COMP 1537 – Assignment 5

Due: The week of November 21st, before the beginning of your scheduled class, for marks! This assignment is out of **40** – heavier because you are doing **much** more work!

This assignment must be done individually.

Objective

To build the basic architecture of a web app – minus the DB (i.e., data tier).

Overall Structure

For this assignment, you are going to develop the basic constructs of a web app. This will include a front-end as well as a back-end. For now, no DB. So far, you've been shown Node.js, Express.js (and how to route), the HTTP methods (GET, POST), and how to download modules with NPM. Now it's time to create something that demonstrates an applied use of them together.

You are going to create both a front-end, as well as a back-end (via Node.js). Your app will consist of:

1. One client-side JS script that will handle AJAX GET calls in the browser (external to the HTML), called client.js which is referenced from your HTML document
2. One server-side JavaScript script that will answer requests and return JSON as well as HTML code depending upon the path that is accessed; this is called assignment5.js and running on port 8000
3. One HTML document that for the main page
4. One or more CSS files that contains your style, referenced from your HTML document

Content

Pick a context for your basic web application. So for example, your app could display:

- Recipes for cooking
- Hockey team scores
- Movie listings for a theatre
- Courses for a program (e.g., CST)

Requirements

What you pick is up to you, but the following requirements must be met in order to receive full marks:

- Must use Node.js with Express.js
- Use the folder structure that has been provided in the examples – don't just dump files in the root
- Remove all test output on the server as well as in the browser
- You are creating your own content and not pulling content from another server; that is, your server application must be the one serving up your JSON and HTML
- Contain CSS styling that creates a layout that is responsive and that supports at least two resolutions
    - You can utilize float/clear, CSS grid (however you want to use it) and flex; you are free to decide which of these techniques that you wish to use
- A web template design that is appealing (see below for "appealing") and must be your own – do not use my template, do not use anyone else's, make your own
- Upon access to the page, two calls are made:
    - AJAX call to the server, where the server sends a snippet of HTML – which is either an HTML table or an HTML list (ordered or unordered)
    - AJAX call to the server, where the server sends a snippet of JSON
- Both paths must be accessed by the client-side and the client.js code must "consume" (i.e., load, and display) the content from each path
- Each server path is not from static mappings but with function callbacks associated with paths in Express.js
- The server AJAX calls will return files from the file system, one file is HTML, one file is JSON
    - Later we'll see how to instead return data from a database
- Both the HTML as well as JSON call will return what are equivalent to 10 records with five attributes
    - An example of this is a list of 10 courses where there are five attributes (e.g., course name, course number, number of credits, instructor, prerequisites)
- The HTML code snippet from the server can easily be inserted via DOM into the main page HTML document, however, the JSON will require you to generate HTML code in the client.js document and then perform the insert into the main page

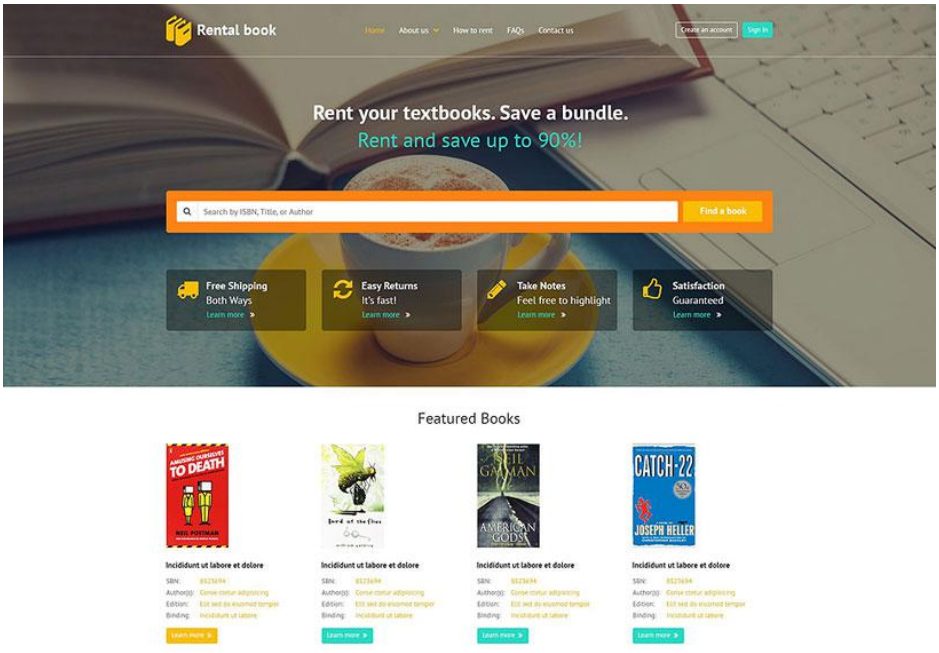
For now there is no authentication (i.e., session handling), and no database. That we will save for the next and final assignment.

And no using client-side frameworks for this assignment. You will still be using node, express, and any packages needed for the back end.

Appealing Web Designs

You may not be designers, but it doesn't mean that you cannot make an appealing web design. You've already learned how to take existing web templates/designs and recreate them. So looking at the following designs, please try and make something that looks similar or is of comparable quality. If you

are in doubt, ask your instructor for feedback on your design. If you have a web template from a previous assignment you can use that for this assignment, but it must be of the caliber of what you see in examples on the next page of this document.



Once you have completed, ensure that your document is valid. You code (HTML, CSS, JavaScript) must all be free of all syntax errors regardless of what validation tools you use (online, editor plugins, reading manually, etc.).

> **Note:** you have three weeks to complete this assignment. With that in mind, do not leave this until the last week or last couple of days to complete. Doing so would be an example of poor time management as an adult learner and will almost certainly results in a failing grade for this assignment. It is worth 40 marks versus only 10 or 20, like the other assignments.

As well, include a readme.txt file that follows this format:

```
[Student Name], [Student ID], [Set], [Date]


This assignment is [enter percent]% complete.


[explanation if not complete, what is working/not
working]
```

Figure 1: readme.txt

Submission

Once you have completed, ensure that your document is valid. It is suggested that you use the HTML 5 validator online at:

https://en.rakko.tools/tools/58/

Ways in which you may lose marks (but not limited to the following list):

- You don't validate your HTML/CSS
- Your JS has errors in it
- Links to files/resources are broken
- You left the content that I provided in the example in your submission (e.g., title element value, element ID values, any of my HTML or CSS

To ensure that you don't lose marks. Also use the CSS validator in Visual Studio Code.

Create a zip archive of any and all text files that are part of your assignment submission. Your HTML files will have the extension ".html", your CSS files will have the extension ".css", and your JavaScript files will have the extension ".js". If you have images, be sure to include them in this archive that you create.

Once you create your zip archive file, rename it to follow the format SurnameFirstnameCOMP1537Assignment5. My zip file would be FergusonArronCOMP1537Assignment5, for instance.

Finally, submit it to the learning hub in the folder labeled "assignment 5".