COMP 306 – Fall 2018 – Assignment 06
Due: Monday, 5 Nov, beginning of class

Problem: In this assignment you will create an app that stores a list of names (first and last) and emails online. Your app will obtain the list by connecting to a Rest API online (as we've done in class). You will also be able to add names to the list (though you won't be able to delete or modify them).

Your app must have 3 tabs, each of which is a separate screen. You will navigate from screen to screen via tabs. The screens are:

1. A splash screen. This screen must have a background that is colored and must have a Text box (with a different color background) and an image. This screen may be part of App.js.

2. A screen that displays lists. This must be the second screen. This screen must store the information for each person in the app *state* using an *array* and display it using a **`flatlist`** component. You must obtain the information from the internet. The connection information is given below. Display only first and last name in the **`flastlist`**. When a user touches a name, the person's email address must appear in a Text box at the bottom of the app. This screen must be in it's own file, not in App.js.

3. The last screen must enable the user to enter a new person (first name, last name and email address). You must connect to the server and pass this information to be entered into a database. The service will either return the string "success" or "fail". This screen must have "update" and "cancel" buttons. The cancel button should take the user back to the second screen without making any changes to the list. . This screen must be in its own file, not in App.js.

There must be a consistent color code across all three screens and the background of each screen must be either a color or an image.

Accessing data. You can get all the values in the database by connecting to the following REST service. The result is an array of objects encoded in JSON. The fields (or variables) in an object are:

ID: an integer that is the unique key. You'll need to use the **`keyExtractor`** property of the Flatlist to associate this field with the key (see the HTTP/Fetch slides).
First: the first name.
Last: the last name.
Email: the email address.

The URL of the REST service to get a JSON array of these objects is:

`http://cs-ithaca.eastus.cloudapp.azure.com/~barr/getNames.php`

Uploading data. You will upload JSON encoded data using POST. You must create a single object and encode it in JSON. It must have the same fields as the object described previously.

The URL of the REST service to put a JSON object is:

http://cs-ithaca.eastus.cloudapp.azure.com/~barr/putNames.php

Submission. I'll look at your app in lab. Submit your App.js file and any other files that you create by zipping them into a folder and submitting the zipped folder through Sakai.