https://github.com/tintran999/Week4

GIT LINK

```html
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <link rel="stylesheet" href="assets/styles.css" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Tran Duc Tin week4</title>
    <script src="https://cdn.jsdelivr.net/npm/vue@2.7.14/dist/vue.js"></script>
</head>

<body>
    <div class="nav-bar"></div>

    <div id="app">
        <div class="cart">
          <p>Cart({{ cart.length }})</p>
        </div>
        <product :premium="premium" @add-to-cart="updateCart" @remove-from-cart="removeItem"></product>
      </div>

    <script src="main.js"></script>
</body>

</html>
```

First, I create some basic HTML code line to structure the webpage and include necessary elements such as the navigation bar, shopping cart, and product components. This helps in organizing the content and making the webpage functional.

Then, I define Vue components for ‘product-details’ and ‘product’. The ‘product-details’ component takes an array of details as a prop and renders a list of these details using the v-for directive. The ‘product’ component takes a Boolean prop ‘premium’ and includes an image tag to display the product image. This helps in modularizing the code and making it reusable

```
                                                        //Challenge 8->10

Vue.component('product-details', { //Challenge 8
    props: {
      details: {
        type: Array,
        required: true
      }
    },
    template: `
      <ul>
        <li v-for="detail in details">{{ detail }}</li>
      </ul>
    `
  })


  Vue.component('product', {
    props: {
      premium: {
        type: Boolean,
        required: true
      }
    },
    template: `
     <div class="product">

        <div class="product-image">
          <img :src="image" />
        </div>
```

Next, I define the structure of the product component. This includes the product image, product information such as availability (in stock or out of stock), pricing, and shipping details. I also add functionality for adding the product to the cart and removing it from the cart using JavaScript functions. This ensures that the webpage is interactive and provides a seamless

shopping experience for users.

```
template:
<div class="product">

  <div class="product-image">
    <img :src="image" />
  </div>

  <div class="product-info">
      <h1>{{ title }}</h1>
      <p v-if="inStock">In Stock</p>
      <p v-else>Out of Stock</p>
      <p>Shipping: {{ shipping }}</p>

      <product-details :details="details"></product-details> <!-- Challenge 8 -->

      <div class="color-box"
           v-for="(variant, index) in variants"
           :key="variant.variantId"
           :style="{ backgroundColor: variant.variantColor }"
           @mouseover="updateProduct(index)"
           >
      </div>

      <button v-on:click="addToCart"
        :disabled="!inStock"
        :class="{ disabledButton: !inStock }"
        >
      Add to cart
      </button>

      <button @click="removeFromCart">   <!-- Challenge 9 -->
```

```
        <button @click="removeFromCart">   <!-- Challenge 9 -->
        Remove from cart
        </button>

      </div>

      <div>
        <p v-if="!reviews.length">There are no reviews yet.</p>
        <ul v-else>
          <li v-for="(review, index) in reviews" :key="index">
            <p>{{ review.name }}</p>
            <p>Rating:{{ review.rating }}</p>
            <p>{{ review.review }}</p>
          </li>
        </ul>
      </div>

      <product-review @review-submitted="addReview"></product-review>

    </div>
  `,
  data() {
    return {
      product: 'Socks',
      brand: 'Vue Mastery',
      selectedVariant: 0,
      details: ['80% cotton', '20% polyester', 'Gender-neutral'],
      variants: [
        {
```

Then, I add functionality for removing items from the cart and displaying product reviews. The ‘removeFromCart’ method is triggered when the ‘Remove from cart’ button is clicked. I also include a section to display reviews if there are any, and a form for submitting new reviews. This enhances the user experience by allowing them to manage their cart and read or submit reviews for products

```
      <b>Please correct the following error(s):</b>
      <ul>
        <li v-for="error in errors">{{ error }}</li>
      </ul>
    </p>

    <p>
      <label for="name">Name:</label>
      <input id="name" v-model="name">
    </p>

    <p>
      <label for="review">Review:</label>
      <textarea id="review" v-model="review"></textarea>
    </p>

    <p>
      <label for="rating">Rating:</label>
      <select id="rating" v-model.number="rating">
        <option>5</option>
        <option>4</option>
        <option>3</option>
        <option>2</option>
        <option>1</option>
      </select>
    </p>

    <p>Would you recommend this product?</p>
    <label>
      Yes
```

Next, I create a form for submitting reviews and ratings. The form includes input fields for the user's name, review text, rating, and a question asking if they would recommend the product. I also add error handling to display any validation errors. This ensures that users can provide feedback on the products and helps in collecting valuable reviews.

```
var app = new Vue({
    data: {
      premium: true,
      cart: []
    },
    methods: {
      updateCart(id) {
        this.cart.push(id)
      },
      removeItem(id) { //Challenge 9
        for(var i = this.cart.length - 1; i >= 0; i--) {
          if (this.cart[i] === id) {
            this.cart.splice(i, 1);
          }
        }
      }
    }
})
```

Finally, I create a Vue instance to manage the application's state and methods. The 'app' object is created using the 'new Vue' constructor, and it contains data properties such as 'premium' set to true and 'cart' initialized as an empty array. I also define methods like 'updateCart' to add items to the cart and 'removeItem' to remove items from the cart. This ensures that the application is interactive and can handle user actions efficiently

The Result

Vue Mastery Socks

In Stock

Shipping: Free

- 80% cotton
- 20% polyester
- Gender-neutral

Add to cart | Remove from cart

There are no reviews yet.

Name:

Review:

Rating:

Would you recommend this product?

Yes

No

Submit