

## 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	Carlo Mamo
Unit Number & Title	ITMMD-506-2002 Client-side Scripting I		
Assignment Number, Title / Type	1 – Develop a single page application using a front-end JavaScript framework/Home		
Date Set	18/12/2020	Deadline Date	18/01/2021
Student Name		ID Number	
		Class / Group	

Assessment Criteria	Maximum Mark
<b>L01: Apply basic requirements to build the presentation layer of a web application using HTML, CSS and JavaScript.</b>	
SE1: Develop a web application front-ends using common concepts such as data binding, events and methods.	10%
AA1: Use HTML, CSS and JavaScript to build rich user experiences for web applications using components.	7%
KU1: Illustrate how to access and manipulate the DOM.	5%
KU2: Construct JavaScript techniques to add interactivity in the web application.	5%
<b>L02: Develop a responsive web application using a CSS Framework.</b>	
SE2: Design a responsive mobile-first web application.	10%
AA2: Develop responsive forms with proper client-side validation messages using a CSS framework.	7%
KU3: Construct a fluid layout with fluid images by using a CSS Framework.	5%
SE3: Create basic transitions and animations.	10%
<b>L03: Implement using a JavaScript Framework/Library a Web Application that fetches and consumes data.</b>	
KU4: Show how a JavaScript framework can be used to build a web application.	5%
AA3: Apply filters to format the values data retrieved for display to the user.	7%
KU5: Construct small, self-contained reusable components that can communicate and pass data.	5%
AA4: Use a JavaScript framework to parse an exchange data format such as JSON data and display in the view.	7%
<b>L04: Deploy a web application with proper navigation which can handle and validate user input data through forms input data through forms.</b>	
KU6: Show how routing allows to navigate through the web application.	5%
AA5: Use validation techniques to validate data input by the user.	7%
KU7: Reproduce the web application online.	5%

Total Mark	100%
------------	------

**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/Unicheck 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.

**Notes to Students:**

- This assignment is in the form of a **take-home assignment**.

The College operates a cheating/**plagiarism policy** and any copied work will be penalized according to this policy. **The work you present must be your own work! Plagiarism will not be tolerated! You must *not* work in groups.**

- You will be required to demo your program and answer/justify technicalities that you used in your program. Note that if you fail to justify your choices and explain the code you used in your program, your lecturer may mark your assignment as copied or deduct marks accordingly. All the institute procedures rules and regulations apply to this assignment.
- This assignment should be submitted by not later than **the 18th of January 2020.**
- 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](http://mcast.classter.com))
- The assignment is to be submitted electronically on Moodle as per lecturer's instructions. No hard copy is to be submitted. A copy of the working program (zipped) together with the assignment documentation should be uploaded on Moodle. **node\_modules folder should not be uploaded**

## Brief

A non-profit organization wants to have an online application to manage its members. The members are football players and the following data for each member should be stored:

- Unique id, Name, Surname, Email, Date of Birth, Gender, Mobile including country code, Status (Full-time, Part-time, Amateur), Club

The employees of the organisation must be able to add members to the list through a form. The data must be stored on a Firebase Realtime Database.

Moreover, the employees should also be able to delete any record from the list and update the details of each members. The website should include a menu with a minimum of 3 items (List, Add member, Local Data).

### 1) List of members

View	Identification	Name	Surname	Email	
<a href="#">View</a>	123456M	Joe	Borg	jb@gmail.com	<a href="#">Update</a> <a href="#">Delete</a>

*Figure 1 - List of members*

Employees should be able to view a list with all members with the details (id, name, surname, email) shown above. They should also be able to click on the view button where a detailed view of that respective member showing all details is displayed.


### 2) Updating the details of a member


### Personal Info


Identification:

Name:

Surname:

Date of Birth:  

Nationality:  


Gender:  


### Contact Info

Email:

Telephone/Mobile No.:

### Player Info

Player Status:  

Club:  

[Submit](#)

*Figure 2 - Updating details of a member*

After the user clicks the update button they should be shown a form similar to Figure 2. It's important that the input fields in the form are populated with the current data of that respective member. Once the user clicks the submit button the user should be redirected to that member respective detailed view.

### 3) Add member form.

<b>Name*</b>	<input type="text" value="Joe"/>	✓
<b>Surname*</b>	<input type="text" value="Borg"/>	✓
<b>Date of Birth*</b>	<input type="text" value="01/01/1995"/>	📅
<b>Gender*</b>	<input type="text" value="Male"/>	▼
<b>ID Card No.   Passport No.   Residence Permit *</b>	<input type="text" value="123456M"/>	✓
<b>Mobile*</b>	<input type="text" value="+356 99999999"/>	✓
<b>Email *</b>	<input type="text" value="jb@gmail.com"/>	✓
<b>Password *</b>	<input type="password" value="....."/>	✓
<b>Confirm password *</b>	<input type="password" value="....."/>	✓
<b>Nationality*</b>	<input type="text" value="Maltese"/>	▼
<b>Club</b>	<input type="text" value="NO CLUB M"/>	▼
<b>Status*</b>	<input type="text" value="Full-time Professional"/>	▼
<b>Terms and Conditions*</b>	<input type="checkbox"/> I have read and understood the Terms and Conditions	
	<b>Submit</b>	

Figure 3 – Add a new member

Criteria		Marks	Achieved
KU1	<b>Illustrate how to manipulate the DOM</b>  Use the v-for, v-if and v-else directives to show the list of members within a table (3 marks). If no members are present the table header <b>must not be</b> shown and a text message showing <b>No members yet</b> should be shown instead (2 marks).	5	
KU2	<b>Construct JavaScript techniques to add interactivity in the web application.</b>  Use correctly components including the template, script and style sections to add interactivity to the website (5 marks).	5	
KU3	<b>Construct a fluid layout with fluid images by using a CSS Framework</b>  Use a CSS Framework such as Bootstrap to create a design where the layout including forms (2 marks) and images resize as the window size is changed. Add an item to the menu (Local Data) where data is retrieved from an array (not a DB on the server). The array must include idcard, name, surname and profileImage of at least 3 employees (3 marks).	5	
KU4	<b>Show how a JavaScript framework can be used to build a web application</b>  Use interpolation (1 mark) and two-way data binding using v-model (1 mark) to connect and show data correctly within the web application. Correctly use methods with arguments to pass data from the add/update member forms (3 marks).	5	
KU5	<b>Construct small, self-contained reusable components that can communicate and pass data</b>  Correctly create a component without any syntax errors for each of the following: <ul style="list-style-type: none"> <li>- To display the list (1 mark)</li> <li>- To add a member (1 mark)</li> <li>- To update a member (1 mark)</li> <li>- To delete a member (1 mark)</li> <li>- To show employees data (Use correctly props and custom events were necessary (1 mark).</li> </ul>	5	
KU6	<b>Show how routing allows to navigate through the web application</b>  Create a menu with at least 3 items; one showing the members list, another to add a member and another one to show the data of at least 3 employees. Browser should not refresh when a user navigates from one page to another. Use Vue Router. (3 marks) All pages must have a back button which redirects the user to the previous visited page (2 marks).	5	

KU7	<b>Reproduce the web application online</b>  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).	5	
AA1	<b>Use HTML, CSS and JavaScript to build rich user experiences for web applications using components.</b>  A component to add and another component to delete members should be created (3 mark). Before deleting a member a user should see a confirmation prompt whether he/she wants to proceed and delete (2 marks). Only the selected member must be deleted from the database (2 marks).	7	
AA2	<b>Develop responsive forms with proper client-side validation messages using a CSS Framework</b>  Make use of the 3 <sup>rd</sup> party library Vuelidate to display proper error messages when field is not filled correctly. The following rules apply: <ul style="list-style-type: none"> <li>- Disable the submit button until all fields are filled in correctly (2 marks)</li> <li>- All fields are required (1 mark)</li> <li>- Mobile should have a minimum length and a maximum length of 8 numbers (0.5 marks)</li> <li>- Mobile must be numeric (0.5 marks)</li> </ul> Style the error messages by showing text of error message, border and label in red. (3 marks)	7	
AA3	<b>Apply filters to format the values data retrieved for display to the user.</b>  Use method calls or computed properties to show: <ul style="list-style-type: none"> <li>- the name in uppercase and the surname in lowercase (3 marks)</li> <li>- display only the first 3 letters of the club name (2 marks)</li> <li>- display the + sign in front of the country code. (2 marks)</li> </ul> (Note that filters are no longer supported in VUE 3)	7	
AA4	<b>Use a JavaScript Framework to parse an exchange data format such as JSON data and display in the view</b>  Add a backend by creating a Firebase Realtime database (2 marks). Use the lifecycle hook mounted to immediately render the list of members from the Firebase database on screen(1 mark). Data should be properly stored on the database when a new member is added (2 marks) and when data is updated (2 marks).	7	

AA5	<b>Use validation techniques to validate data input by the user</b>  Use the 3 <sup>rd</sup> party library Vuelidate to perform the following validation using blur event and \$touch method (2 marks): <ul style="list-style-type: none"> <li>- Password must be at least 3 characters long (1 mark)</li> <li>- Confirm password must match the password (use sameAs) (1 mark).</li> <li>- Email must be a valid email (1 mark).</li> <li>- IdCard must contain only alphanumeric (1 mark).</li> <li>- Name and surname must accept only alphabet characters (1 mark).</li> </ul>	7	
SE1	<b>Develop a web application front-end using common concepts such as data binding, events and methods.</b>  Use data binding and events to trigger the methods to add, update and delete data from the database (6 marks). Correctly use the Vue modifier <b>trim</b> on all data input by the user (2 marks). Moreover use the Vue modifier <b>number</b> on the mobile number (2 marks).	10	
SE2	<b>Design a responsive mobile-first web application.</b>  Use a CSS Framework to have a complete responsive website. Using a template such as a bootstrap template and theme is also permitted and recommended. The top navigation menu must become a hamburger menu once the website is viewed on a mobile phone (3 marks). All html forms must be responsive (2 marks). Modals should display properly on different screen sizes before deleting a user. The table list should also display properly when viewed in different sizes (2 marks). Once you have deployed your website online check your url on <a href="https://search.google.com/test/mobile-friendly">https://search.google.com/test/mobile-friendly</a> . Fix possible issues(3 marks).	10	
SE3	<b>Create basic transitions and animations.</b>  Use the transition Vue component to build a simple transition on the web applicaiton. E.g. When the modal enters the screen (enter transition) and leaves from the screen (leaves transition).	10	

\*\*\*\*\*End of Assignment\*\*\*\*\*