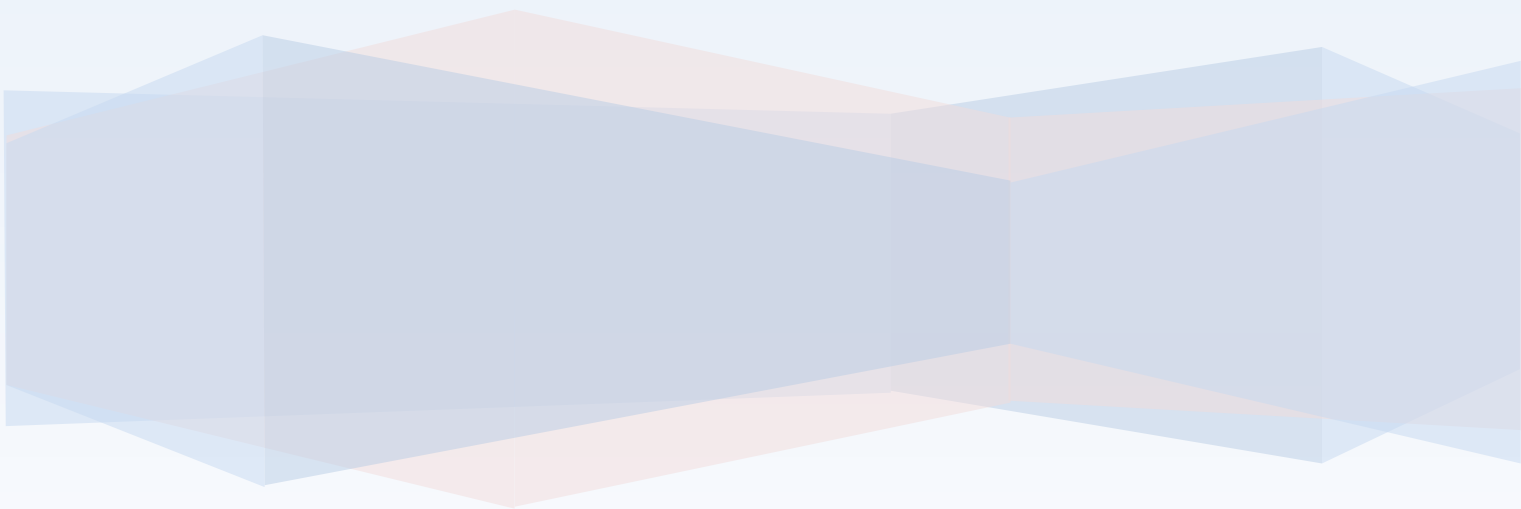


# COS30043 – Interface Design and Development

---

*Learning Summary Report*

Le Gia Hoang An - 104789808



## Self-Assessment Details

The following checklists provide an overview of my self-assessment for this unit.

	Pass (D)	Credit (C)	Distinction (D)	High Distinction (HD)
Self-Assessment (please tick)				X

### *Self-assessment Statement*

	Included (please tick)
Learning Summary Report	X
Use of Bootstrap that demonstrate coverage of core concepts	X
Use of VueJS that demonstrate coverage of core concepts	X

### *Minimum Pass Checklist*

	Included (please tick)
Progress on Credit Tasks	X
All Pass Tasks signed off	X

### *Minimum Credit Checklist, in addition to Pass Checklist*

	Included (please tick)
Credit and Pass Tasks done, and Progress on Distinction Tasks.	x
Custom program meets Distinction criteria	x
Design report with screenshots for custom program	x

### *Minimum Distinction Checklist, in addition to Credit Checklist*

	Included (please tick)
Research report, and associated pieces	x
Custom project meets HD requirements	x

### *Minimum High Distinction Checklist, in addition to Distinction Checklist*

## Declaration

I declare that this portfolio is my individual work. I have not copied from any other student's work or from any other source except where due acknowledgment is made explicitly in the text, nor has any part of this submission been written for me by another person.

Signature: Hoang An

## Introduction

This report summarises what I learnt in COS30043 – Interface Design and Development. It includes a self-assessment against the criteria described in the unit outline, a justification of the pieces included details of the coverage of the unit's intended learning outcomes, and a reflection on my learning.

## Overview of Pieces Included

This section outlines the pieces that I have included in my portfolio...

1. **Task 1.1P:** Show casing best HTML and CSS project.
2. **Task 1.2P:** Create the classic "Hello World" web page.
3. **Task 1.3P:** Use online Accessibility Validator Achecker to check if websites met the Web Accessibility Initiative and the Web Content Accessibility Guidelines.
4. **Task 1.4P:** Fix the website and use online Accessibility Validator Achecker to check if websites met the Web Accessibility Initiative and the Web Content Accessibility Guidelines.
5. **Task 2.1P:** Create the classic "Hello World" web page with Bootstrap Framework.
6. **Task 2.2P:** Create a calculator web page with Bootstrap Framework.
7. **Task 2.3P:** Create a website template that implements Bootstrap's Grid system.
8. **Task 3.1P:** Implement Vue directives to HTML form input.
9. **Task 3.2P:** Create a data table with search features with JSON data set, Bootstrap, and VueJS Framework.
10. **Task 3.3C:** Create a BMI computing web app with Bootstrap and VueJS Framework.
11. **Task 3.4C:** Create a Mobile Phone Registration App with Bootstrap and VueJS Framework.
12. **Task 4.1P:** Create a web app game that guesses number with VueJS Framework.
13. **Task 5.1P:** Create a web app that insert a status into the timeline. As there is no persistent storage, the timeline will initially be empty. The latest status will always be displayed on top pushing down previous status. Each status will also contain a delete button that will enable the user to delete the specific status.
14. **Task 5.2P:** Create a menu list component that uses v-bind directive.
15. **Task 5.3C:** Create a table with a link that uses Vue router to show additional data of the clicked data row.
16. **Task 6.1C:** Implement form validation on HTML form with VueJS and Bootstrap frameworks.
17. **Task 6.2HD:** Creating a custom web application that uses VueJS and Bootstrap frameworks, Fetch API, and MySQL.
18. **Task 7.1P:** Display data from jQuery getJSON() method
19. **Task 7.2P:** Display data from JavaScript Fetch API method
20. **Task 8.1P:** Create a web page that displays data from JavaScript Fetch API with Vue Pagination, VueJS and Bootstrap Frameworks.
21. **Task 8.2C:** Create a web page that displays data set from Task 7.2 from JavaScript Fetch API with Vue Pagination, VueJS and Bootstrap Frameworks.
22. **Task 9.1P:** Create a single page application including router components, with incorporation of Task 3.1 "Name Test," Task 5.1 "Post," and Task 8.1 "Student Marks."
23. **Task 9.2C:** Creating a single page application like Task 9.1P with Vue router, a dashboard for CRUD, pagination to show units results.
24. **Task 10.1P:** Creating a VueJS web application with Vue CLI and NodeJS.
25. **Task 10.2HD:** Create a coding video tutorial on a fully functional login page with form validation and fetch API. This includes content from Lecture 4, 6, and 7.

## Coverage of the Intended Learning Outcomes

This section outlines how the pieces I have included demonstrate the depth of my understanding in relation to each of the unit's intended learning outcomes.

### ILO 1: Apply Design

*Apply fundamental design concepts and standards to the development of user interfaces*

The following pieces demonstrate my ability in relation to this ILO:

- **Task 1.1P:** Show casing best HTML and CSS project.
- **Task 1.2P:** Create the classic "Hello World" web page.
- **Task 1.3P:** Use online Accessibility Validator Achecker to check if websites met the Web Accessibility Initiative and the Web Content Accessibility Guidelines.
- **Task 1.4P:** Fix the website and use online Accessibility Validator Achecker to check if websites met the Web Accessibility Initiative and the Web Content Accessibility Guidelines.
- **Task 2.1P:** Create the classic "Hello World" web page with Bootstrap Framework.
- **Task 2.2P:** Create a calculator web page with Bootstrap Framework.
- **Task 2.3P:** Create a website template that implements Bootstrap's Grid system.
- **Task 3.1P:** Implement Vue directives to HTML form input.
- **Task 3.2P:** Create a data table with search features with JSON data set, Bootstrap, and VueJS Framework.
- **Task 3.3C:** Create a BMI computing web app with Bootstrap and VueJS Framework.
- **Task 3.4C:** Create a Mobile Phone Registration App with Bootstrap and VueJS Framework.
- **Task 4.1P:** Create a web app game that guesses number with VueJS Framework.
- **Task 5.1P:** Create a web app that insert a status into the timeline. As there is no persistent storage, the timeline will initially be empty. The latest status will always be displayed on top pushing down previous status. Each status will also contain a delete button that will enable the user to delete the specific status.
- **Task 5.2P:** Create a menu list component that uses v-bind directive.
- **Task 5.3C:** Create a table with a link that uses Vue router to show additional data of the clicked data row.
- **Task 6.1C:** Implement form validation on HTML form with VueJS and Bootstrap frameworks.
- **Task 6.2HD:** Creating a custom web application that uses VueJS and Bootstrap frameworks, Fetch API, and MySQL.
- **Task 7.1P:** Display data from jQuery getJSON() method
- **Task 7.2P:** Display data from JavaScript Fetch API method
- **Task 8.1P:** Create a web page that displays data from JavaScript Fetch API with Vue Pagination, VueJS and Bootstrap Frameworks.
- **Task 8.2C:** Create a web page that displays data set from Task 7.2 from JavaScript Fetch API with Vue Pagination, VueJS and Bootstrap Frameworks.
- **Task 9.1P:** Create a single page application including router components, with incorporation of Task 3.1 "Name Test," Task 5.1 "Post," and Task 8.1 "Student Marks."
- **Task 9.2C:** Creating a single page application like Task 9.1P with Vue router, a dashboard for CRUD, pagination to show units results.
- **Task 10.1P:** Creating a VueJS web application with Vue CLI and NodeJS.
- **Task 10.2HD:** Create a coding video tutorial on a fully functional login page with form validation and fetch API. This includes content from Lecture 4, 6, and 7.

### ILO 2: Use Frameworks

*Use contemporary frameworks to create dynamic user interfaces.*

The following pieces demonstrate my ability in relation to this ILO:

- **Task 2.1P:** Create the classic "Hello World" web page with Bootstrap Framework.

- **Task 2.2P:** Create a calculator web page with Bootstrap Framework.
- **Task 2.3P:** Create a website template that implements Bootstrap's Grid system.
- **Task 3.1P:** Implement Vue directives to HTML form input.
- **Task 3.2P:** Create a data table with search features with JSON data set, Bootstrap, and VueJS Framework.
- **Task 3.3C:** Create a BMI computing web app with Bootstrap and VueJS Framework.
- **Task 3.4C:** Create a Mobile Phone Registration App with Bootstrap and VueJS Framework.
- **Task 4.1P:** Create a web app game that guesses number with VueJS Framework.
- **Task 5.1P:** Create a web app that insert a status into the timeline. As there is no persistent storage, the timeline will initially be empty. The latest status will always be displayed on top pushing down previous status. Each status will also contain a delete button that will enable the user to delete the specific status.
- **Task 5.2P:** Create a menu list component that uses v-bind directive.
- **Task 5.3C:** Create a table with a link that uses Vue router to show additional data of the clicked data row.
- **Task 6.1C:** Implement form validation on HTML form with VueJS and Bootstrap frameworks.
- **Task 6.2HD:** Creating a custom web application that uses VueJS and Bootstrap frameworks, Fetch API, and MySQL.
- **Task 7.1P:** Display data from jQuery getJSON() method
- **Task 7.2P:** Display data from JavaScript Fetch API method
- **Task 8.1P:** Create a web page that displays data from JavaScript Fetch API with Vue Pagination, VueJS and Bootstrap Frameworks.
- **Task 8.2C:** Create a web page that displays data set from Task 7.2 from JavaScript Fetch API with Vue Pagination, VueJS and Bootstrap Frameworks.
- **Task 9.1P:** Create a single page application including router components, with incorporation of Task 3.1 "Name Test," Task 5.1 "Post," and Task 8.1 "Student Marks."
- **Task 9.2C:** Creating a single page application like Task 9.1P with Vue router, a dashboard for CRUD, pagination to show units results.
- **Task 10.1P:** Creating a VueJS web application with Vue CLI and NodeJS.
- **Task 10.2HD:** Create a coding video tutorial on a fully functional login page with form validation and fetch API. This includes content from Lecture 4, 6, and 7.

### ILO 3: Develop User Interfaces

*Design and develop user interfaces optimised for a range of devices and platforms.*

The following pieces demonstrate my ability in relation to this ILO:

- **Task 2.1P:** Create the classic "Hello World" web page with Bootstrap Framework.
- **Task 2.2P:** Create a calculator web page with Bootstrap Framework.
- **Task 2.3P:** Create a website template that implements Bootstrap's Grid system.
- **Task 3.1P:** Implement Vue directives to HTML form input.
- **Task 3.2P:** Create a data table with search features with JSON data set, Bootstrap, and VueJS Framework.
- **Task 3.3C:** Create a BMI computing web app with Bootstrap and VueJS Framework.
- **Task 3.4C:** Create a Mobile Phone Registration App with Bootstrap and VueJS Framework.
- **Task 4.1P:** Create a web app game that guesses number with VueJS Framework.
- **Task 5.1P:** Create a web app that insert a status into the timeline. As there is no persistent storage, the timeline will initially be empty. The latest status will always be displayed on top pushing down previous status. Each status will also contain a delete button that will enable the user to delete the specific status.
- **Task 5.2P:** Create a menu list component that uses v-bind directive.
- **Task 5.3C:** Create a table with a link that uses Vue router to show additional data of the clicked data row.
- **Task 6.1C:** Implement form validation on HTML form with VueJS and Bootstrap frameworks.

- **Task 6.2HD:** Creating a custom web application that uses VueJS and Bootstrap frameworks, Fetch API, and MySQL.
- **Task 7.1P:** Display data from jQuery getJSON() method
- **Task 7.2P:** Display data from JavaScript Fetch API method
- **Task 8.1P:** Create a web page that displays data from JavaScript Fetch API with Vue Pagination, VueJS and Bootstrap Frameworks.
- **Task 8.2C:** Create a web page that displays data set from Task 7.2 from JavaScript Fetch API with Vue Pagination, VueJS and Bootstrap Frameworks.
- **Task 9.1P:** Create a single page application including router components, with incorporation of Task 3.1 "Name Test," Task 5.1 "Post," and Task 8.1 "Student Marks."
- **Task 9.2C:** Creating a single page application like Task 9.1P with Vue router, a dashboard for CRUD, pagination to show units results.
- **Task 10.1P:** Creating a VueJS web application with Vue CLI and NodeJS.
- **Task 10.2HD:** Create a coding video tutorial on a fully functional login page with form validation and fetch API. This includes content from Lecture 4, 6, and 7.

## ILO 4: Evaluate User Interfaces

*Evaluate user interfaces with respect to usability and accessibility using appropriate techniques and propose improvements.*

The following pieces demonstrate my ability in relation to this ILO:

- **Task 2.1P:** Create the classic "Hello World" web page with Bootstrap Framework.
- **Task 2.2P:** Create a calculator web page with Bootstrap Framework.
- **Task 2.3P:** Create a website template that implements Bootstrap's Grid system.
- **Task 3.1P:** Implement Vue directives to HTML form input.
- **Task 3.2P:** Create a data table with search features with JSON data set, Bootstrap, and VueJS Framework.
- **Task 3.3C:** Create a BMI computing web app with Bootstrap and VueJS Framework.
- **Task 3.4C:** Create a Mobile Phone Registration App with Bootstrap and VueJS Framework.
- **Task 4.1P:** Create a web app game that guesses number with VueJS Framework.
- **Task 5.1P:** Create a web app that insert a status into the timeline. As there is no persistent storage, the timeline will initially be empty. The latest status will always be displayed on top pushing down previous status. Each status will also contain a delete button that will enable the user to delete the specific status.
- **Task 5.2P:** Create a menu list component that uses v-bind directive.
- **Task 5.3C:** Create a table with a link that uses Vue router to show additional data of the clicked data row.
- **Task 6.1C:** Implement form validation on HTML form with VueJS and Bootstrap frameworks.
- **Task 6.2HD:** Creating a custom web application that uses VueJS and Bootstrap frameworks, Fetch API, and MySQL.
- **Task 7.1P:** Display data from jQuery getJSON() method
- **Task 7.2P:** Display data from JavaScript Fetch API method
- **Task 8.1P:** Create a web page that displays data from JavaScript Fetch API with Vue Pagination, VueJS and Bootstrap Frameworks.
- **Task 8.2C:** Create a web page that displays data set from Task 7.2 from JavaScript Fetch API with Vue Pagination, VueJS and Bootstrap Frameworks.
- **Task 9.1P:** Create a single page application including router components, with incorporation of Task 3.1 "Name Test," Task 5.1 "Post," and Task 8.1 "Student Marks."
- **Task 9.2C:** Creating a single page application like Task 9.1P with Vue router, a dashboard for CRUD, pagination to show units results.
- **Task 10.1P:** Creating a VueJS web application with Vue CLI and NodeJS.
- **Task 10.2HD:** Create a coding video tutorial on a fully functional login page with form validation and fetch API. This includes content from Lecture 4, 6, and 7.

## Reflection

### The most important things I learnt:

The most important things I have learnt in this unit are implementation of the CSS grid system and Single Page Web Application concept. The CSS grid system provides a clear layout for the content in my web page to be rendered, making it more organised. Additionally, the Single Page Web Application allows me to learn about folder structures for my code project and it makes code management easier. Thus, these two things are within my expectation of what I will be learning in this unit.

### The things that helped me most were:

The documents of the tools that I have used so far in this units such as Bootstrap and VueJS were extremely helpful when I am working on my assignments. It guided me to create the basic structures of my web application.

### I found the following topics particularly challenging:

The most challenging topic for me was Fetch API. It was the most complex topic due to the entire structure that must be correctly written for it to work. Debugging an API was also time consuming because I must look through each line to find out where the code is causing errors.

### I found the following topics particularly interesting:

There is a topic that interests me is the Bootstrap Grid System. Although it was difficult to grasp at first due to my weak CSS skills, I find it fascinating that there is a much easier way to structure my web page without messing up it's responsive nature.

### I feel I learnt these topics, concepts, and/or tools really well:

I feel that I was able to apply VueJS and Bootstrap well, as shown in my calculator app, search filter app, and router app. I was able to create functional web application with dynamic data that are supported by features from VueJS and Bootstrap tools.

### I still need to work on the following areas:

I still need to work on my skill in writing Fetch API, as I was not able to fully understand the entire its entire structure in Task 7.1 and Task 7.2.

### My progress in this unit was ...:

In my opinion, my progress in this unit was decent, as I was able to complete and submit Pass and Credit tasks. However, due to my time management skills and my research efficiency, I was a bit slow with working on my custom program.

### This unit will help me in the future:

This unit will help me in my future software development career. What I had learnt in this unit are the fundamental concepts in developing a web application. I believe that what I have learnt in this unit will be used soon for further studying.

### If I did this unit again I would do the following things differently:

If I do this unit again, I would try to improve my time management, so that I can have more time to further enhance my work in this unit.

### Other...:

I have said all my reflections above.