Course Name: Web Programming 2

Course Code: COMP1842

Term: Spring 2024

Full Name: Nguyen Vinh Khang

ID Number: GCS220252

**Table of Contents**

[1. Shoppe: 3](#_Toc168673309)

[1.1. Image Switcher: 3](#_Toc168673310)

[1.2. Simple cart: 3](#_Toc168673311)

[2. Calculator: 4](#_Toc168673312)

[2.1. Calcu: 4](#_Toc168673313)

[2.2. Keyboard event listener: 4](#_Toc168673314)

# Shoppe:

## Image Switcher:

const app = new Vue({

        el: "#vue\_app",

        data: {

          inpText: "initial text",

          sources: [

            "https://images.pexels.com/photos/1294886/pexels-photo-1294886.jpeg",

            "https://images.pexels.com/photos/1366998/pexels-photo-1366998.jpeg",

            "https://images.pexels.com/photos/3510/hand-apple-iphone-smartphone.jpg",

          ],

          img\_src: 1,

        },

      });

el**: "#vue\_app"**: This tells Vue to control the element with the ID "vue\_app".

data:

* **sources**: An array containing image URLs for the product.
* **img\_src**: An integer representing the index of the currently displayed image in the sources array.

Vue uses directives to bind data to HTML elements:

* **v-bind:src="sources[img\_src]"**: This directive dynamically sets the src attribute of the image with class "mainImg" based on the value of img\_src.
* **v-on:change="img\_src=0"**: This directive binds the change event of a radio button to a function that sets img\_src to 0, displaying the first image. A similar logic applies to the other radio buttons.

Clicking a radio button triggers the corresponding v-on:change event, updating img\_src and displaying the selected image.

## Simple cart:

const cart\_calcing = new Vue({

        el: "#cart\_calcing",

        data: {

          cart: 0,

          inp: 0,

        },

        methods: {

          calc: function() {

            if (!isNaN(parseInt(this.inp))) {

              this.cart += parseInt(this.inp);

            } else {

              this.cart += 0;

            }

          },

        },

      });

**data**:

* **cart**: Stores the current total quantity of items.
* **inp**: Holds the user input for the desired quantity.

The **v-model="inp"** directive binds the input field's value to the inp data property.

The **@click="calc"** event on the "Add to cart" button triggers the calc method.

The **calc** method checks for valid user input (a number) and updates the cart value accordingly.

# Calculator:

## Calcu:

el: "#calcu" - manage the element with the ID "calcu" (the entire calculator table).

data:

* + - **result**: Stores the current display value (empty string initially).
    - **isCompleted**: A flag indicating if the last operation resulted in a complete answer (initially false).

methods:

**calc**(event): This function is triggered when clicking a button. It checks the clicked button's value and updates the result accordingly:

* + - * + If the value is "=": Evaluate the expression in the result using eval and update the result with the answer. Sets isCompleted to true.
* If the value is "c": Resets the calculator by clearing the result and setting is completed to false.
* If the value is an operator (+, -, \*, /):
  + Prevents adding multiple operators in a row.
  + If the last character in the result is also an operator, replace it with the current operator.
    - * + Checks if the value is ".": Prevents adding multiple decimal points in a row or within a number segment after an operator.**reset**(): Clears the calculator display and resets the isCompleted flag.

**mounted**(): This lifecycle hook is called once after the Vue instance is mounted (the DOM is ready). It attaches click event listeners to all calculator buttons using addEventListener and loops through them using forEach. These listeners call the calc function whenever a button is clicked.

## Keyboard event listener:

This listener is added outside of Vue and captures keyboard presses. It checks the pressed key and:

* + If "Enter" is pressed, it triggers a simulated button click with value "=" using calcu.calc().
  + If "Escape" is pressed, trigger a simulated button click with value "c" using calcu.calc().
  + If the pressed key is an operator, number, or ".", triggers a simulated button click with the corresponding value using calcu.calc().