**Navigation using ” React Navigation”**

**First : The need for Navigation in our app:**

* Since the app consists of many screens , we needed to use navigation to achieve moving between these screens.

* There are many ways to navigate inside a react native app to work on both (ios-android) platforms .The most popular options are :
  1. Using **React Navigation** :which is a standalone library that allows developers to set up the , screens of an app with just a few lines of code , and it’s a javascript implementation for navigation
  2. Using **React Native Navigation** :which is also a standalone library wrote by wix.It uses the original native components instead of depending on pure Javascript alternatives.
* We chose the first option which is :**React Navigation.**
* The reason we chose React Navigation is because :
  1. Its documentation is great, and it covers the most important scenarios in dealing with navigation.
  2. It provides an easy to use navigation solution , and it’s not complex to set it up and start working ,as it doesn’t require editing native files.
  3. It’s recommended in the react native documentation.

**Second :The main components we used while using React Navigation library:**

* A navigator is a component that implements a navigation pattern (eg: tabs)
* Each route must have a name and a screen component.
  + The name is usually unique across the app.
  + The screen component is a React component that is rendered when the route is active.
  + The screen component can also be another navigator.
* Each navigator must have one or more routes.
  + A navigator is a parent of a route.
  + A route is a child of a navigator.
* In our app we used two kinds of navigators:
  + Stack Navigator:Screens are stacked on top of each other.
  + Tab Navigator: User can switch between different tabs.

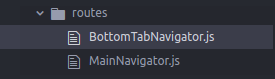
**Third : installing the library :**

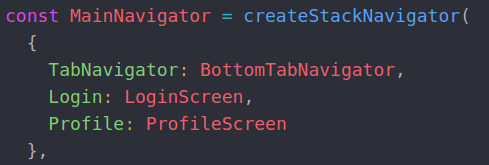
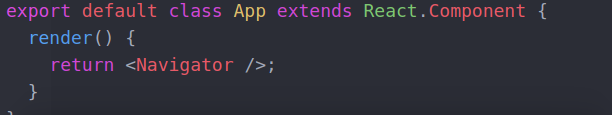


We wrote the following command in the terminal in the same directory of the project in order to install the library.

**Fourth : using the library :**

* We created a folder called ‘routes’ in the src folder.

****

* It consists of two files:
  + **BottomTabNavigator.js:**
    - We imported the createBottomTabNavigator function .
    - Inside this function we created 5 routes :
      * Featured.
      * MyServices.
      * NewService.
      * Notifications.
      * Account.
    - Inside the **naviagationOptions** object we assigned each of these routes to a certain screen component (*the screen components are imported from a folder we made called screens and we put our screen components inside it*) and we also determined icon & label for each of them to be shown in the Tab Navigator
    - The following image shows a code snippet of what’s described 
  + **MainNavigator.js:**
    - The main navigator is a stack navigator which contains the following : the tab navigator, and any other screen that’s supposed to be outside the tab navigator or just an inner screen of a screen that already exists in the tab navigator .
    - For now we have the following routes in the main navigator as shown in the image below :
      * 
    - Then depending on which screen the user is in..we change the title and styling of the stack navigation header ..example of the code used is below :
      * 
* Finally in app.js we imported the main navigator and rendered it:
  + 
* And this was the view after applying the navigation for the first time to the app:
  + 