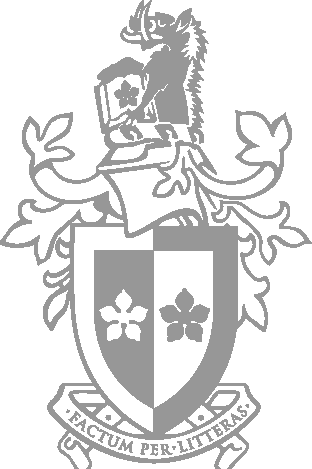
Faculty of Science, Engineering and Technology





**Interface Design and Development**

Credit Task 4: Consuming External Services

**Overview**

In lab 3, models are initialised using ngInit. In Lab 4, controller is introduced and is preferred to initialise models. However data are still hard coded in the JavaScript app. In this task, the data is to be loaded from a table using the controller. You will later manipulate the data and plot a chart with the manipulated data using HighChart.js library.

.

**Purpose:** Learn how to retrieve data from a text file using AngularJS. Demonstrate that you can perform simple data manipulation i.e. aggregation from a set of data given in JSON format and produce a chart from the manipulated data.

**Task:** Use $http.get method to retrieve data from a text file to generate a table. After that, count number of unit available based on its type and plot the number into a dynamic chart using HighChartJS library.

**Time:** This task should be completed in your lab class and submitted for feedback before the start of week 10.

**Resources:**

* + - Lecture notes #4 and #8
    - <https://www.highcharts.com>

***Submission Details***

You must submit the following files to Canvas:

* **Web page source code (units4.html).**
* **Units controller source code (appunits4.js).**
* **Screenshot of the table webpage**

Make sure that your task has the following in your submission:

* Code must be HTML5 standard and observe indentation.
* **Your webpage uses the AngularJS, Bootstrap framework and HighChart library appropriately**.



Generating Table Using $http

**Instructions**

Replace the ngInit (in html) / $scope.units (in js) used in credit task 3 with a controller that uses the $http.get method.

1. Copy units3.html from lab 6 to units4.html for lab 8
2. Open Brackets and open units4.html from your lab08 directory.
3. Start the controller found in lecture 08.
4. Implement a web application with the following logic:
   1. Using the following data, create a text file and encode the data in JSON format and save it in the data directory

|  |  |  |  |
| --- | --- | --- | --- |
| Unit codes | Units | Credit points | Type |
| ICT10001 | Problem Solving with ICT | 12.5 | Core |
| COS10005 | Web Development | 12.5 | Core |
| INF10003 | Introduction to Business Information Systems | 12.5 | Core |
| INF10002 | Database Analysis and Design | 12.5 | Core |
| COS10009 | Introduction to Programming | 12.5 | Core |
| INF30029 | Information Technology Project Management | 12.5 | Core |
| ICT30005 | Professional Issues in Information Technology | 12.5 | Core |
| ICT30001 | Information Technology Project | 12.5 | Core |
| COS20001 | User-Centred Design | 12.5 | Software Development |
| TNE10005 | Network Administration | 12.5 | Software Development |
| … | … | … | … |
| COS30043 | Interface Design and Development | 12.5 | Software Development |
| COS30017 | Software Development for Mobile Devices | 12.5 | Software Development |
| INF20012 | Enterprise Systems | 12.5 | Systems Analysis |
| ACC10007 | Financial Information for Decision Making | 12.5 | Systems Analysis |
| INF20003 | Requirements Analysis and Modelling | 12.5 | Systems Analysis |
| ACC20014 | Management Decision Making | 12.5 | Systems Analysis |
| INF30005 | Business Process Management | 12.5 | Systems Analysis |
| INF30003 | Business Information Systems Analysis | 12.5 | Systems Analysis |
| INF30020 | Information Systems Risk and Security | 12.5 | Systems Analysis |
| INF30001 | Systems Acquisition & Implementation Management | 12.5 | Systems Analysis |

Table 1: Unit Information

* 1. Retrieve the data using the $http.get method in the controller

1. Your web app should now be complete. Make sure you test it on the browser to make sure that it works as you expect. Ensure your codes are well commented and styled appropriately.

My Unit App with Charting Tool

# Instructions

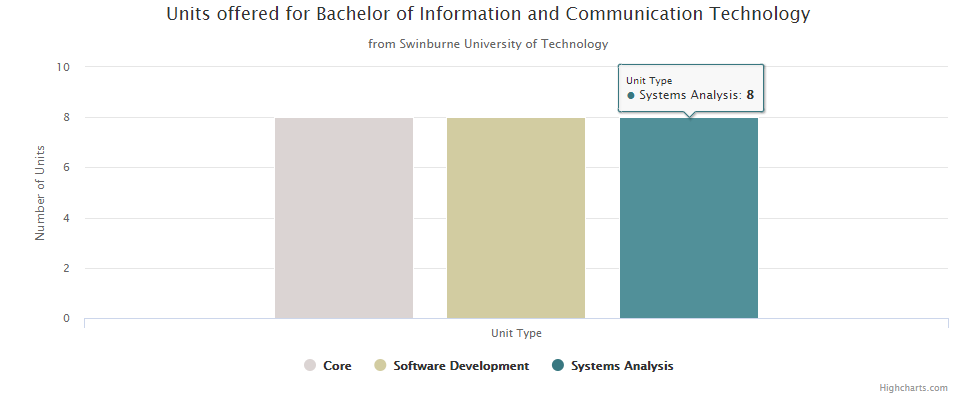
1. Continue to work on solution from above task (units4.html and appunits4.js) in your lab08 directory.
2. Download HighChart.JS library from <https://www.highcharts.com/download>. Copy the “code” folder and paste into your “framework” folder. Rename it as highchart. You may also use the demo files to do this task.

**Important**: This is very important step to ensure your source and library files are well organized.

1. Open Brackets and open the file as units4.html. Define a <div> with id=”myfirstchart” above the unit table. Then import “highchart.js” from “highchart” folder into your webpage.
2. Open appunits4.js, perform aggregation function (count) to count number of unit available by type from the data.json. You may define your own aggregation function or import library from online resources (make sure you include the reference).

**Tip**: You can define a few functions which take an Array of JSON objects, manipulate and parse into HighChart compatible format.

1. Develop a **column** chart based on manipulated data. The chart shall contain appropriate title, a subtitle with text “from Swinburne University of Technology”, title for both x axis and y axis. Take a screenshot of your chart which looks something like this:



1. Customize your chart to make it more interesting i.e. changing its color, enable data label/value, enable export features from highchart library etc. (write down those add-on below your chart in bullet list)

**Tip**: You can make lots of customizations by referencing the [HighChart documentation](https://www.highcharts.com/docs) page

1. Take a screenshot of your chart.
2. Your web app should now be complete. Make sure you test it on the browser to make sure that it works as you expect. Ensure your codes are well commented and styled appropriately.