ASSESSMENT AND INTERNAL VERIFICATION FRONT SHEET (Individual Criteria)

(Note : This version is to be used only for assignments uploaded via Classter)

Course Title	Bachelor of Science (Honours) in Multimedia Software Development		Lecturer Name & Surname		
Unit Number & Title	ITMMD-506-2003 & Client-side Scripting				
Assignment Number, Title / Type	Assignment 2 / Develop a Single Page Application using VUE / Take Home				
Date Set	11/05/2023	Deadline Date	5/06/2023		
Student Name		ID Number		Class / Group	

Assessment Criteria	Mark
LO1 - Apply different animations and transitions effects within the presentation layer of a web application	
AA1.4 - Construct animations and transitions to focus user attention and improve perceived performance	*7*
SE1.1 - Design animations to make the User Interface more intuitive	*10*
LO2 - Develop a web application with state management in place and provides users authentication	
KU2.4 - Present a feature where users can access different sections of the web application depending on their role	*5*
SE2.1 - Design a web application which provides state management and hence provides a data structure to record the outcomes of the user's actions.	*10*
SE2.2 - Develop a feature where users can log in if their credentials are correct	*10*
LO3 - Implement using a JavaScript framework/library a web application that allows users to create, read, update and delete data	
KU3.1 - Show how a JavaScript framework can be used to build a web application and insert, update and delete data	*5*
LO4 - Deploy a web application with proper navigation which can handle and validate user input data through forms	
AA4.2 - Use validation techniques to validate data input by the user	*7*
KU4.3 - Reproduce the web application into production	*5*
TOTAL	***59***

Notes to Students:

- This assignment brief has been approved and released by the Internal Verifier through Classter.

- Assessment marks and feedback by the lecturer will be available online via Classter (Http://mcast.classter.com) following release by the Internal Verifier

- Students submitting their assignment on Moodle will be requested to confirm online the following statements:

Student's declaration prior to handing-in of assignment

❖ I certify that the work submitted for this assignment is my own and that I have read and understood the respective Plagiarism Policy

Student's declaration on assessment special arrangements

❖ I certify that adequate support was given to me during the assignment through the Institute and/or the Inclusive Education Unit.

❖ I declare that I refused the special support offered by the Institute.

UNIT ITMMD-506-2003– CLIENTSIDE SCRIPTING 2

ASSIGNMENT: DEVELOP A SINGLE PAGE APPLICATION USING VUE

Unit Level: MQF LEVEL 6

General Guidelines

- This is a Take Home Assignment.
- This is an open book assignment.
- Students must carefully read the tasks and the rubrics before they attempt the answers.
- Deadline of this assignment is the **5th of June 2023**.
- Plagiarism is strictly prohibited.
- Following assignment submission, the lecturer will hold submission reviews. Students will be expected to explain their code. The lecturer may decide not to award some or all marks if not satisfied that the work presented is the student's own.
- Milestone submissions and the final assignments are to be submitted electronically on Moodle. No hard copies are to be submitted.

Submit milestones as specified in the table below.

Milestone	Task	Date	Deliverable
1	Proposal	16th May 2023	Section A - Template document
2	Final submission	5th June 2023	All sections

SECTION A - Proposal Document Template

The aim of this document is presenting an idea of a project and listing the requirements that the system needs to have. The lecturer will in turn give feedback and discuss with the student in case any changes are necessary. **Kindly keep with the proposal deadline as specified on the assignment.**

Write not more than one page outlining the features of the application. Make sure to include the following:

- What is the main objective of the Single Page Application?
- What technologies will be used to develop the SPA? (Database, JavaScript Framework, CSS Framework/UI Components Library, 3rd party libraries e.g. Axios, Vuelidate or Veevalidate)
- Where will the Single Page Application be hosted?
- What different access will logged in users (authenticated users) have?
- What data will be retrieved, updated, added or delete from the database?
- How will VUEX (state management) be implemented in the web application?

SECTION B – Implementation

The submitted project must comply with the proposal as specified in Section A.

The application is to have, at least, the following, and these requirements must make sense in the context in which they are used:

1) Advanced Routing
2) Authentication
3) VUEX – State Management
4) Making use of a VUE UI component library (Examples here)
5) Animations and Transitions
6) Connecting with the database (retrieve, update, delete and add)
7) Pagination of list retrieved
8) Http requests using fetch API or 3rd party library such as Axios
9) Using forms and validation
10) Hosting online

Criteria

AA1.4	Construct animations and transitions to smooth changes between states

Make use of the VUE transition component to apply group transitions to a list of elements. Apply an animation when a new item is added (2 marks) to the list and when an item is deleted from the list (2 marks).

Add pagination to the list being retrieved. A maximum number of items should be displayed on each page (2 marks). User can then move to the next page or going back to the previous page only if these are available. If these are not available, no page numbers should be displayed. (1 mark)

SE1.1	Design animations to make the User Interface more intuitive

Make use of the VUE transition component to:

- Use page transitions when the user navigates between routes/pages (2 marks).
- Display a message before a user deletes an item from the list. Provide a transition to provide more context of where the element came from. The user must confirm the deletion by clicking a button and then is redirected to the proper page (2 marks).

Use a VUE UI component library (Examples here) to improve the User Experience across the web application and build a clean, responsive and consistent User Interface. (6 marks).

SE2.1	Design a web application which provides state management and hence provides a data structure to record the outcomes of the user's actions

Use the VUEX library to store the information whether the user is logged in or not. Clearly use:

- mutations to have the logic to update the state (3 marks)
- getters to read the state (3 marks)
- actions to commit mutations (4 marks)

SE2.2	Develop a feature where users can log in if their credentials are correct

You are not required to create a backend and write server-side code (e.g. Node.js and PHP). However a service such as Firebase must be used. Cloud Firestore or Realtime Database must be used to permanently store data. Data should be stored and retrieved successfully from this backend (4 marks).

Create a registration form where a user enters at a minimum his/her detail such as username/email, password etc. This data should then be stored in the database (3 marks).

Create a login form where a user can log in to his/her account. This data is sent to the server. If the data is valid (3 marks) a token should be sent by the server to the frontend.

KU2.4	Present a feature where users can access different sections of the application depending on their role

Use the token sent by the server and attach it to the requests so that the server grants different access to other pages to logged in users and returns the appropriate data. If token is either invalid or missing access should be denied (5 marks).

KU3.1	Show how a JavaScript framework can be used to build a web application and insert, update and delete data

Create a single page application where a user can:

- Render a list of data from the database (1 mark)
- Add data to a database (1 mark)
- Update data on a database (2 marks)
- Delete data from a database (1 mark)

AA4.2	Use validation techniques to validate data input by the user

Make use of a 3rd party library such as Vuelidate or VeeValidate to display proper error messages when fields are not not filled correctly. The following rules apply:

- Disable the submit button until all fields are filled in correctly (1 marks)
- All fields are required (1 mark)
- The mobile should have a minimum length and a maximum length of 8 numbers (0.5 marks)
- The mobile must be numeric (0.5 marks)
- Password must be at least 3 characters long (1 mark)
- Email must be a valid email (1 mark).

All the above fields must be included in the registration form. Style the error messages by showing text of error message, border and label in red. (2 marks)

KU4.3	Reproduce the web application into production

Deploy your Vue app online (3 marks). Code will still be executed on the browser, but it must be uploaded on a server using a static hosting provider (e.g. Firebase Hosting). Make sure to use the build command to build and optimize the application for production and generate the dist folder (2 marks).

---END OF ASSIGNMENT---