



# Assessment 3

## Server-Side Web Programming

CSE2ICX Internet Client Engineering

# Before you begin

## Objectives

This is an individual assignment. Students are not permitted to work in a group when writing this assignment.

## Copying and Plagiarism

This is an individual assignment. Students are not permitted to work in a group when writing this assignment. Plagiarism is the submission of another person's work in a manner that gives the impression that the work is their own. La Trobe University treats plagiarism seriously. When detected, penalties are strictly imposed.

Further information can be found on <https://www.latrobe.edu.au/students/admin/academic-integrity>

## Submission Guidelines

Your assignment submission should be typed, not written/drawn by hand.

Submit the electronic copy of your assignment through the subject LMS.

Submission after the deadline will incur a penalty of 5% of the available assignment mark per day capped at 5 days. No assignment will be accepted after 5 days. If you have encountered difficulties that lead to late submission or no submission, you should apply for special consideration.

## Contents

Server-Side Web Programming .....	1
Contents .....	3
Background .....	4
Task 1: Home (index.aspx) .....	4
Task 2: Add record (add.aspx) .....	4
Task 3: Invoice (invoice.aspx and print.aspx) .....	5
invoice.aspx page: .....	5
print.aspx page: .....	6
Task 4: Stocktake (stocktake.aspx) .....	7
Task 5: Chart (chart.aspx) .....	8
Task 6: Organisation .....	8
(1) Programming language: .....	8
(2) Third-party components: .....	8
(3) Images: .....	8
(4) Integrated development environment: .....	8
(5) Template: .....	8
Task 7: Layout .....	8
Task 8: Navigation .....	9
Task 9: Presentation .....	9
Task 10: Code .....	9
Submission .....	10
Assessment marking criteria .....	10
Appendix A Plant List .....	13

## Background

You are working in a development role at the organisation 'Website Implementation Perfection' (WIP). One of WIP's clients is an Australian plant nursery named 'Good Gums'. Good Gums already have a client facing website, but they are now interested in developing an internal site that will assist in the management of the business. Your manager has asked if you might be able to put together a fully functioning server-side prototype based on the requirements outlined below.

The assessment is broken up into nine separate tasks

- Tasks 1-5 describe the functional requirements of the site, i.e. what information each page on the site should show, and what the pages will do.
- Tasks 6-10 are the non-functional requirements of the site. These are the high standards that WIP want to have for all the sites they develop.

It is recommended that you read through all the tasks prior to commencing work.

### Task 1: Home (index.aspx)

- The home page is the entry point for the website. As this is the first page that will be viewed by users it should be designed in an attractive manner.
- The home page should include a GridView control that will be used to display a list of customer purchases. The purchase data will include the following data:
  - ID
  - Full name
  - Email address
  - Product Name
  - Unit Price
  - Quantity
- Data for the GridView will be taken from an XML data file and populated into the GridView using data binding – a sample data file can be downloaded from where you obtained this assignment document. You must also demonstrate your familiarity in working with the data file format by adding extra records to the data.
- The home page should include a 'Label' control. The 'Label' control should display the number of 'Large Orders', which is defined as any order where the customer has ordered 10 or more plants. Note the following:
  - The Label only needs to display how many 'Large Orders' exist – you do not need to provide any extra info.
  - Make sure that you include some 'larger' orders in your sample XML data.

### Task 2: Add record (add.aspx)

This page is used to add details of new customer orders.

- The page should contain a form that allows the user to make changes to the XML data file by adding records one at a time.
- An appropriate form field should be used for each separate field of the record.
- There is no need for the user to enter the ID field when adding records (this should be automatically determined depending on other ID fields). The ID field should be unique for each record.



- A 'Submit' button should be used to write changes to the XML data file.
- A 'Label' control should be used to display a message when the XML data file has been successfully updated.

### Task 3: Invoice (invoice.aspx and print.aspx)

This feature is intended to allow staff to prepare a printable invoice. It is divided into two pages

- invoice.aspx
- print.aspx

*Note that it is not expected that this page be data bound to the XML file used in previous tasks.*

#### invoice.aspx page:

This page collects details about the invoice items. The form is designed to accept 3 rows of data for the invoice. A mock-up of the interface is below:

Full name of customer:			
<input type="text"/>			
Item name:	Unit price:	Quantity:	
<input type="text"/>	<input type="text" value="0.00"/>	<input type="text" value="0"/>	
<input type="text"/>	<input type="text" value="0.00"/>	<input type="text" value="0"/>	
<input type="text"/>	<input type="text" value="0.00"/>	<input type="text" value="0"/>	
		<input type="button" value="Submit"/>	

Some notes about the interface:

- **Full name of Customer:** Must be no longer than 50 characters to be valid. Required.
- **Item name:** Must be no longer than 100 characters to be valid.
- **Unit price** and **quantity:** Text boxes that must be validated to receive non-negative numbers only. A maximum of "1000.00" is allowed for unit price and a maximum of "100" is allowed for quantity.
- **Submit:** Validates the form and submits the form if valid. User is taken to the print.aspx page after submission.

*The interface must also have appropriate validation controls and a ValidationSummary control at the end. Validation for individual inputs will be sufficient. A mock-up of the interface showing these is shown in the next page:*

Full name of customer:			*
<input type="text"/>			
Item name:	Unit price:	Quantity:	
<input type="text"/>	<input type="text" value="a"/>	<input type="text" value="0"/>	*
<input type="text"/>	<input type="text" value="0.00"/>	<input type="text" value="1.5"/>	*
<input type="text"/>	<input type="text" value="-1"/>	<input type="text" value="0"/>	*
		<input type="button" value="Submit"/>	

You must correct the following errors:

- Full name is required.
- Unit price (row 1) must be currency between 0.00 and 1000.00.
- Quantity (row 2) must be an integer between 0 and 100.
- Unit price (row 3) must be currency between 0.00 and 1000.00.

#### print.aspx page:

This is the 'destination' page after the form is submitted.

- This page will be a print-friendly version of the invoice form with simple labels and totals. All non-essential header, navigation and footer page components are omitted for print-friendliness.
- The invoice must include the name and contact details of the organisation at the top. A mock-up of the remaining output format is below:

Invoice for: Joe Bloggs

Date issued: 25/09/2020

Item:	Unit price:	Quantity:	Subtotal:
name of first item	\$700.00	1	\$700.00
name of second item	\$550.00	2	\$1,100.00
—	\$0.00	0	\$0.00
Total:			\$1,800.00

Terms: Invoice is due to be paid in full in 14 days.

Due date: 09/10/2020

## Task 4: Stocktake (stocktake.aspx)

This page allows administrators to take a general stock inventory for any day-to-day items related to the organisation. It is expected that you will use examples from 'Good Gums' range of products. Refer to the assessment appendix for these examples.

The stocktake form will allow the user to enter up to three rows of items. The form will be "pre-loaded" with dummy data to *simulate* taking the data from a data source.

The stocktake form also has a text input for the 'reorder level'. The reorder level is the level that the stock must be replenished to. Only stock that has dipped below the reorder level needs to be replenished back to the reorder level.

A mock-up of the interface is below:

Reorder level: - 10 +				
Item name:	Unit price:	Stock quantity:	Reorder quantity:	Subtotal:
placeholder for item 1	5.50	2	8	44.00
placeholder for item 2	16.00	7	3	48.00
placeholder for item 3	29.90	15	0	0.00
Calculate			Total:	92.00

Some notes about the interface:

- **Reorder level:** Read only. Default 10. Modified with "-" and "+" buttons. Range is 0 to 1000. Additionally performs the function of the "Calculate" button (see below).
- **Item name:** Must be no longer than 100 characters to be valid.
- **Stock quantity** and **unit price:** Required. Text boxes that must be validated to receive non-negative numbers only. A maximum of '1000.00' is allowed for unit price and a maximum of '100' is allowed for stock quantity. May be set to zero to indicate unused rows.
- **Reorder quantity, subtotal** and **total:** Derived data in labels.
- **Calculate:** Validates the form and recomputes calculations (if valid).

The interface must also have appropriate validation controls and a ValidationSummary control at the end. A mock-up of the interface showing these is below:

Reorder level: - 10 +				
Item name:	Stock quantity:	Unit price:	Reorder quantity:	Subtotal:
placeholder for item 1		5.50	8	\$44.00 *
placeholder for item 2	7	1001.00	3	\$48.00 *
placeholder for item 3	a	29.90	0	\$0.00 *
Calculate			Total:	\$92.00

You must correct the following errors:

- Stock quantity (row 1) is required.
- Unit price (row 2) must be currency between 0.00 and 1000.00.
- Stock quantity (row 3) must be an integer between 0 and 100.

## Task 5: Chart (chart.aspx)

This page includes an ASP.NET Chart control to model data for the organisation with the Chart Type is set to Line. The data will model the total earnings for last 5 years.

You only need to populate the chart with static data. i.e. You are not expected to perform any data binding. It does not matter if your data does not automatically update to remain consistent with any other page in the site.

The chart must be given attention to detail for its presentation, including all components that you may expect for this type of chart: title, axis labels, data labels.

## Task 6: Organisation

The following organisational aspects must be put in place for your solution:

### (1) Programming language:

The website will be a server-side website using ASP.NET for the web forms and C# for the code-behind functionality. Some HTML and CSS may also be used where appropriate.

### (2) Third-party components:

The website will be hand-coded by yourself. Therefore, you may not include any third-party components except for those mentioned in the assignment.

### (3) Images:

Even though this internal website is not public-facing it is still expected that you would make use of some images as part of your website design. Store the images in a subfolder called images". Note that the website cannot use copyrighted images. To avoid this problem, you had been provided with the necessary images of the different plants sold by Good Gums and that of its employees. The images are available for download as a separate zip file within the assessment activity, where you downloaded this assessment file. Note: you are not allowed to use these images anywhere other than for this assessment

### (4) Integrated development environment:

The website must be built using Visual Studio 2017.

### (5) Template:

The website will be built from scratch beginning with an empty folder. Follow these steps:

- Create a new folder called **xxx\_cse2icx\_assignment3**, where xxx is your student ID.
- Open Visual Studio 2017. Select "File" -> "Open" -> "Website..." and select the new folder.
- Select "File" -> "Close Solution" immediately. This will prompt you to save a solution file (\*.sln suffix). This must be saved in your new folder. Close Visual Studio 2017.
- Double-click on the solution file to open Visual Studio 2017 and your solution again. Continue to use this method for opening your solution going forward for convenience.
- You are now ready to go. Add your project assets next.

## Task 7: Layout

The body of the website pages must be organised to include distinct header, navigation, main and footer semantic tags with the following content:

- Header: Site name and branding.
- Navigation: Navigation bar.
- Main: This is the main content of each page.
- Footer: Copyright statement.



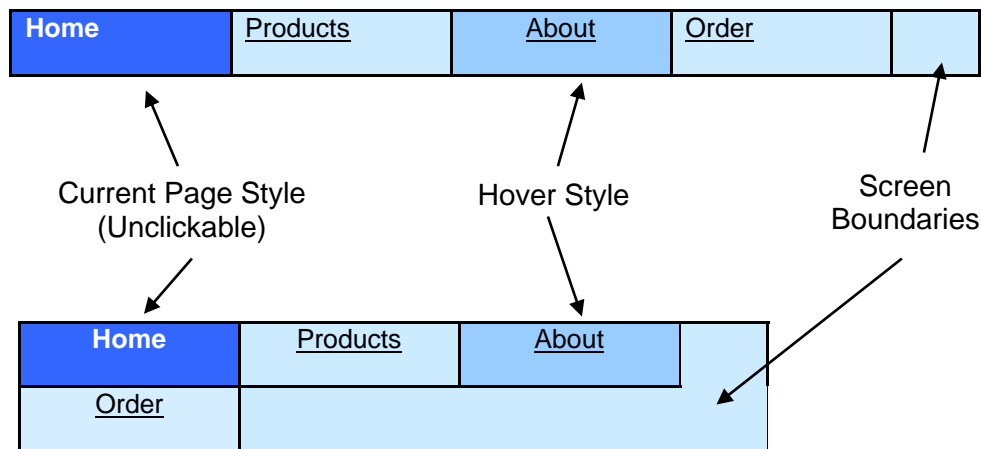
## Task 8: Navigation

The navigation bar appears on every page of the website and contains the following aspects:

1. The navigation bar has links to the five main pages of the website (Tasks 1-5).
2. Each navigation item must be the same size.
3. The link for the current page is disabled and has a different colour/appearance.
4. The links for the non-current pages have a different colour/appearance when hovering.
5. The navigation bar is responsive, meaning that the menu options gracefully wrap to the next line when they cannot all fit horizontally on the screen.

Examples of the standard WIP navigation bar layout is below

- The first is normal mode
- The second is responsive mode



## Task 9: Presentation

The presentation of the website will largely be controlled by CSS with the following aspects:

1. All CSS code will be stored in a site-wide style file called **styles.css** and referenced on each web page.
2. The styling will include a variety of styling for layout.
3. The styling will include a variety of styling for content.
4. The presentation will endeavour to give the site a contemporary appearance, including HTML version 5 and CSS version 3 where practical.

A good way to test presentation is to perform cross-browser testing. I.e.: View your website in a range of popular web browsers and check that the presentation is consistent.

## Task 10: Code

This task will award marks according to the quality of your code according to these aspects:

1. Code indentation is consistent with one level of indentation per block.
2. All ASP, C# and CSS file types each use some comments (<%-- comment --%>, <!-- comment -->, /\* comment \*/ or // comment as appropriate, minimum of 3 per file type).
3. The Visual Studio 2017 Error List reports zero errors and zero warnings.

*Although marks are not allocated, it is also good practice for code to consistently use best-practice casing (lowercase, camelCase, TitleCase as needed).*

## Submission

When you have completed, submit your answers via the link on the Learning Portal. You must submit the following archive:

- a) Zip your Visual Studio 2017 solution folder to a file called **xxx\_cse2icx\_assignment3.zip**, where xxx is your student number. Take care to zip your solution folder rather than the contents of the folder to avoid a [arbomb effect](#). The following files will be expected in your folder:
- ASP.NET × 6: One file for each of Tasks 1-5 plus print.aspx (Task 3).
  - C# × 2: Data model classes used with serialisation code (Tasks 1-2).
  - XML × 1: data file (Tasks 1-2).
  - Subfolder: "images" subfolder with graphic assets (Task 6c).
  - Text file: references.txt for image references (Task 6c).
  - Solution file: xxx\_cse2icx\_assignment3.sln.
  - Web.config: Visual Studio may generate this file, which you can leave alone.

Note: The Learning Portal will impose a 20mb limit for uploading your files. You will need to make changes to your submission if you exceed this limit. A common reason for exceeding this limit is including unscaled photographs. Another common reason is including unnecessary assembly files in your Visual Studio solution. Refer to Task 7 on creating your Visual Studio 2017 solution to avoid unnecessary assemblies.

## Assessment marking criteria

The marks for this assignment will be awarded as per the following marking rubric:

Task	Marks	Description	Total
1	0	The home page design could be improved.	7
	1	The home page design was attractive.	
	0	The data file was not provided.	
	1	The data file was provided.	
	0	The data file did not have valid changes.	
	1	The data file was validly modified and/or extended.	
	0	The code-behind was poor.	
	1	The code-behind could be improved.	
	2	The code-behind was done well: file reading, deserialisation and data binding.	
	0	The data grid data was not rendered.	
	1	The data grid data was rendered.	
	0	The label summary was incomplete.	
	1	The label summary code and output were correct.	
2	0	The form controls were incomplete.	7
	1	The form controls need improvement.	
	2	The form controls were complete.	
	0	The form control validation was incomplete.	
	1	The form control validation needs improvement.	
	2	The form control validation was complete.	
	0	The form did not submit correctly.	
	1	The form submits correctly.	
	0	The data was serialised and saved incorrectly.	
	1	The data was serialised and saved with errors.	
	2	The data was serialised and saved correctly.	
3	0	The form controls were incomplete.	7
	1	The form controls need improvement.	
	2	The form controls were complete.	
	0	The form control validation was incomplete.	

	1	The form control validation needs improvement.	
	2	The form control validation was complete.	
	0	The form did not submit correctly.	
	1	The form submits correctly.	
	0	The printable invoice data had errors.	
	1	The printable invoice data was correct.	
	0	The printable invoice presentation needs improvement.	
	1	The printable invoice was well presented.	
4	0	The form controls were incomplete.	8
	1	The form controls need improvement.	
	2	The form controls were complete.	
	0	The form control validation was incomplete.	
	1	The form control validation needs improvement.	
	2	The form control validation was complete.	
	0	The increment/decrement buttons did not behave correctly.	
	1	The increment/decrement buttons behaved correctly.	
	0	The calculate button did not behave correctly.	
	1	The calculate button behaved correctly.	
	0	The stocktake calculations were incorrect.	
	1	The stocktake calculations were correct in part.	
	2	The stocktake calculations were correct.	
5	0	The chart data was poor.	5
	1	The chart data could be improved.	
	2	The chart data was good.	
	0	The chart components were poor.	
	1	The chart components could be improved.	
	2	The chart components were good.	
	0	The chart presentation was unsatisfactory.	
	1	The chart presentation was satisfactory.	
6	0	Image references were incomplete.	1
	1	A list of image references correctly acknowledged the name, author, location and access date of the third-party images.	
7	0	The header areas were incomplete.	8
	1	The header areas could be improved.	
	2	A header area was included on all pages with the site name and branding.	
	0	The navigation areas were incomplete.	
	1	The navigation areas could be improved.	
	2	A navigation area was included on all pages with a navigation bar.	
	0	The main areas were incomplete.	
	1	The main areas could be improved.	
	2	A main area was included on all pages with the main content.	
	0	The footer areas were incomplete.	
	1	The footer areas could be improved.	
	2	A footer area was included on all pages with the copyright statement.	
8	0	The navigation bar consistency was poor.	9
	1	The navigation bar consistency could be improved.	
	2	The navigation bar links to the 5 main pages were on all pages.	
	0	Each navigation item was not the same size.	
	1	Each navigation item was the same size.	
	0	The navigation item for the current page was not always disabled.	
	1	The navigation item for the current page was always disabled.	
	0	The links for the non-current pages did not have a different colour/appearance when hovering.	
	1	The links for the non-current pages had a different colour/appearance when hovering.	
	0	The navigation bar responsive component was incomplete.	

	1	The navigation bar responsive component needs improvement.	
	2	The navigation bar was responsive, meaning that the menu options gracefully wrapped to the next line when they cannot all fit horizontally on the screen.	
	0	The navigation bar links did not go to the correct locations.	
	1	The navigation bar links sometimes went to the correct locations.	
	2	The navigation bar links always went to the correct locations.	
9	0	All CSS code was not stored in a site-wide style file.	8
	1	All CSS code was stored in a site-wide style file.	
	0	The style file was not correctly referenced on each web page.	
	1	The style file was correctly referenced on each web page.	
	0	The styling for layout was poor.	
	1	The styling for layout could be improved.	
	2	The styling included a variety of good styling for layout.	
	0	The styling for content was poor.	
	1	The styling for content could be improved.	
	2	The styling included a variety of good styling for content.	
	0	The presentation was poor.	
	1	The presentation could be improved.	
	2	The presentation gave the site a contemporary appearance, including HTML version 5 and CSS version 3 where practical.	
10	0	Indentation was poor.	10
	1	Indentation could be improved.	
	2	Indentation only needs minor improvement.	
	3	Code indentation was consistent with one level of indentation per block.	
	0	ASP file commenting was poor.	
	1	ASP file commenting was good (minimum of 3).	
	0	C# file commenting was poor.	
	1	C# file commenting was good (minimum of 3).	
	0	CSS file commenting was poor.	
	1	CSS file commenting was good (minimum of 3).	
	0	The Error List reported 5+ errors.	
	1	The Error List reported 1-4 errors.	
	2	The Error List reported 0 errors.	
	0	The Error List reported 5+ warnings.	
	1	The Error List reported 1-4 warnings.	
	2	The Error List reported 0 warnings.	
	0	Feedback will also be given regarding submission timeliness, length and format. Penalties may apply.	0
		<b>Total available marks</b>	<b>70</b>

## Appendix A Plant List

Common Name	Botanical Name	Description
<b>Silver Princess Gum</b>	<i>Eucalyptus caesia</i>	A native of Western Australia this gum grows to between 6-8m and can live for up to 150 years.
<b>Snow Gum</b>	<i>Eucalyptus pauciflora</i>	Native to eastern Australia the Snow Gum grows to a height of 5-7m and will put up with some frost
<b>Red Flowering Gum</b>	<i>Corymbia ficifolia</i>	This flowering gum will grow to a size of around 3-4 m and produces stunning red flowers in late spring
<b>Crimson Bottlebrush</b>	<i>Callistemon citrinus</i>	This tough plant is great to use as hedging and displays wonderful red flowers from Spring through to Summer
<b>Happy Wanderer</b>	<i>Hardenbergia violacea</i>	With its pretty purple flowers and quick rate of expansion this climbing plant is ideal for covering pergolas, fences, and trellises.
<b>Poa Grass</b>	<i>Poa labillardierei</i>	<i>Poa labillardierei</i> is a native grass that can grow to a height of approximately 120cm and looks especially attractive around water features.
<b>Golden Wattle</b>	<i>Acacia pycnantha</i>	A tree that grows to a height of 10m it delights with vibrant yellow flowers in spring.
<b>Lilly Pilly</b>	<i>Syzygium smithii</i>	Growing to a height of 3-5m grown together these plants can be used for hedging. Has pretty flowers in the summer and fruits in winter.
<b>Canberra Bells</b>	<i>Correa federation bell x mannii</i>	This pretty shrub grows to a height of 1m and produces lovely red and white flowers in autumn.
<b>Grevillia Firecracker</b>	<i>Grevillea alpina x rosmarinifolia</i>	This easy to maintain shrub grows to a height of 0.5m and has colourful red and yellow flowers from Autumn right through until Spring
<b>Silky Oak</b>	<i>Grevillia robusta</i>	This graceful tree can grow to a height of between 10-30m and produces wonderful orange and gold flowers in spring
<b>Waratah</b>	<i>Telopea speciosissima</i>	This evergreen shrub grows to a height of 3m. It displays astounding red flowers throughout spring
<b>Rose Boronia</b>	<i>Boronia serrulata</i>	This shrub will grow up to 1 m in height and does best in well drained soils. Pretty pink flowers are produced during the spring.
<b>Tall Kangaroo Paw</b>	<i>Anigozanthos flavidus</i>	Growing up to a meter long this hardy plant flowers from late spring all through summer
<b>Rusty Gum</b>	<i>Angophora costata</i>	This tree is best known for its colourful peeling bark. It grows up to 30m in height and has pretty white flowers.