Faculty of Science, Engineering and Technology

Interface Design and Development

Credit Task 1: AngularJS- Conditional, Loop & Filters

Overview

The loop directive in AngularJS allows developers to create programs that dynamically generates HTML elements using an array of data objects. In this task you will create a small web application that will list down a list of units in a table and allow the user to filter by unit code and unit description. On top of that, you will also create a small phone registration web application where some subsequent questions depend on the answer of the previous question.

Purpose: Learn to use loop and filter directives to respond to user actions.

Task: Create a web app that list units and support a search filter.

Time: This task should be completed in your lab class and submitted for feedback before the start of week 5.

Resources:

- Lecture notes #3
- AngularJS https://angularjs.org/

Submission Details

You must submit the following files to Blackboard:

- Unit Search & Registration Form source code (units.html & register.html).
- Screenshot of the web app.

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

- The Unit Search & Registration Form web application is HTML5 compliant.
- Demonstrates understanding in using the AngularJS framework.
- Demonstrates use of AngularJS expression, conditional and conditional directives.

SWINBURNE UNIVERSITY OF TECHNOLOGY

Instructions

Component 1: Unit Search Web App

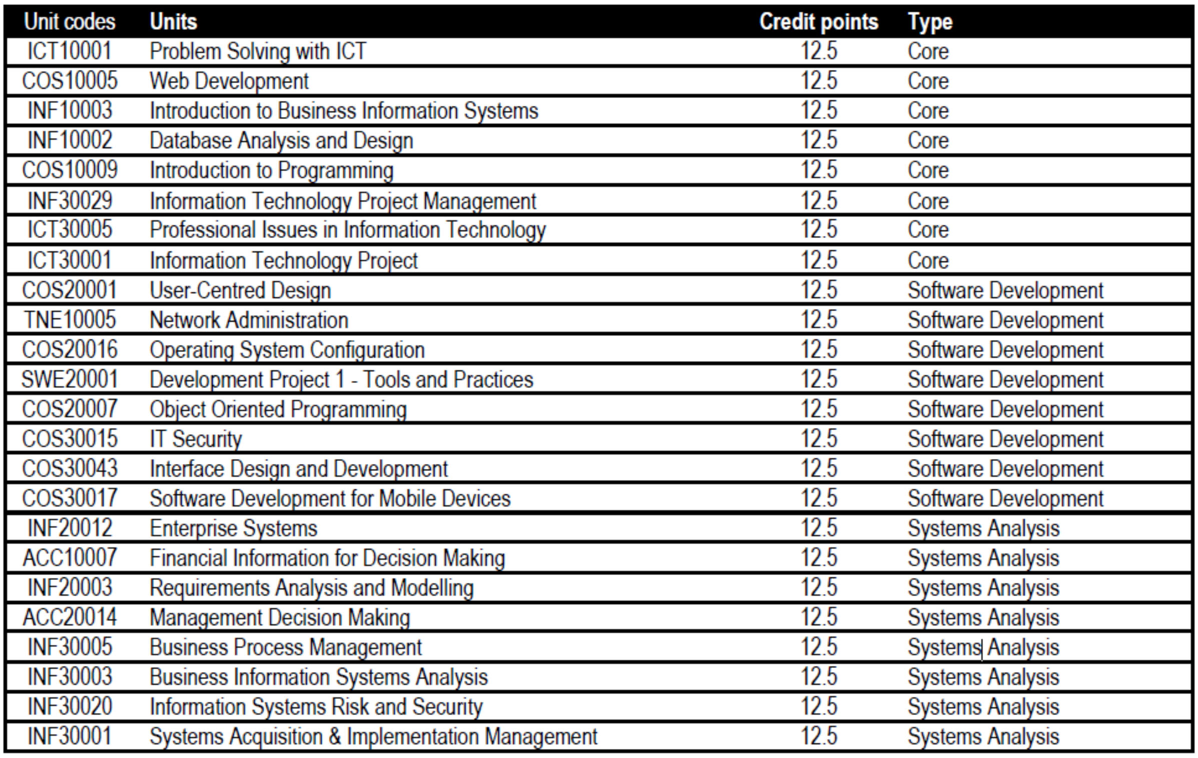
Loops allow one to programmatically generate a list or table based on a given data, instead of having to hard code each data item. This requires the data set to be downloaded to the client, which may add to the loading time of the web application.

While the list or table can also be programmatically generated in the server and send to the client as HTML, adding search and filter client-side user interactivity will be difficult to implement.

1. Open Brackets and save the blank file as units.html in your lab03 directory.

2. Start the web application code with the template for AngularJS found in lecture 03.

3. Implement a web application with the following logic:
   - Initialise the array of objects with the following unit information



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
COS20016	Operating System Configuration	12.5	Software Development
SWE20001	Development Project 1 - Tools and Practices	12.5	Software Development
COS20007	Object Oriented Programming	12.5	Software Development
COS30015	IT Security	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

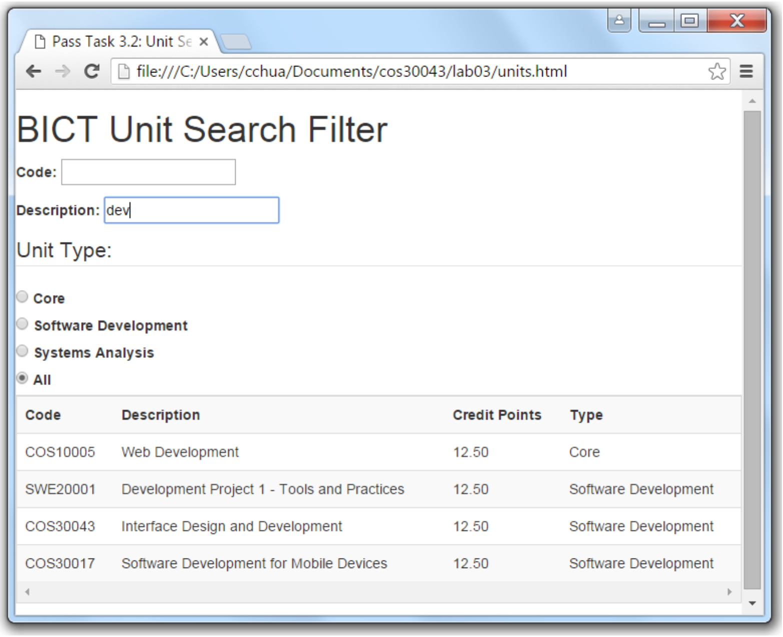
   - It reads 2 text inputs, namely: unit code and unit description.
   - It has a radio input to select unit type, namely: core, software development, systems analysis and all
   - List the units in a table ordered by unit description with unit code, unit description and type as filter. Filter will not be case sensitive, and credit point number must have 2 decimal places.

```
Web App: units.html

Uses: AngularJS
------------------------------
--- Model:
- obj.code (stores the unit code)
- obj.desc (stores the unit description)
- obj.type (specifies the unit type)
------------------------------
--- Steps:
1: Initialise the array of objects
2: Assign obj.code and/or obj.desc using ng-model with
    the prompt: 'Search filter:'
3: Create a radio input obj.type with the options
    'core', 'software development', 'systems analysis'
    or 'all' units
4: Use ng-repeat to display the table
```

Tip: Use the table class from the Bootstrap framework.

```
<div class="table-responsive">
  <table class="table table-striped table-hover">
     :
  </table>
</div>
```



BICT Unit Search Filter

Code:

Description: dev

Unit Type:

○ Core
○ Software Development
○ Systems Analysis
● All

Code	Description	Credit Points	Type
COS10005	Web Development	12.50	Core
SWE20001	Development Project 1 - Tools and Practices	12.50	Software Development
COS30043	Interface Design and Development	12.50	Software Development
COS30017	Software Development for Mobile Devices	12.50	Software Development

Figure 1: Screenshot of the web app with Bootstrap mark up

Now that the task is complete you can submit it for assessment, which will help prepare it for your portfolio.

Component 2: Registration Form Web App

Instructions

Users appreciate personalised experiences. Using AngularJS, we can create a registration form that adapts subsequent questions based on previous answers. It can also provide feedback in terms of progress. In this task, we implement a simple form where some input choices are dependent on the previous answer.

1. Open Brackets and save the blank file as register.html in your lab03 directory.

2. Start the web application code with the template for AngularJS found in lecture 03.

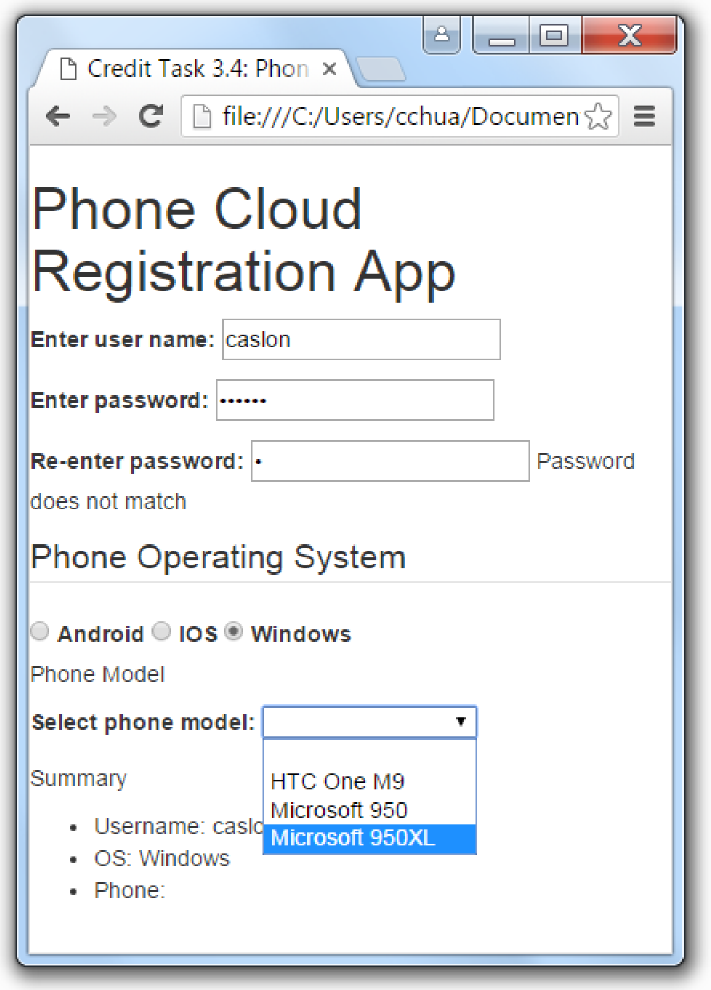
3. Implement a web application with the following logic:
   - A field for a username
   - A field for a password
   - A field for the user to re-enter the password. A message will appear beside the input box if it does not match the original password
   - Using a radio button input, prompt the user to select the mobile operating system he/she is using, namely android, ios or windows. Default is Android when the app is loaded.
   - Populate the selection options for a phone model drop down box input using a loop directive that is based on the selected mobile OS. The following is the list of phone models based on OS.

Android	IOS	Windows
HTC 8X	IPhone 6	HTC One M9
Samsung Galaxy 7	IPhone 6s	Microsoft 950
Sony Xperia Z5	IPhone 6s Plus	Microsoft 950XL

Table 1: Phone Information

   - Display a summary list that shows the username, phone OS and phone model.

```
Web App: register.html
________________
Uses: AngularJS
--------------------------------
- Model:
- input variables
- phone information list
--------------------------------
- Steps:
1: prompt for user name
2: prompt for password
3: prompt to re-enter password
4:   ng-show or ng-if – display if different
5: initialise radio default and prompt for radio input
6: initialise options and generate drop down input
7: generate summary list
```



Phone Cloud Registration App

Enter user name: caslon

Enter password: ••••••

Re-enter password: • Password does not match

Phone Operating System

○ **Android** ○ **IOS** ● **Windows**

Phone Model

Select phone model: ▾

HTC One M9
Microsoft 950
Microsoft 950XL

Summary

- Username: caslo
- OS: Windows
- Phone:

Figure 1: Screenshot of the web app with no Bootstrap mark up

Tip: Name attribute may appear optional in HTML input element as there is no server processing, however it is a requirement for radio input type.

Note: The select options must be populated using ng-repeat.

Hint: Use the filter for the ng-repeat. Do not forget to use object variable in the radio input.

Now that the task is complete you can submit it for assessment, which will help prepare it for your portfolio.