CSCI 2720 – Building Web Applications

Assignment Two: Simple Image System in React

Released: 25th October 2023

Due: 23:59, 15th November 2023

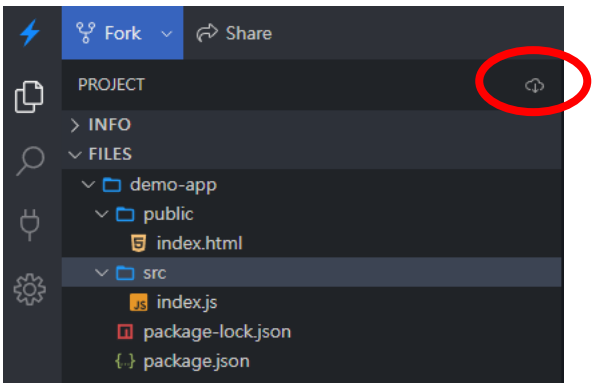
Synopsis:

You are going to set up a system for showing images using React. It includes different features specified in Problems 1-5.

Environment setup:

You may start with the given code in Lab 5 (the picture system) and Lab 6 (via StackBlitz). As StackBlitz does not support file upload with free accounts, you need set up Node.js and npm on your local computer to finish the assignment. Noide.js of version at least v16 should be used. You may get started with these steps:

1. Download the Lab 6 project from http://stackblitz.com/edit/node-inpmqg, or your finished version of Lab 6.



2. In lab06, it is very easy to follow page 8 of the slides to render the app in StackBlitz. To perform the same locally on your own computer, you should follow the below instructions from (a) to (e):
   a. Download Node.js: Download | Node.js (nodejs.org), **please download node.js v18.17.0 to avoid potential problems**, and here are the links:

      Windows: https://nodejs.org/dist/v18.17.0/node-v18.17.0-x64.msi

      MacOS: https://nodejs.org/dist/v18.17.0/node-v18.17.0.pkg

      Source Code: https://nodejs.org/dist/v18.17.0/node-v18.17.0.tar.xz

   b. CD to the project path: cd [/path_to]/demo-app
   c. Install the react and react-scripts package by running: npm install react react-dom react-scripts.
   d. Install the react-router-dom by running: npm install react-router-dom.

e. Run npm start: npm start.

3. The Lab 5 images folder can be put into ***public/***.
4. Your code from Lab 5 will be useful in this assignment. Copy the relevant code to your ***src/index.js***.

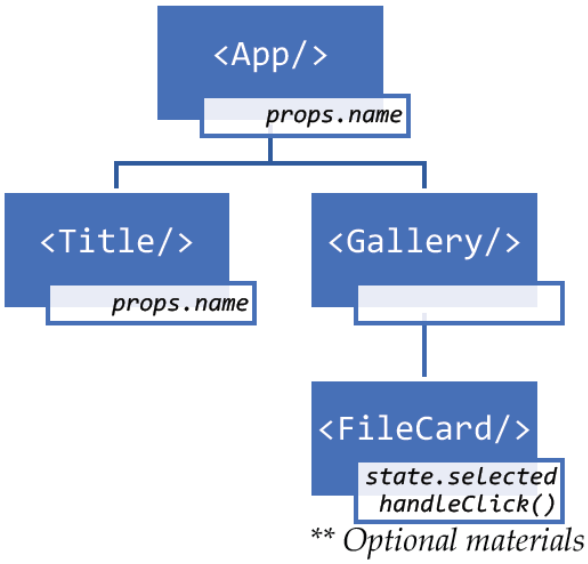
You may use the following declarations in your .js file:

```
import ReactDOM from "react-dom/client";
import React from 'react';
import {BrowserRouter, Routes, Route, Link,} from 'react-router-dom';
```

Problem 1: Basic system (25%)

(Done in Lab 5) Build an image system to display an array of images files. The file names and relevant information of each file are stored as a variable in a JS array.

The React components in the basic system are structured in this manner:



** *Optional materials*

The images are shown in a group of Bootstrap cards with a width of 200px each, using the **map()** function to render multiple **<FileCard/>** components. The filename and year are shown along with the image. More details of Problem 1 **are specified in Lab 5**.

*Note: In this assignment, you will use another event to replace the optional materials (i.e., **handleClick()** function) in Lab 5.*

Problem 2: Mouseover event (25%)

In the optional materials in Lab 5, you are instructed to set up an **onClick** event to enlarge the image card to 100% width when clicked. The file remarks are shown as well. This is be done using the **onClick** event to change the state **selected** to the index of the image.

Now, change this to an **onMouseEnter** event so that when the cursor hovers over the image card, and enlarge the card to **400px** of width instead of **200px**. Return to normal(**200px**) when the cursor moves away using **onMouseLeave**.

Problem 3: Routing menu (10%)

(Partially done in Lab 6) Add a navigation menu for routing within the page. Show three items:

1. Home: with *path /* should show component **<Home/>**
2. Images: with *path /gallery* should show component **<Gallery/>** (i.e., Problem 1-2)
3. Slideshow: with path /slideshow should show a new component **<Slideshow/>**

When refreshing a page or typing in a URL with path directly, the correct routed component will be shown. Set up a **<NoMatch/>** component to catch all URL that cannot found.

Problem 4: Slideshow (30%)

In component **<Slideshow/>**, display the following:

1. Five buttons: *Start slideshow*, *Stop slideshow*, *Slower*, *Faster*, *Shuffle*.
2. Image showing item [0] in the array
3. Image file name

Set up *five* event handlers to deal with the buttons when they are clicked:

- *Start slideshow*: the next image (and filename) in the array should be shown one by one, and looping back to the beginning after each round (default interval: every 1500ms)
- *Stop slideshow*: there should be no more change in the image
- *Slower*: the changing interval would be increased by 200ms.
- *Faster*: the changing interval would be decreased by 200ms. The interval should not be decreased lower than 200ms.
- *Shuffle*: Change the order of existing images [Hints: use ***Math.floor(Math.random())***. Note that ***math.random()*** will generate a random number between 0 and 1.]

There should be at least two state variables being changed to affect the React image display, e.g., **currentImageID** and **currentInterval**. They should be accessible in *Developer Tools -> Components -> ... ->Slideshow*. You may use more states if needed. *Note: State variables in functional components may not be shown properly.*

You may design the page layout to your liking, as long as all required items are clearly shown. We WILL NOT test the Slower and Faster buttons before starting the slideshow.

Problem 5: Component diagram (10%)

Inside the **<Home/>** component, create a tree diagram representing the hierarchy of React components you've created in your ***app.jsx*** file, excluding components from libraries such as **<Route/>**. The diagram should be saved as a ***PNG or JPG image in the public folder***, and you can use any software of your choice to create the diagram. The diagram should clearly show the hierarchical relationship between the components. Additionally, provide a list of props, states, and event handlers for each component in the diagram. An example is provided in Problem 1.

Libraries, features, and frameworks:

You should be using the latest version of React and ReactRouter with no other libraries. They are already installed in Lab 6. Lab 5 has incorporated the use of Bootstrap, to be added to the HTML file. Other than anything specified, there is no cosmetic requirement for this assignment. You are welcome to implement extra styles and features at your own ability. It is fine for extra React components or different hierarchy from required in this document.

You are allowed to utilize additional methods beyond what has been covered in our lectures and labs. If you use any unique methods, please include a readme file that provides instructions for our TA to execute your files. However, please note that you cannot expect our TA to use any external software or tools to run your files. During the grading process, we will only utilize Google Chrome and Node.js.

Submission:

We will only visit your web page submission using ***Google Chrome*** (almost-latest versions) and React development server. Please utilize the ***Developer Tools*** with ***React Developer Tools*** for debugging.

Plagiarism is heavily penalized. Never share your code. Please read this article carefully: https://www.cuhk.edu.hk/policy/academichonesty/

Include your full name and student ID ***on top of all code files using comments***. You should also include the required ***code header for honesty declaration***. Check that the file names and formats are correct. Zip all your files and submit it on the course site on BlackBoard. You can have unlimited submission. We will only grade you latest zip file. Name your zip file as:

`(SID)_asg1.zip`

Please ***keep the folder structure from Lab 6***. Your zip file should only include:

1. One `.html` file
2. One `.js/.jsx` file
3. One diagram image
4. `package.json` and `package-lock.json`

Important: Remove the *node_modules/* and *images/* folders. We will use a similar set of images as in Lab 5 in an *images/* folder for grading.

If you need help, please feel free to contact the main TA for Assignment Two:

Mr. Xu Peng (pxu22@cse.cuhk.edu.hk or 1155187705@link.cuhk.edu.hk), SHB905, Saturday from 8 a.m. to 5 p.m.

Or the course instructor:

Dr Colin Tsang (colintsang@cuhk.edu.hk), SHB130, by appointment.

Remark: We welcome any questions, but we will not do debugging for you.

Late submission: A 30%-mark deduction will be applied for late submissions within three days. Late submissions more than three days will not be graded.

Code header for honesty declaration:

```
<!-- I declare that the lab work here submitted is original
except for source material explicitly acknowledged,
and that the same or closely related material has not been
previously submitted for another course.
I also acknowledge that I am aware of University policy and
regulations on honesty in academic work, and of the disciplinary
guidelines and procedures applicable to breaches of such
policy and regulations, as contained in the website.

University Guideline on Academic Honesty:
https://www.cuhk.edu.hk/policy/academichonesty/

Student Name : <your name>
Student ID : <your student ID>
Class/Section : <your class/section>
Date : <date> -->
```