

# Title page

Pham Duc Thang,

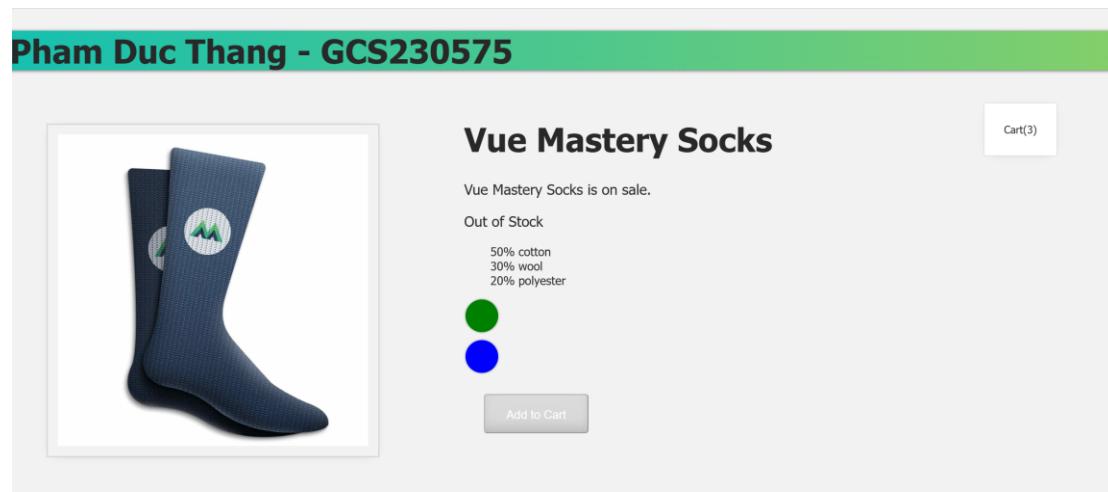
ID: 001407356,

Course Title: Web Programming 2,  
COMP1842

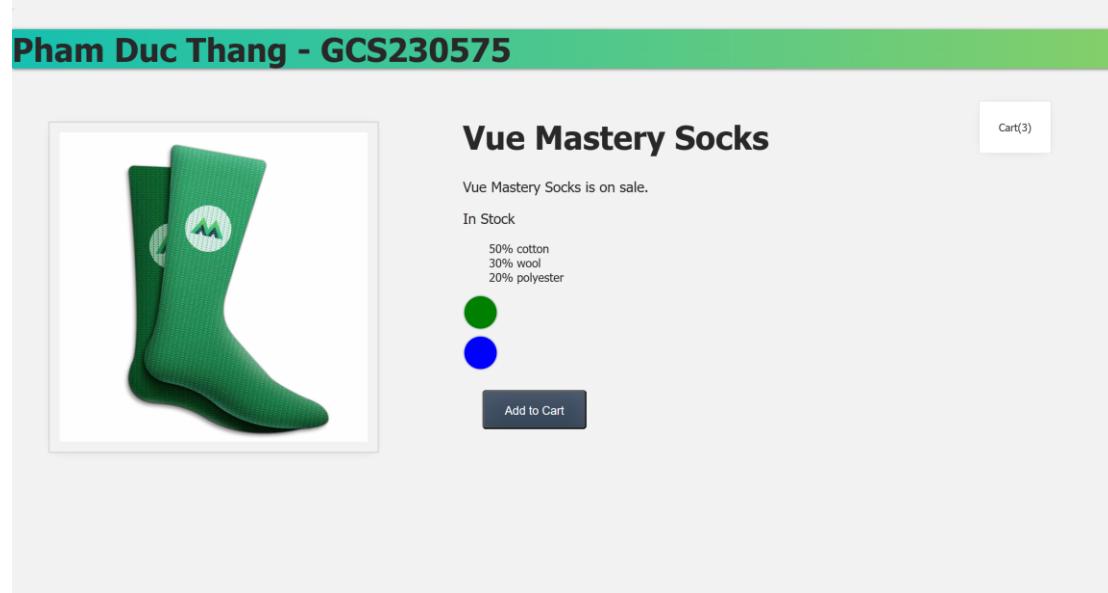
**Note: All we need in this logbook is screenshots.**  
**All code must be uploaded separately in a ZIP file.**

## Weekly Tasks

Logbook 3 - Week 3 tasks



The screenshot shows a product page for 'Vue Mastery Socks'. The title 'Pham Duc Thang - GCS230575' is at the top. The main image shows a pair of dark blue socks with a white circular logo featuring a stylized 'M'. Below the image, the text 'Vue Mastery Socks' is displayed, followed by 'Vue Mastery Socks is on sale.' and 'Out of Stock'. It lists the material composition: 50% cotton, 30% wool, and 20% polyester. There are two color options shown: green and blue. A grey 'Add to Cart' button is at the bottom right. In the top right corner, there is a 'Cart(3)' button.

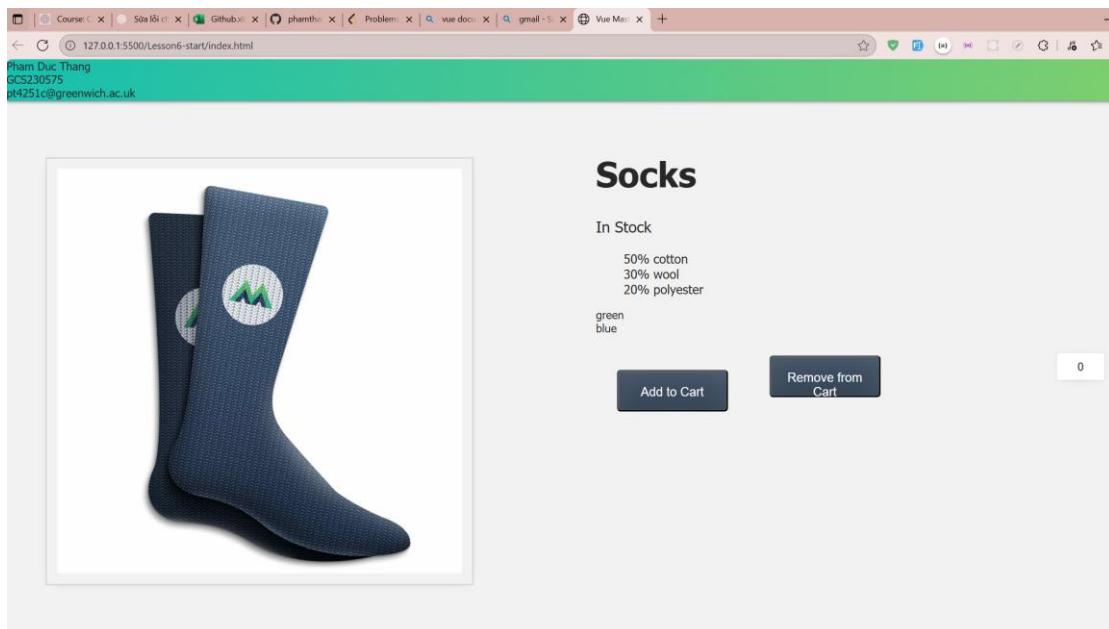
  


The second screenshot shows the same product page for 'Vue Mastery Socks' but with green socks instead of blue. The title 'Pham Duc Thang - GCS230575' is at the top. The main image shows a pair of green socks with a white circular logo featuring a stylized 'M'. Below the image, the text 'Vue Mastery Socks' is displayed, followed by 'Vue Mastery Socks is on sale.' and 'In Stock'. It lists the material composition: 50% cotton, 30% wool, and 20% polyester. There are two color options shown: green and blue. A dark grey 'Add to Cart' button is at the bottom right. In the top right corner, there is a 'Cart(3)' button.

### Lab 3 – week 3 – lesson 8. Computed Properties

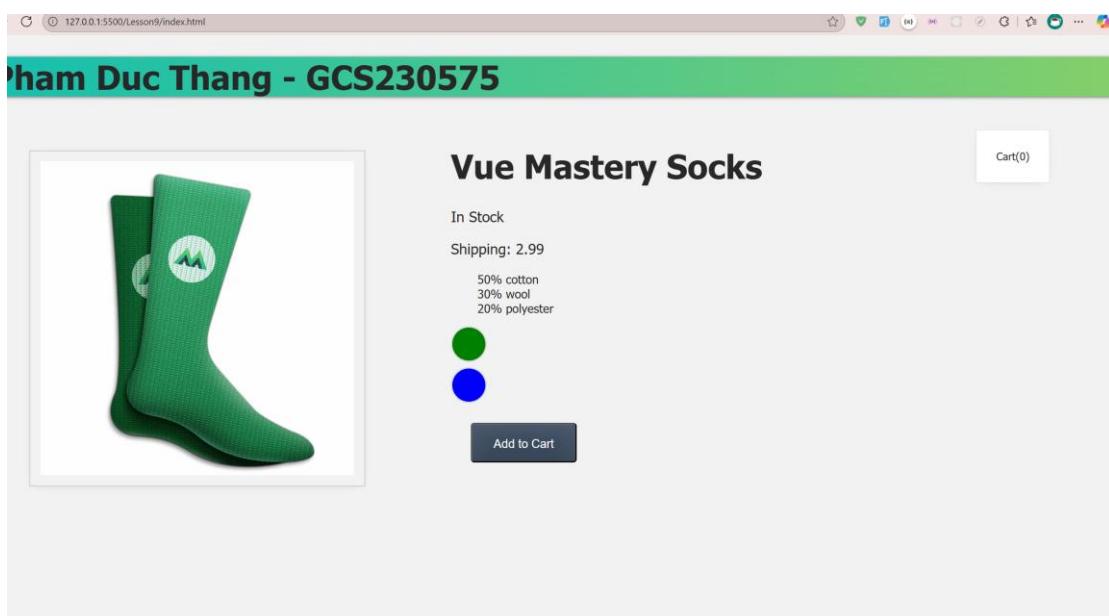
Computed properties are attributes in Vue.js that are calculated based on reactive data. They allow to perform operations or process data without having to call a function each time, helping to keep the code clean and improve performance.

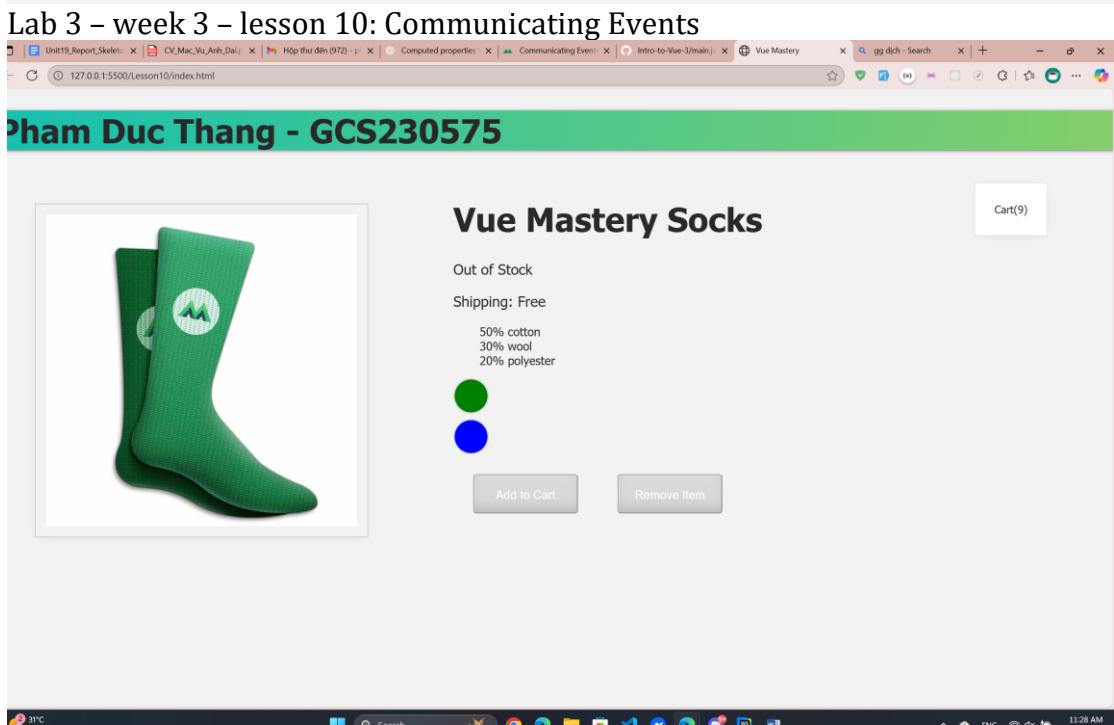
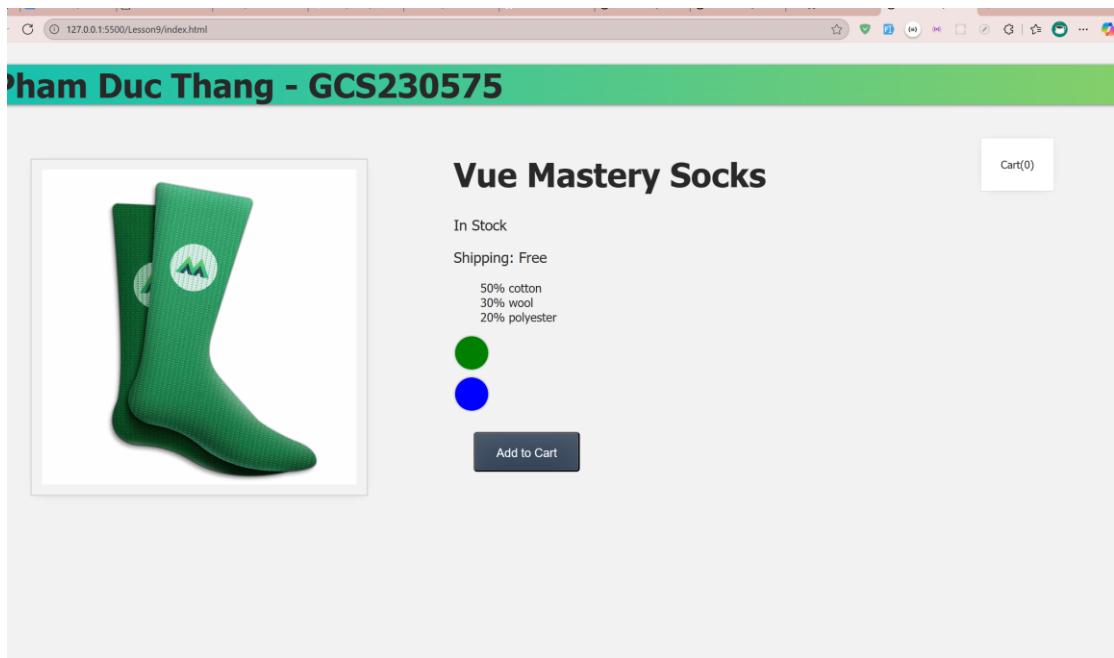
Unlike methods, computed properties are only recalculated when their dependent data changes, which helps optimize performance.

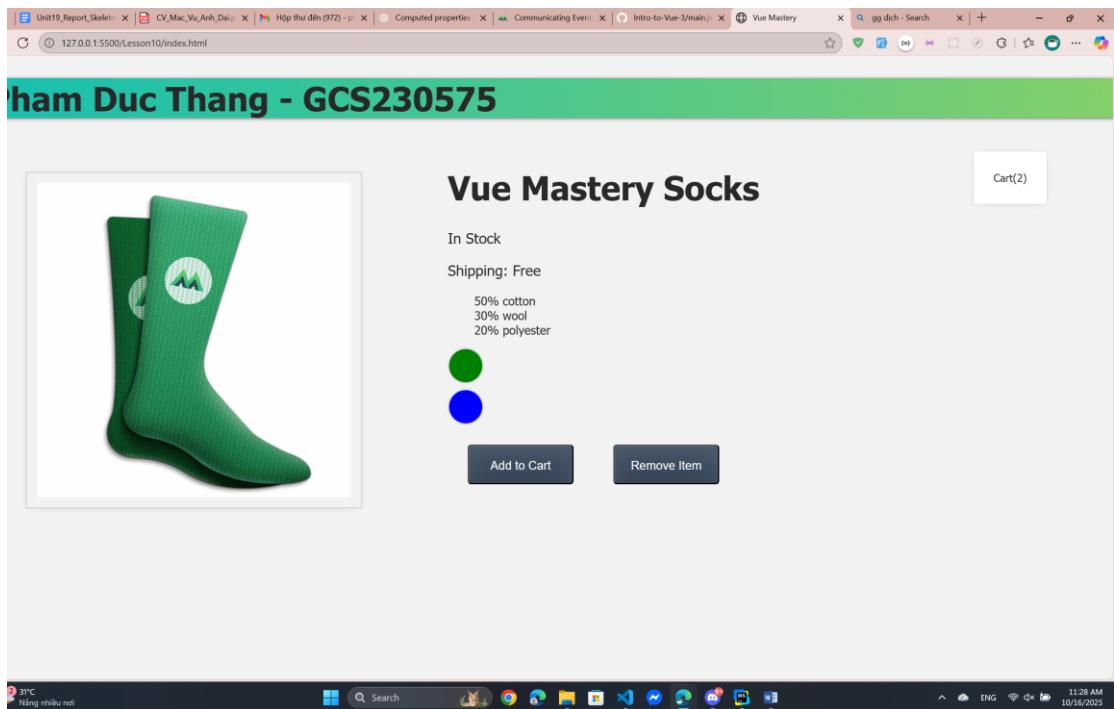


Lab 3 – week 3 – lesson 9 : Components and Props

- **Components:** Break complex UI into small, reusable parts to keep `main.js` clean.
- **Props:** Custom attributes to pass data from parent to child components; like React props, `...rest` can be used to gather remaining props.







- **Problem:** Components have their own scope, so variables like `cart` in `main.js` cannot be accessed directly inside a component like `ProductDisplay`. The solution is using **events**.
- **Goal:** Understand how components communicate in Vue 3 using events.
- **Child → Parent:** Use `this.$emit('event-name', data)` to send events from child to parent.
- **Parent → Child:** Pass data via `props`.

Example:

Child component: `this.$emit('update', newValue);`

Parent component:

```
<ChildComponent @update="handleUpdate" />
```

methods: {

```
  handleUpdate(newValue) {
```

```
    this.value = newValue;
```

```
}
```

```
}
```

**Benefit:** Keeps components **reusable** and helps manage **state efficiently**.