CS385 Mobile Application Development 2023/2024

Lab Session #3 - 20th October 2023

You must upload your solutions to Moodle (The link is on the Moodle Section for this Lab) by 15:59 Tuesday 24th October 2023. Remember, you can upload partial solutions. You do not need to upload fully working solutions. Please name your files as .txt files when you are uploading to Moodle.

Lab Description

Please refer to CS385 Lecture 7 materials for guidance. It is strongly suggested that you use this material as guidance rather than using Google or StackOverflow.

If you need a template, you should use the React/Javascript TEMPLATE as provided in **`Lab2_Template.txt`** from the Moodle page.

On Moodle you will find a file called **airlines.js** – this contains 500 randomly generated flights in Europe. There are four properties: **flight** (corresponding to the flight number), **dept** (corresponding to the departure city or airport), **dest** (corresponding to the arrival city or airport), and **status** (indicating the current status of the flight). This is NOT real-time data but can be interpreted as being a snapshot of real-time data being drawn from a source such as a flight tracker API.

You must import this array of JSON objects into your application.

TASK 1 – using props to faciliate parent child communication.

In this task you are asked to use Lecture 7 as a basis for delivering a simple application providing the following:

- Import the array of JSON objects from the **airlines.js** file.
- Design and implement two components
  - **a parent component** which provides a TEXTBOX allowing the user to search the JSON array of flights
  - **a child component** which is responsible for searching the JSON array and displaying/rendering the results.
- Render all properties of the objects in the JSON array within the search results.
- The search functionality should search the **flight**, **dept**, and **dest** properties of the JSON objects. It should not search the **status** property.

When you are finished task 1 – please save this as Lab3Task1.txt – (this will be uploaded to Moodle later) you can then start working on task 2 for this lab.

TASK 2 – Conditional Rendering

You can use your code from Task 1 as the starting point for this task. We need to complete the following steps.

- Use conditional rendering in the Child component. Write four conditional rendering statements which will render the following based on the number of search results generated.
  - **0 results** – then you should render “No flight information available”
  - **1 result** – “One flight available”
  - **Between 2 and 20 (inclusive)** – “Several flights available”
  - **Greater than 20** – “A large number of search results – please consider narrowing your search”

TASK 3 – OPTIONAL – DO NOT UPLOAD THIS TASK TO MOODLE

If you want to try out some additional parent-child communication code – you can attempt the following example – **PLEASE DO NOT UPLOAD TASK 3 TO MOODLE.** This is optional and is not part of your Lab 2 assessment.

TASK – Change your code from Task 2 as follows:

- **Add a button to the Child component** – when this button is clicked – it will clear the search results. To ensure that the text box is cleared you might need to add another command to the **<input>** tag. For example – if you used searchTerm as your state variable then the **<input>** line for your text box would need to be something similar to

  ```
  <input onChange={onSearchFormChange} type="text"
  value={searchTerm} />
  ```

UPLOADING YOUR SOLUTIONS

- You can upload separate files for each task or a single file into Moodle. It is strongly advised that you simply copy your code from CodeSandBox.io into a notepad (or otherwise) text editor. Save your files as txt files. This is the simplest way to do this. DO NOT USE WORD DOCUMENTS. Programmers do not copy-paste code into MS Word Documents.