AUCKLAND INSTITUTE OF STUDIES NEW ZEALAND

INFORMATION TECHNOLOGY PROGRAMME

SOFT703 WEB APPLICATIONS DEVELOPMENT

Individual Assignment 1: Trimester 3, 2025

Due Date:	24th October 2025 4:00 PM
Where:	Submit to Moodle

This assignment is worth 30% of the total marks in the course and will cover the following learning outcomes as listed in the Course Outline:

1. Perform a proper analysis and design of a given business case study for implementation in Microsoft .NET Technology.
2. Design and develop the presentation layer for the web application using: ASP.Net, HTML5, CSS3 and Java-script.
4. Develop data access layer (DAL) and Business logic layer (BLL) and then construct web applications with web service component to access either business logic layer (BLL) or Data access layer (DAL) of given business case study.
6. Develop workplace soft skills including working in groups, writing formal reports, carrying out individual research and/or delivering oral presentations.

General Instructions:

- Include a cover page that includes your name, student ID, course code and title, assignment title
- All sources should be acknowledged in the main body of the assignment and in a reference list at the end of the assignment. You are also required to reference any tools/software/applications used. Refer to the 7th edition of the APA referencing style guide. Examples of how to present references can be found here: https://apastyle.apa.org/style-grammar-guidelines/references/examples.
- Your assignment should be your own work to demonstrate your thinking. Any direct quotes from any source should be placed in “quotation marks”.

- Upon submission, your assignment will be checked for copied materials as well as the use of AI (artificial intelligence), with penalties to be applied where appropriate and if not referenced.
- You may resubmit draft versions of your assignment **until the due date**. After the due date, the latest version will be accepted as the final version. Note that similarity reports may take 24 hours to generate.
- Resources to assist with writing skills can be found at https://moodle.ais.ac.nz/course/view.php?id=4#section-4
- Late submissions will be penalised by 5% of the available marks from the marks awarded per working day (24 hours) for up to four days and receive a zero mark after four days. (96 hours).
- Mark Distribution is as follows:

TASKS	MARKS DISTRIBUTION	MARKS EARNED
Task 1 (Analysis & Design)	28	
Task 2 (Develop Web App)	60	
Task 3 (Additional Requirements)	52	
Task 4 (App Demo)	20	
TOTAL	160	

This assignment is marked out of 160 marks; and is worth 30% of the total course marks.

Marking Rubric for Assignment 1 provided in Appendix A

Business Scenario

Suppose you are hired as a software developer by the Software Development Unit (SDU) of the Auckland Institute of Studies (AIS). The SDU currently provides a web-based solution using ASP.NET Core MVC, C#, HTML CSS, JavaScript, and jQuery. SDU has asked you to develop a Web Application. For the choice of application, you can develop application on any of the following application examples. However, the application should be user-friendly, access all the data from the back end (Microsoft SQL Server), and fulfill all the requirements given in the tasks below.

Note: Your Lecturer is the client for this Web Application. Before final submission, you can show your application for feedback on the user interface or functionality. You may also discuss with your Lecturer for any clarification regarding the application scope.

Application examples:

- New Zealand fishing web app
- Online quiz
- New Zealand driving test
- Online deals like www.grabone.co.nz
- Real client application (i.e. for restaurant, automotive, health care).

Fictitious company i.e. Web Application for Educational Institute targeting any Private Tertiary Establishment (PTE). **Note:** If you want to use different web applications other than above stated examples, you need to discuss with your Lecturer and get approval within the five days of receiving this assignment. CONFIRM WITH YOUR LECTURER THE WEB APP YOU CHOOSE!

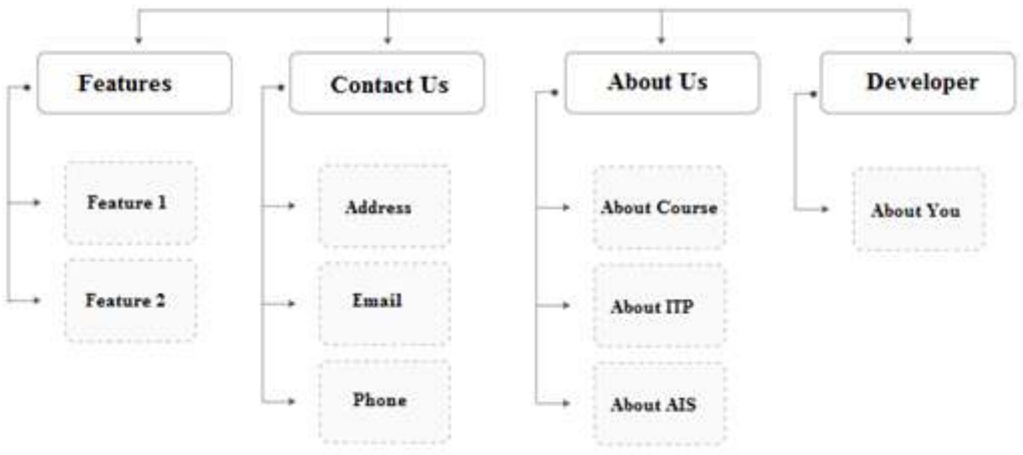

Figure 1: An example of application sitemap.

TASKS

TASK 1 – ANALYSE & DESIGN: (28 Marks)

Conduct a proper analysis of the business that needs the web application and produce the following designs ready to develop a web application in ASP.NET Core MVC:

- Activity Diagrams
- Use Case Diagram for the Web Application
- Sitemap
- User Interface Designs
- Database Design such as ERD
- Class Diagram
- List of functional and non-functional requirements

TASK 2 – DEVELOP WEB APPLICATION: (60 MARKS)

Your application should have:

1) Web Application for CUSTOMERS or USERS (25 marks)

Where customers or users can:

i. View (browse, navigate) PRODUCTS/SERVICES from a specific Category/Type that your web application provides or sells. (15 marks)

ii. View a selected PRODUCTS/SERVICES details and add it to cart. (10 marks)

Note: Marks may be reduced if any requirement is incomplete or not applied appropriately.

2) Content Management System: (35 marks)

i. **PRODUCTS/SERVICE Management System** used by staff to maintain (add, edit, and delete) the PRODUCTS/SERVICES records (20 marks)

ii. **Customer Management System** used by staff to maintain (add, edit, and delete) customers' details (15 marks)

Note: Marks may be reduced if any requirement is incomplete or not applied appropriately.

TASK 3: OTHER REQUIREMENTS: (52 MARKS)

Your web application should:

1) Have a MVC View master page that consists of a Header, Content, and Footer should be implemented and used by all the pages of your website. (9 marks)
2) Implement an ASP.NET index/home page by using MVC View Master Page. (4 marks)
3) Appropriate links/buttons/menus should be added to the application to enable smooth navigation between the pages of the website. (4 marks)
4) Apply different types of appropriate data controls (such as DataList, GridView, ListView, DetailsView, and FormView, etc.) to display and maintain your database data appropriately. (8 marks)
5) Validate the data users enter where appropriate and display error messages accordingly. (6 marks)
6) Always display an error message if an exception occurs or if the operation cannot be performed while maintaining the database. (6 marks)
7) Provide professional design of web pages. (6 marks)
8) The PRODUCTS/SERVICES and customer data of your web application should be saved in a MS SQL Server database file which should be added to your web application folder. (3 marks)
9) Briefly describe the n-tier architecture in relation to 3-Tier architecture and MVC. (6 marks)

TASK 4: APPLICATION DEMO: (20 MARKS)

Record a complete demonstration of your application covering the key learning points especially to support your answers for Task1, 2 and 3 and upload it to Moodle drop box. The recording time should be no more than 20 minutes. The upload link is Assignment1 Demo.

APPENDIX A – MARKING RUBRIC FOR ASSIGNMENT 1

TASK 1 – ANALYSE & DESIGN:			**(28 Marks)**		
Deliverable	**4 – Excellent**	**3 – Good**	**2 – Fair**	**1 – Poor**	**0 – Very Poor**
1. Activity Diagram	Diagram created using the appropriate tool like Visio and it shows clearly **all** the necessary business processes in a logical order.	Diagram created using the appropriate tool like Visio and it shows clearly **most** of the necessary business processes in a logical order	Diagram created using the appropriate tool like Visio and it shows a **few** of the business processes in a simple sequence.	Diagram created using inappropriate tool and it **fails** to show clearly the business processes in any meaningful order.	No diagram provided or meaningless diagram provided.
2. Use Case Diagram	Diagram created using the appropriate tool like Visio and it shows clearly **all** the necessary business use cases and designed properly.	Diagram created using the appropriate tool like Visio and it shows clearly **most** of the necessary business use cases and designed properly.	Diagram created using the appropriate tool like Visio and it shows clearly a **few** of the necessary business use cases with simple designs.	Diagram created using inappropriate tool and it **fails** to show clearly the business use cases with unclear designs.	No diagram provided or meaningless diagram provided.
3. Sitemap	Diagram created using the appropriate tool like Visio and it shows clearly **all** the necessary web pages in a meaningful structure and must align well to other	Diagram created using the appropriate tool like Visio and it shows clearly **most** of the necessary web pages in a meaningful structure and may align well to other designs like User interfaces.	Diagram created using the appropriate tool like Visio and it shows clearly a **few** of the necessary web pages in a meaningful structure and may not align well	Diagram created using inappropriate tool and it fails to show clearly the necessary web pages in a meaningful structure and hard to align well to other designs like User interfaces.	No diagram provided or meaningless diagram provided.

	designs like User interfaces.		to other designs like User interfaces.		
4. User Interface Design	Diagrams created using the appropriate tool like Visio wireframe, Pencil project, Figma, etc and it shows clearly **all** the functional and non-functional features of the designs and satisfy completely the type of prototype used: low or high fidelity.	Diagrams created using the appropriate tool like Visio wireframe, Pencil project, Figma, etc and it shows clearly **most** of the functional and non-functional features of the designs and satisfy the type of prototype used: low or high fidelity.	Diagrams created using the appropriate tool like Visio wireframe, Pencil project, Figma, etc and it shows clearly a **few** of the functional and non-functional features of the designs and satisfy partially the type of prototype used: low or high fidelity.	Diagrams created using inappropriate tool and it **fails** to show clearly the functional and non-functional features of the designs and satisfy partially the type of prototype used: low or high fidelity.	No diagram provided or meaningless diagram provided.
5. Database Designs	Diagram created using the appropriate tool like Visio Crow Foot notation and it shows **clearly and correctly** the entities, attributes and relationships between entities; diagram must align to other design like class diagram.	Diagram created using the appropriate tool like Visio Crow Foot notation and it shows **clearly** the entities, attributes and relationships between entities; diagram may align to other design like class diagram.	Diagram created using the appropriate tool like Visio Crow Foot notation and it shows **simple** entities, attributes and relationships between entities; diagram may not align well to other design like class diagram.	Diagram created using the inappropriate tool and it shows **unclear** entities, attributes and relationships between entities; diagram does not align well to other design like class diagram.	No diagram provided or meaningless diagram provided.

6. Class Diagram	Diagram created using the appropriate tool like Visio and it shows **clearly and correctly** the classes, attributes, methods and relationships between classes; diagram must align to other design like database diagram.	Diagram created using the appropriate tool like Visio and it shows **clearly** the classes, attributes, methods and relationships between classes; diagram may align to other design like database diagram	Diagram created using the appropriate tool like Visio and it shows **simple** classes, attributes, methods and relationships between classes; diagram may not align well to other design like database diagram	Diagram created using the inappropriate tool and it shows **unclear** classes, attributes, methods and relationships between classes; diagram does not align well to other design like database diagram	No diagram provided or meaningless diagram provided.
7. Functional & Non-Functional Requirements	Comprehensive and relevant list of functional and non-functional requirements of the application is presented well in user story or standard format.	Accurate and relevant list of functional and non-functional requirements of the application is presented mostly in user story or standard format.	Accurate and simple list of functional and non-functional requirements of the application is presented partially in user story or standard format.	Simple list of functional and non-functional requirements of the application is presented but unclear in user story or standard format.	No list provided or meaningless list provided.

TASK 2 – DEVELOP WEB APPLICATION: (60%)					
1) Web Application for CUSTOMERS or USERS (25 %)					
Deliverable	**10 – Excellent**	**9 – 7: Good**	**6 – 4: Fair**	**3 – 1: Poor**	**0 – Very Poor**
1. View Product/Service (15%)	Run the web app and test the view feature for product/service by category/type. The app runs successfully where the authorised user can easily browse and navigate from page to page and the app can display well all or specific product/service when needed. At least five test cases are successful.	Run the web app and test the view feature for product/service by category/type. The app runs successfully where any user can easily browse and navigate from page to page and the app can display well most or specific product/service when needed. At least three test cases are successful.	Run the web app and test the view feature for product/service by category/type. The app runs successfully where any user can browse and navigate from page to page and the app can display a few or specific product/service when needed. At least two test cases are successful. .	Run the web app and test the view feature for product/service by category/type. The app runs successfully where any user cannot easily browse and navigate from page to page and the app cannot display well all or specific product/service when needed. At least one test case is successful.	No web app provided, or web app failed to run.
2. View selected Product/Service (10%)	Run the web app and test the view feature for product/service by category/type. The app runs successfully where the authorised	Run the web app and test the view feature for product/service by category/type. The app runs successfully where any user can easily	Run the web app and test the view feature for product/service by category/type. The app runs successfully where any user can navigate	Run the web app and test the view feature for product/service by category/type. The app runs successfully where any user cannot easily	No web app provided, or web app failed to run.

	user can easily navigate from page to page and the app can display well all the detailed information of a specific product/service being selected/clicked to view. At least five test cases are successful.	navigate from page to page and the app can display well most of the detailed information of a specific product/service being selected/clicked to view. At least three test cases are successful.	from page to page and the app can display well few of the detailed information of a specific product/service being selected/clicked to view. At least two test cases are successful.	navigate from page to page and the app cannot display well the detailed information of a specific product/service being selected/clicked to view. At least one test case is successful.	
2) Content Management System: (35%)					
Deliverable	**10 – Excellent**	**9 – 7: Good**	**6 – 4: Fair**	**3 – 1: Poor**	**0 – Very Poor**
1. Product Management (20%)	Run the web app and test the maintenance feature for product/service. The app runs successfully where the authorised user can easily perform add new item, edit and delete existing item and the app executes it successfully and shows appropriate	Run the web app and test the maintenance feature for product/service. The app runs successfully where any user can easily perform add new item, edit and delete existing item and the app executes it successfully and shows appropriate messages. At least three test cases are successful.	Run the web app and test the maintenance feature for product/service. The app runs successfully where any user can easily perform add new item, edit and delete existing item and the app executes it and shows appropriate messages, but product is partly saved. At least two test	Run the web app and test the maintenance feature for product/service. The app runs successfully where any user cannot easily perform add new item, edit and delete existing item and the app executes it and may show messages, but product is partly or not	No web app provided, or web app failed to run.

	messages. At least five test cases are successful.		cases are successful. .	saved. At least one test case is successful.	
2. Customer Management (15%)	Run the web app and test the maintenance feature for customer/user. The app runs successfully where the authorised user can easily perform add new user, edit and delete existing user and the app executes it successfully and shows appropriate messages. At least five test cases are successful.	Run the web app and test the maintenance feature for customer/user. The app runs successfully where any user can easily perform add new user, edit and delete existing user and the app executes it successfully and shows appropriate messages. At least three test cases are successful.	Run the web app and test the maintenance feature for customer/user. The app runs successfully where any user can perform add new user, edit and delete existing user and the app executes it successfully and may show appropriate messages. At least two test cases are successful.	Run the web app and test the maintenance feature for customer/user. The app runs successfully where any user cannot easily perform add new user, edit and delete existing user and the app executes it and may not show appropriate messages. At least one test case is successful.	No web app provided, or web app failed to run.

TASK 3 – OTHER REQUIREMENTS: (52 %)					
Deliverable	**10 – Excellent**	**9 - 7: Good**	**6 – 4: Fair**	**3 – 1: Poor**	**0 – Very Poor**
1. View Master Page (9%)	Run the web app and examine the source code to clearly identify the master page in the context of ASP.NET Core MVC. The master page/view is easy to locate and is well structured to contain a header, footer and content sections and is aligned well according to the designs like sitemap and user interface prototype.	Run the web app and examine the source code to clearly identify the master page in the context of ASP.NET Core MVC. The master page/view is easy to locate and has simple structure to contain a header, footer and content sections and is aligned well according to the designs like sitemap and user interface prototype.	Run the web app and examine the source code to clearly identify the master page in the context of ASP.NET Core MVC. The master page/view is not easy to locate and has simple structure to contain a header, footer and content sections and is aligned according to the designs like sitemap and user interface prototype.	Run the web app and examine the source code to clearly identify the master page in the context of ASP.NET Core MVC. The master page/view is hard to locate and has unclear structure to contain a header, footer and content sections and is not aligned well according to the designs like sitemap and user interface prototype.	No web app provided or web app failed to run.
2. Home Page (4%)	Run the web app and examine the source code to clearly identify the home page in the context of ASP.NET Core MVC. The home	Run the web app and examine the source code to clearly identify the home page in the context of ASP.NET Core MVC. The home page is easy to locate and is well	Run the web app and examine the source code to clearly identify the home page in the context of ASP.NET Core MVC. The home page is easy to locate	Run the web app and examine the source code to clearly identify the home page in the context of ASP.NET Core MVC. The home page is hard to identify	No web app provided, or web app failed to run.

	page is easy to locate and is well structured professionally and is aligned according to the designs like sitemap and user interface prototype.	structured but not professional and is aligned according to the designs like sitemap and user interface prototype.	and has simple structure but not professional and is aligned according to the designs like sitemap and user interface prototype.	and has simple structure but not professional and is not aligned according to the designs like sitemap and user interface prototype.	
3. Link/Button/Menu (4%)	Run the web and test the link, button or/and menu system to ensure a smooth navigation between views/pages is taken place. At least five test cases are successful.	Run the web and test the link, button or/and menu system to ensure a smooth navigation between views/pages is taken place. At least three test cases are successful.	Run the web and test the link, button or/and menu system to ensure a smooth navigation between views/pages is taken place. At least two test cases are successful.	Run the web and test the link, button or/and menu system to ensure a smooth navigation between views/pages is taken place. At least one test case is successful.	No web app provided, or web app failed to run.
4. Data Control (8%)	Run the web app and test the data control features such as data-list, grid-view, list-view, detail-view, form-view, etc to ensure they are working accordingly. At least five test cases are successful.	Run the web app and test the data control features such as data- list, grid-view, list-view, detail-view, form-view, etc to ensure they are working accordingly. At least three test cases are successful.	Run the web app and test the data control features such as data-list, grid-view, list-view, detail-view, form-view, etc to ensure working accordingly. At least two test cases are successful.	Run the web app and test the data control features such as data-list, grid-view, list-view, detail-view, form-view, etc to ensure working accordingly. At least one test case is successful.	No web app provided, or web app failed to run.

5. Validation (6%)	Run the web app and test it with various data entry validation scenarios such as numeric, date, and other necessary data type. An informative error message is displayed when validation occurs and violated. At least five test cases are successful.	Run the web app and test it with various data entry validation scenarios such as numeric, date, and other necessary data type. An informative error message is displayed when validation occurs and violated. At least three test cases are successful.	Run the web app and test it with various data entry validation scenarios such as numeric, date, and other necessary data type. A simple error message is displayed when validation occurs and violated. At least two test cases are successful.	Run the web app and test it with various data entry validation scenarios such as numeric, date, and other necessary data type. An unclear error message is displayed when validation occurs and violated. At least one test case is successful.	No web app provided, or web app failed to run.
6. Error Message (6%)	Run the web app and test it with various exception handling scenarios such as numeric or date data type, calculation, database/file, etc. An informative error message is displayed when an exception occurs. At least five test cases are successful.	Run the web app and test it with various exception handling scenarios such as numeric or date data type, calculation, database/file, etc. An informative error message is displayed when exception occurs. At least three test cases are successful.	Run the web app and test it with various exception handling scenarios such as numeric or date data type, calculation, database/file, etc. A simple error message is displayed when exception occurs. At least two test cases are successful.	Run the web app and test it with various exception handling scenarios such as numeric or date data type, calculation, database/file, etc. An unclear error message is displayed when exception occurs. At least one test case is successful.	No web app provided, or web app failed to run.

7. Professional Design (6%)	Run the web app and compare it to high fidelity prototype (user interface design) which shows above 90% match.	Run the web app and compare it to high fidelity prototype (user interface design) which shows above 70% match.	Run the web app and compare it to high fidelity prototype (user interface design) which shows above 50% match.	Run the web app and compare it to high fidelity prototype (user interface design) which shows above 30% match.	No web app or high-fidelity prototype provided, or web app failed to run.
8. Saved to database (3%)	Run the web app follows by entering relevant data; once the appropriate link or button is clicked, the system shows an informative saved to database message. Verify it to database the complete data is saved.	Run the web app follows by entering relevant data; once the appropriate link or button is clicked, the system shows an informative saved to database message. Verify it to database but incomplete data is saved.	Run the web app follows by entering relevant data; once the appropriate link or button is clicked, the system shows meaningful response. Verify it to database but no data is saved.	Run the web app follows by entering relevant data; once the appropriate link or button is clicked, the system shows meaningless response. Verify it to database but incomplete or no data is saved.	No web app provided, or web app failed to run.
9. N-Tier architecture (6%)	Comprehensive explanation of the N-Tier architecture and MVC pattern which shall cover clearly and completely each of those two concepts, their significant differences, usage	Adequate explanation of the N-Tier architecture and MVC pattern which covers clearly but partially each of the two concepts, their significant differences, usage and how helpful to this web	Sufficient explanation of the N-Tier architecture and MVC pattern which covers clearly but partially each of the two concepts, their significant differences, usage and how helpful	General explanation of the N-Tier architecture and MVC pattern which does not cover well each of the two concepts, their significant differences, usage and how helpful to this web	No explanation provided or meaningless explanation provided.

	and how helpful to this web app provided with relevant example.	app provided with relevant example.	to this web app provided with simple example.	app provided with unclear example.	

TASK 4 – APPLICATION DEMO: (20 Marks)					
Deliverable	**4 – Excellent**	**3 – Good**	**2 – Fair**	**1 – Poor**	**0 – Very Poor**
1. Focus/Scope	Purpose of demonstration is clear from the outset. Supporting ideas maintain clear focus on the topic.	Topic of the demonstration is clear. Content generally supports the purpose	Topic of demonstration is fairly presented with content partially supports the purpose.	Demonstration lacks clear direction. Big ideas not specifically identified.	No focus at all. Audience cannot determine purpose of demonstration.
2. Organization	Student presents information in logical, interesting sequence that audience follows	Student presents information in logical sequence that audience can follow	Audience has fairly following in a simple sequence that audience can partially follow.	Audience has difficulty following because student jumps around.	Audience cannot understand because there is no sequence of information

3. Time Management	Students start and finish demonstration in time; presented all items.	Students start and finish demonstration in time and could finish in time; presented some parts.	Students could start demonstration in time, but they finish in time,	Students could not star demonstration in time, but they finish in time,	Students could not start and could not finish demonstration in time,
4. Knowledge	Show clear and advanced knowledge and understanding of the OOP concepts, Designs and programming languages	Show clear knowledge and understanding of the OOP concepts, Designs and programming languages	Show fair knowledge and understanding of the OOP concepts, Designs and programming languages	Show basic knowledge and understanding of the OOP concepts, Designs and programming languages	Show no knowledge and not understanding of the OOP concepts, Designs and programming languages
5. Application features	Demonstrate all the application features such as functional and non-functional features with clear examples.	Demonstrate almost all the application features such as functional and non-functional features with clear examples.	Demonstrate fairly most parts of the application features such as functional and non-functional features with examples	Demonstrate some parts of the application features such as functional and non-functional features but unclear examples	Unable to demonstrate the application features such as functional and non-functional features.