

# COMP 2537 – Web Development 2

## Assignment 3

BCIT CST — Due: May 16<sup>th</sup>, 11:59 PM

Late submissions result in zero

*This is a project group submission (groups based upon your COMP 2800 course)*

### Introduction

You are going to practice making an app that allows for session handling for login/authentication.

### Details

Create an app (mid-to-high level of fidelity) that allows a user to log in. This means that you will need a DB and at least one record for a user to log in as. For simplicity purposes (and for streamlining the process), use the following for login:

User email: [arron.ferguson@bcit.ca](mailto:arron.ferguson@bcit.ca)

Password: admin

That way I don't have to guess. You will also utilize some form of templating. You can use EJS, Express custom solution, JSDOM/jQuery (like how I did), or anything else you wish to use. But you must have portions of your page that are coming from different sources. If you are not sure, ask me and I will help clear this up. The responsibility is on you to clear what you are using.

Requirements are:

- Create the data/DB as how I have done in the “templating-and-sessions” example (i.e., checks if the DB exists, if not, it creates it, if the table doesn’t exist, creates it, and if there’s already one record there, it doesn’t create any more – this is for testing purposes)
- As mentioned, mid-to-high level fidelity – no placeholders and not lorem ipsum (use your project app as your content, for example)
- Two pages: login screen, main page/dashboard – populate with whatever your COMP 2800 app has if you need content; don’t leave this blank!
- You must use Node.js and session store for your authentication and only store one or two pieces of information in the session
- Users must be able to log in and log out
- If users try to access their dashboard/main page, redirect to the login page – this happens from the server side, not the client
- Login must have via AJAX so you can send a message (again, like the example)
- You must have at least two parts/components on the page that are controlled by some means of templating

## Submission Requirements

Zip up your submission and upload to the COMP 2537 learning hub “Assignment 3” folder, one submission from each group.

The uploaded app must contain all of your names, full names and student IDs that show up in the main page. Failure to do so will result in not getting marked for work done. If I have to do a marks change after the course is over because of this, the entire group will get docked marks.

## Grading Scheme

Your assignment will be marked out of 20:

- 6 marks for templating
- 2 marks for login logic & display
- 2 marks for logout & display
- 2 marks for dashboard template
- 6 marks for express session handling
- 2 marks for your DB code

You will lose marks for:

- **Not using strict**
- Invalid code of any type
- Runtime errors
- Not following the requirements of the assignment
- Using variable names or file names that are very clearly example code or example files (e.g., lab7-starter.js)
- Content that is boilerplate (e.g., dummyimage images, lorem ipsum)

If you have questions pertaining to any design choices, please see your lab instructor or lecture instructor.

Submit naming format:

Your zip file containing your source code files in a zip named COMP2537-assignment3-<lastname>-<firstname>.zip (only one zip file for all source code) and only one submission per project group