**Assignment 5: Final**

Submission checklist:

* Tell me who you worked with, or confirm you worked on your own
* Include a report, especially details of your image work
* Estimate your grades (except depth)
* Avoid submitting your git repo (move to the folder above)
* Avoid submitting your node\_modules (move to the folder above)

The main assessment for the unit will be a web site driven by a web server. The assessment is the same for COMS32500 and COMSM0104, except that COMSM0104 students (and any non computer scientists on COMS32500) will not be expected to be quite as proficient in direct JavaScript programming, and instead will be assessed a little more on integrating existing tools, frameworks, library modules or scripts. As with any 10 credit unit, the total amount of time spent is supposed to be about 50 hours.

**Pairs:** I expect most of you to work in pairs. I will leave you to pair up as you like. The idea is that the two of you should work together and both learn everything, preferably in the pair programming style, not just divide up the work. If one of you is to be the expert in a particular area, teach the other about it. Make sure that you both put in equal effort, e.g. each both taking the lead on some topics, so that you are happy to receive the same mark. If a pair breaks up, or the work becomes unbalanced, let me know and I will do what I can to sort it out. If you would prefer to work on your own, that is fine. You are still expected to cover the full range of skills, but your work is not expected to be as extensive or polished.

**Topic:** The idea is to build a web site of your own choosing, building an interesting site structure, and adding a server to deliver pages and handle a database. Example sites might be: shopping for sports gear, reviewing cameras, discussing politics, supporting a particular charity or organisation, teaching astronomy, uploading and showing off photos, social media, playing games. However, you need to bear in mind that you will need to demonstrate your grasp of a wide variety of different technologies. So, you need a topic with scope for expansion. If you choose a narrow topic, you may need to find extensions, or even effectively create an unrelated subsite, to show off some of the skills. It doesn't necessarily matter if your site doesn't make complete coherent sense in the end.

**Requirements** You must use standard HTML, CSS, JavaScript, PNG, SVG and other integrated client-side technologies, following the advice from the lectures. Your server must be Node-based, and your database must be embedded, preferably SQLite. Your design can be desktop-first or mobile-first, and you can use client-side or server-side techniques for creating dynamic pages. You can write everything yourself, or use any existing JavaScript-based frameworks or libraries or modules or scripts that you like. In general, if you write everything yourself, you will be expected to learn more about the underlying mechanisms, and if you use existing frameworks, you will be expected to get further. You should use open source tools as much as possible.

**Resources** You can get technical or creative ideas from anywhere on the Web, or use any existing images or HTML/CSS fragments or JavaScript frameworks or libraries or modules or scripts. But you must (a) check any restrictions (b) acknowledge them in your report (c) adapt them to follow the standards and advice in the lectures where necessary and (d) explain your added value (e.g. "I just copied it and learnt how to use it" or "I changed it a little" or "I understood it fully and re-wrote it").

**Assessment**

The assessment will be partly on breadth and partly on depth. The breadth will be marked very simply (almost like ticking boxes) under a number of headings for the various skills. The depth will be a single mark, which represents your overall level of sophistication, hard work, polish, technical achievements, creativity and so on, and this can be demonstrated by work in any direction or any combination of areas. So your best strategy is *first* to make sure you have covered all the skills to some extent, and *then* to put extra effort in any direction you like. The headings are:

<ul>

<li>X for HTML</li>

<li>X for CSS</li>

<li>X for JS</li>

<li>X for PNG</li>

<li>X for SVG</li>

<li>X for Server</li>

<li>X for Database</li>

<li>X for Dynamic pages</li>

<li>X for Depth (out of 40)</li>

</ul>

*You can estimate your own marks.* To form your estimate, you might like to copy the bullet point list above into your report web page and change the letter against each heading:

* ? means you don't want to estimate
* X means you have done nothing in this area
* D means you started but got stuck or ran out of time
* C means you have done basic work
* B means you have done some solid work
* A means you have done sophisticated or extensive work

In order to end up with a mark in line with the [standard scale](https://www.cs.bris.ac.uk/handbook/2009-0/assessment.html), I will give each breadth heading a mark out of 10, with X=0, D=4, C=5, B=6 and A=7 (assuming that I agree with your grade). Marks over 7 are not usually awarded; excellent work is usually rewarded via the depth mark. The depth mark, which you might or might not want to estimate, will be out of 40 (with A ≅ 28+, B ≅ 24..27 etc). So the total will be out of 120, which will of course end up being reduced to a percentage.

**Details**

Here are some details to help you estimate your grade under each heading. The descriptions are just examples, so you will need to apply some common sense.

* HTML: I expect most of you to claim A, because you will have gained a lot of experience by the end
  + C means you've got a page or two to work
  + B means you've used XHTML delivery, or a validator, to make sure your pages are correct
  + A means you have investigated a variety of different issues and gained a general high level of confidence with the structure of HTML pages (or with generating HTML via a framework)
* CSS: again, I expect many of you to claim A
  + C means you've got some style to work
  + B means you've made sure there are no style tags or style attributes in your HTML pages, and you've gained some experience with stylesheets
  + A means you have investigated a variety of different issues and gained a general high level of confidence with CSS style (or with generating CSS via a framework)
* JS: this is client-side JavaScript for effects or animation or interaction, including use of client-side frameworks, but excluding aspects which are to do with dynamic page construction
  + C means you've got some scripting to work, by writing a small amount of JavaScript, or by slightly adapting a script you found, or by closely following a tutorial
  + B means you've written a substantial script yourself, or written a number of script functions with different issues involved, or gained a medium amount of experience with client-side frameworks
  + A means you have gained a high level of understanding of how client-side JavaScript works, or a high level of expertise in using client-side frameworks
* PNG: this is working with bitmap graphics in Gimp or Krita
  + C means you've sorted out basic skills such as converting images to PNG, cropping away unwanted edges, changing resoluation
  + B means you've gained experience with some basic tools such as using filters or changing colours or combining existing images or creating simple shapes or filling
  + A means you have gained experience with some more sophisticated tools such as handling layers and transparency, or airbrushing or creating original artwork
* SVG: this is working with vector graphics in Inkscape
  + C means you've created a basic drawing in Inkscape, probably by copying something else
  + B means you've gained experience with some of Inkscape's features such as shape tools, freehand drawing, simplification
  + A means you have gained a higher level of experience, e.g. with path editing, grouping, transformations, gradients, patterns, etc., or put a lot of effort into vector artwork
* Server: this is creating or adapting a server, either programming it yourself or using express and its add-ons
  + C means you've created a server by minimally adapting the one provided, or closely following a tutorial to set up express
  + B means you've dealt with things like port numbers, URL validation, content negotiation for old browsers, sending redirections to browsers, handling UTF-8
  + A means you've dealt with things like https and certificates, or web sockets, or cloud hosting, or security issues beyond URL validation, or auto-testing, or cookies, or running under reduced privilege
* Database: this is integrating a database with your server
  + C means you've manually created a database, and then extracted data from it in your server
  + B means you've managed to update or insert data as well as extract it, and you've got the hang of callbacks for getting things to happen in the right order
  + A means you've gained a lot of experience with SQL, or you've put a lot of effort into organising database access (e.g. into a separate server-side module) or you've put a lot of effort into database design or details of handling your data
* Dynamic pages: this is *either* inserting data into templates on the server side and delivering dynamic pages *or* requesting data from the server and inserting into existing pages on the client side
  + C means you've created a simple dynamic delivery system or you have used a framework in a simple way
  + B means you've organised dynamic delivery in a more sophisticated way, either doing more of your own programming or using more features of a framework
  + A means you've put in a lot of programming effort or become very fluent in using your chosen framework

Since the work is so open ended, there is room for substituting one thing for another. For example:

* animation done using CSS can count a little towards the JS heading
* using canvas, including its vector features, can take the place of conventional SVG work
* something clever with web sockets for a chat or game application might take the place of conventional dynamic page delivery

Understanding something properly counts for more than blindly copying something or blindly following a tutorial. If you have used a framework, making it do what you want counts for more than just producing a site that looks like most of the other sites that use that framework. You may be better able to judge some of these issues than me.

**Submission**

You should submit your site as a zip file (*without* asking SAFE to unzip it). **Beware:** on Windows, zip can't cope with long names, and sometimes produces unreadable zip files, so install 7-zip and use that.

I will install any Node modules needed, but otherwise your server should be ready to run (even if it is just the basic one provided). Your zip file should contain everything your server needs such as downloaded images that it serves, but should **not** include your node\_modules folder or your git repository, if you are using one.

You should also submit a report as a web page or a pdf. There are no marks for the report as such, but it is vital that you provide one to help with the marking. For example, under the PNG heading, if all I can see is an image, I will have **no idea** what you actually did to produce it, or if you just downloaded it from somewhere, so if you don't tell me in the report, I will have to give a mark of zero.

Your report should give your estimated grades under each heading, and then list things you've done under each heading to justify the estimate. For depth, you might want to write something longer which explains your overall aims and design, anything you are particularly proud of, anything I might miss when I am browsing your site, and anything which took a substantial amount of time and effort but didn't make it into the site.