# Components

## Overview

In this lab you'll refactor the "library" web page from the previous lab, so that it uses components to render portions of the web page.

You can either choose to implement your components as class-based components or as functional components. We provide solutions for both techniques.

## Source folders

* ReactDev\Student\03-Components
* ReactDev\Solutions\03-Components

## Roadmap

Here's a brief summary of the exercises in this lab. More detailed instructions follow later in this lab document:

1. Familiarization with the 'solution' web pages
2. Getting started with the 'student' web page
3. Defining an ItemsList component
4. Creating the overall content for the web page
5. (If time permits)Defining a Table component

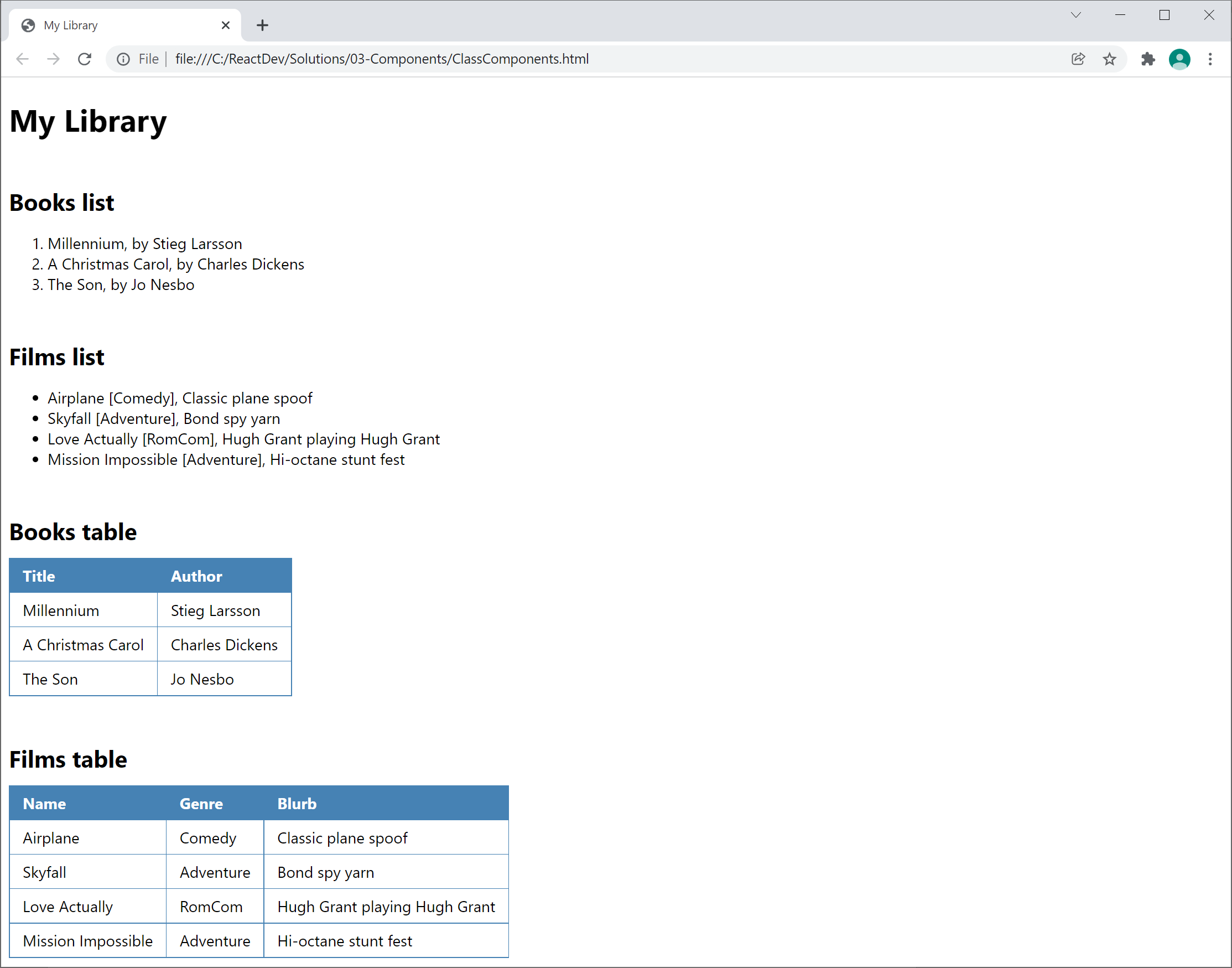
## Exercise 1: Familiarization with the solution web pages

Open either of the following web pages in a browser:

ReactDev\Solutions\03-Components\ClassComponents.html

ReactDev\Solutions\03-Components\FunctionalComponents.html

These two web pages are semantically equivalent; one defines class-based components, and the other defines functional components. Both have the same appearance:



Note the following points:

* The upper half of the web page displays books and films as HTML lists. You will implement this functionality in the "core" part of the lab.
* The lower half of the web page displays the books and films as HTML tables. You will implement this functionality in the "if time permits" part of the lab.

## Exercise 2: Getting started with the 'student' web page

Now go to the ***student*** folder and open the following web page in a text editor:

ReactDev\Student\03-Components\index.html

This is the same as the solution from the previous lab. Take a moment to reacquaint yourself with the code.

## Exercise 3: Defining an ItemsList component

Define an ItemsList component to render an array of objects as an HTML list, as follows:

* Receives 2 properties:
  + items - an array of objects to display
  + ordered - a bool indicating whether to display an ordered or unordered list
* Returns either an <ol> or <ul> accordingly, containing <li> child elements
* Uses the objects' toString() function to display data for each object in the list

Once you've defined the ItemsList component, you can delete the original code that declared the booklist and filmlist variables.

## Exercise 4: Creating the overall content for the web page

Near the bottom of the web page, locate the code that declares the library variable. Modify the call to React.createElement() as follows:

* Rather than passing booklist as a parameter, call React.createElement() to create an ItemsList component. Pass in the following properties to the component:
  + items: books
  + ordered: true
* Rather than passing filmlist as a parameter, call React.createElement() to create an ItemsList component. Pass in the following properties to the component:
  + items: films
  + ordered: false

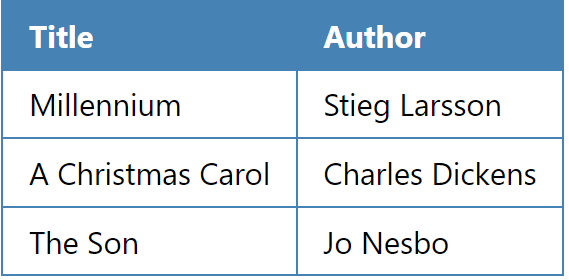
When you've made all these changes, open the web page in a browser and verify everything works correctly.

## Exercise 5 (if time permits): Defining a Table component

Define a Table component to render an array of objects as an HTML table, as follows:

* Receives a single property named items, an array of objects to display
* Returns a <table> with a separate row per object. Display each property value in a separate column.
* For added merit, it would be nice if the table has a heading to display the property names.

Here's a reminder of what a table looks like in the solution:



You'll also have to enhance the code at the bottom of the web page, where the library variable is declared. Enhance this code so that it also renders a Table component for the books, and another Table component for the films.