when they are successfully registered.

Loading is there to help us when we fetching some data so the user can know that somehing is happening