**Building a Basic Web App with Node.js**

**Due Date: Two weeks from now – demo in class (late results in zero)**

**Done in pairs (no groups larger than two or if there is an odd number, one group of three … only one though otherwise you’ll lose marks).**

Lab exercises to show your instructor for marks.

For this lab, you are going to get build a simple app with Node. 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 your web development skills.

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 JavaScript script that will handle AJAX calls in the browser (external to the HTML)
2. One server-side JavaScript script that will answer requests and return JSON and HTML code
3. One HTML document that contains your markup
4. One CSS file that contains your style (so external to the HTML)

## Other details

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)

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

* Return at least 20 records of data (10 in HTML, 10 in JSON)
* Contain CSS styling that creates a layout that is responsive
* A web template design that is appealing (see below for “appealing”)
* The data returned in HTML format must be in the form of either an ordered list, an unordered list, a table, or a list of paragraphs
* The data returned in JSON format must be then visualized on the client-side and turned into an ordered list, an unordered list, a table, or a list of paragraphs

Your server script will output one list but offer two formats: HTML and JSON. Like the example code (from ajax-app.js), you need to output two versions of your list of records: one in HTML and one in JSON. This means that your client-side code for your app must be able to receive HTML and place it inside of the layout as well as receive JSON and construct a part of your page using the data from your JSON.

This lab exercise will be weighted out of 10 since it’s fairly complex and requires two weeks’ worth of effort.

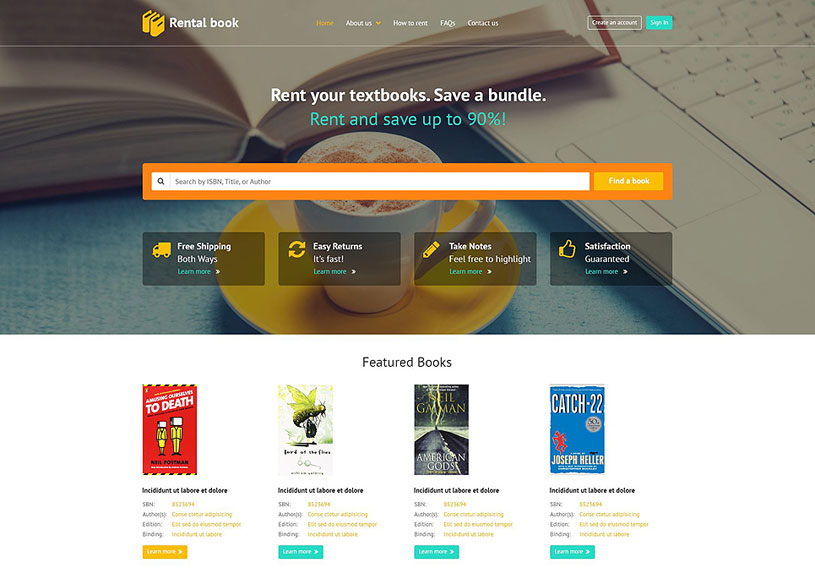
## Important

This document may change in the future, so please check back as I will update it as questions arise. If this document is update, a new version number will be posted and the updated/new sections will be highlighted in yellow for your attention.

## 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.







The web app design must be responsive and support at least three resolutions, utilize float/clear, use CSS filters (pick at least one), use a transform of some kind, and add at least three CSS animations in your design (e.g., hover, banner with animations – try not to make that too annoying).