Assignment 4

For ACS-3909-050

**Due: December 12, 2022**

**Information:**

* Only submissions **of groups of exactly 3 students** will be accepted unless you have explicit allowance by the instructor.
* Your submission must **clearly indicate all group members names** in each file (use a comment in .js files and in .html files)
* Your **submitted filenames** must clearly indicate to which assignment and exercise they belong. For example, if you solve Assignment 1, exercise 1.1 name your file: “A1\_E1-1\_app.js” or something similar.
* Only **one member** of the group needs to submit for the entire group.
* Any changes in group compositions must be **pro-actively** communicated by the students to the teaching assistant ([gill-a50@webmail.uwinnipeg.ca](mailto:gill-a50@webmail.uwinnipeg.ca)) **and** the instructor ([m.beck@uwinnipeg.ca](mailto:m.beck@uwinnipeg.ca)) by email.
* Keep an eye on Nexus for clarifications/corrections of this assignment!

The exercises will ask you to make yourself familiar with functions, packages, libraries, etc. that we have not explicitly introduced in the lectures. This is by design. I am linking for each exercise resources that can help you in finding the information you might need to fulfill the exercise.

In some instances I will disallow or restrict the modules you are allowed to use. Besides theses instances there are many ways to do things, so see the resources I give you as hints – if you find another (even better) way of doing things: props to you!

# Exercise 4.1 React.js

**Possibly useful resources:**

* React Handling Events: <https://reactjs.org/docs/handling-events.html>

Download the React-example, which is a cleaned up version of the page we have programmed in Lecture 10. Add the following functionality to the *ImageForm*-Component:

* There must be a second Checkbox with label: “FixRatio”
* When it is checked the current ratio between width and height of the images requested is locked
* The sliders will move together when either of them is used, while the FixRatio checkbox is checked.
* All previous functionality of the page must stay intact (i.e., requesting images from Picsum with the given width and height and either grayscaled or not)

A quick demonstration of the new functionality is uploaded as a video to Nexus.

# Exercise 4.2 Multiple Pages in React

**Resources:**

* Conditional Rendering in React: <https://reactjs.org/docs/conditional-rendering.html>
* Navs: <https://react-bootstrap.github.io/components/navs/>
* Cards: <https://react-bootstrap.github.io/components/cards/>

In this exercise you are asked to create a (client-side-rendered) app that has multiple “pages”. You will also use the React-Bootstrap library that will give your app the typical Bootstrap look-and-feel, but is entirely rebuilt in React.

**Part 1:** Create an HTML-file that loads React-Bootstrap and the necessary CSS for using it. Create these simple components with React-Bootstrap components in your file:

* A “Nav” with 3 distinct links
* A “Card” with some image (your choice), a title, and some text

You do not have to worry about functionality of any of the links or buttons for this part.

**Part 2:** Now, edit your file to add the following functionality to it:

* Write an “App”-Component that will encapsulate the entire page, i.e., all components are subcomponents of the “App”-Component (or subcomponents of a component that is eventually a subcomponent of “App”). This App-component renders the following as return:

return (  
 <div className="main">  
 {routing()}  
 </div>  
)

* Create two more “pages” that will be accessed by clicking the links in the NavBar. You can populate these pages with any React or React-Bootstrap components.
* Clicking any of the links in the NavBar will change an internal state of the “App”-component. You can call this state “page” for example.
* Whenever that state is updated the App will render a different page. A demonstration of this functionality is uploaded as video on Nexus.

**Hint:** The function “routing()” is not defined, yet. You will have to write it! That function should check the App’s state and return the respective “page” to be rendered.