Objectives:

- Drawer, Bottom, Top Navigation.
- Making API get request using Axios.
- Creating Custom Hook

Task#1: App Layout 5 points

Design a **Search App** that provides a lookup functionality through the API's get request.

The App has a single **Home-Screen (App.js)**, with two components:

1. Lookup Component (**Lookup.js**)
   a. Responsible to get user input and forward the search-term through the API Get request.

2. Result Component (**Result.js**)
   a. Responsible to render the result received back from the Get-request.

Task#2: Initial Lookup 5 points

1. Note down the API-key and base-URL in a **APIInfo.js** file. Use axios-create function to establish a connection with the external-App.
2. When the app starts, it should automatically do an **initial-lookup** using some default search-term.
3. While making the get-request, set the term, limit, location, and any other required params.

Task#3:

Implement **any ONE** of the two options.

Option#1: Yelp API 20 points

SearchApp communicates with Yelp through API request to provides the following two search options using either Bottom-tab, top-tab or drawer navigation. Also, add appropriate search icons for each option:

1. Search the business for a given phone number in the search (https://docs.developer.yelp.com/reference/v3_business_phone_search)

2. Fetch at-least three Reviews for a given business in the search (https://docs.developer.yelp.com/reference/v3_business_reviews)

Option#2: Any Other API **20 points**

1. Instead of Yelp, select a different App (for example: IMDB, Youtube, Weather, Twitter, Google, Facebook etc.) and provide a search functionality through the API's get request for user inputted keywords. The search term for instance can be–

   a. For IMDB API: the search-term can be movie or actor, and API result will include additional details on the movie/actor.
   b. For Youtube API, the search-term can be keywords, and API result will include list of videos containing some or all of the keywords.
   c. For Weather API, the search-term can be location, and API result will include forecast information (hourly/daily etc.)

Task#4: CustomHook **10 points**

1. Create a folder called CustomHooks with a **useAPIresponse.js** file, which should **define and export a custom-hook**.
2. Extract the searchAPI logic from the App.js (Home Screen) and make it available through the useAPIresponse hook instead.
3. This hook should **return state-variable(s), and a setter-function** responsible to set the values of the variables. Import and use this hook in App.js.

Submission:

- Screenshot(s) with search-term(s) along with their respective output.
- Code from 5 files: App.js, APIInfo.js , Lookup.js, Result.js, and useAPIresponse.js

Put the above code-and-screenshots in a word or pdf document, and upload the document through the submission link made available on blackboard.