

|                            |  |                            |     |
|----------------------------|--|----------------------------|-----|
| <b>Title</b>               | Vue.js web application   | <b>Total Possible Mark</b> | 100 |
| <b>Type</b>                | Web Development  | <b>Weighting</b>           | 35  |
| <b>Learning Outcome(s)</b> | <ol style="list-style-type: none"> <li>2. <b>Examine and apply web development programming languages in the development of a website.</b></li> <li>3. <b>Create and apply multimedia solutions in website development.</b></li> <li>4. <b>Examine current web practices used in the industry and apply this to the development of a website</b></li> </ol> |                            |     |

## TASK

The aim of this assignment is to develop a Vue.js web application that displays a list of apparel with various colours and enables users to add products to the cart, review them, and see the cart's contents. You will be working with components, props, methods, computed properties, events, conditionals, and styling.

## CONDITIONS

- **Refer to the Timetable on Moodle for Assessment Due date.**
- Individual assessment
- 1.5 hours per week will be allocated to you in class-time, for which you must be present.
- Prepare a compressed folder (e.g., ZIP) with all files.
- Please use the following naming convention:
  - A3\_yourName

## TASK DETAILS / INSTRUCTIONS / STEPS / GUIDELINES

### Task 1: Create Vue Event Bus

- **Objective:**
  - Create an event bus to handle communication between components.
- **Actions:**
  - Create a new Vue instance named eventBus.
  - Clear comments and consistent coding style (code indentation)
- **Outcome:**
  - A Vue instance serving as the event bus will be created, enabling different components to emit and listen to events.

### Task 2: Develop product Component

- **Objective:**
  - Develop a product component to display the product details and handle product-related actions.
- **Actions:**
  - Define props to receive premium status.
  - Create a template containing product information, variants, and an add-to-cart button.
  - Define data to handle product information, variants, and reviews.
  - Define methods to handle actions like adding to cart and updating the selected product variant.
  - Define computed properties to calculate image, stock status, price, product status, and shipping fee.
  - Handle mounting of the component to listen to the event bus.
  - Clear comments and consistent coding style (code indentation)
- **Outcome:**
  - A fully functional product component displaying dynamic product information and handling user interactions.

### Task 3: Develop add-review Component

- **Objective:**
  - Develop an add-review component to handle user reviews.
- **Actions:**
  - Create a template containing form fields to take user input for a review.
  - Define data to handle the user's name, review text, rating, and errors.
  - Define a method to handle form submission and validate user input.
  - Clear comments and consistent coding style (code indentation)
- **Outcome:**

- A fully functional add-review component, collecting user reviews, validating input, and emitting events.

#### **Task 4: Develop product-tabs Component**

- **Objective:**
  - Develop a product-tabs component to switch between reviews and adding a review.
- **Actions:**
  - Create a template containing tabs for Reviews and Add Review.
  - Define data to handle the currently selected tab.
  - Define a method to update the selected tab.
  - Clear comments and consistent coding style (code indentation).
- **Outcome:**
  - A fully functional product-tabs component allowing users to navigate between viewing reviews and adding a review.

#### **Task 5: Develop cart-content Component**

- **Objective:**
  - Develop a cart-content component to display cart items.
- **Actions:**
  - Create a template to show the cart items in a table format and provide the option to remove items from the cart.
  - Define methods to handle hiding cart content and deleting items from the cart.
  - Define a computed property to calculate the total price of items in the cart.
  - Clear comments and consistent coding style (code indentation)
- **Outcome:**
  - A fully functional cart-content component showing cart items and total price with options to remove items from the cart.

#### **Task 6: Create Main Vue Instance**

- **Objective:**
  - Develop the main Vue instance to manage the application.
- **Actions:**
  - Define data to handle premium status, cart visibility, and cart items.
  - Define methods to handle updates to the cart and toggle cart content visibility.
  - Attach the Vue instance to an element in the HTML.
  - Clear comments and consistent coding style (code indentation)
- **Outcome:**
  - A configured main Vue instance managing the application state and handling top-level actions.

**AROMATAWAI TUATORU | ASSESSMENT 3 - MARKING CRITERIA**

| Task                              | Criteria                     | Description   | Maximum Points |
|-----------------------------------|------------------------------|---|----------------|
| <b>1: Vue Event Bus</b>           |                              |   | <b>10</b>      |
|                                   | Correct Implementation       | Proper instantiation and configuration of the event bus   | 7              |
|                                   | Documentation & Code Quality | Clear comments and consistent coding style  | 3              |
| <b>2: product Component</b>       |                              |   | <b>25</b>      |
|                                   | Correct Implementation       | Proper use of props, data, methods, and computed properties                                     | 10             |
|                                   | User Interaction             | Correct handling of user actions like adding to cart and selecting variants                     | 5              |
|                                   | Template Implementation      | Proper implementation of the template   | 5              |
|                                   | Documentation & Code Quality | Clear comments and consistent coding style  | 5              |
| <b>3: add-review Component</b>    |                              |   | <b>15</b>      |
|                                   | Correct Implementation       | Proper use of data and methods for form submission and validation                               | 7              |
|                                   | User Interaction             | Correct handling and validation of user inputs  | 3              |
|                                   | Template Implementation      | Proper implementation of the template   | 3              |
|                                   | Documentation & Code Quality | Clear comments and consistent coding style  | 2              |
| <b>4: product-tabs Component</b>  |                              |   | <b>10</b>      |
|                                   | Correct Implementation       | Proper use of data and methods for tab selection  | 4              |
|                                   | User Interaction             | Correct handling of user actions to switch tabs   | 2              |
|                                   | Template Implementation      | Proper implementation of the template   | 2              |
|                                   | Documentation & Code Quality | Clear comments and consistent coding style  | 2              |
| <b>5: cart-content Component</b>  |                              |   | <b>20</b>      |
|                                   | Correct Implementation       | Proper use of methods and computed properties for cart interactions and total price calculation | 6              |
|                                   | User Interaction             | Correct handling of user actions like removing items and toggling cart visibility               | 7              |
|                                   | Template Implementation      | Proper implementation of the template   | 4              |
|                                   | Documentation & Code Quality | Clear comments and consistent coding style  | 3              |
| <b>6: Main Vue Instance</b>       |                              |   | <b>20</b>      |
|                                   | Correct Implementation       | Proper use of data and methods for managing application state and handling top-level actions    | 8              |
|                                   | Integration                  | Correct integration and interaction between all components                                      | 6              |
|                                   | Attachment                   | Proper attachment of Vue instance to HTML element   | 4              |
|                                   | Documentation & Code Quality | Clear comments and consistent coding style  | 2              |
| <b>Total marks for assessment</b> |                              |   | <b>100</b>     |

