

Title page

Pham Duc Thang,

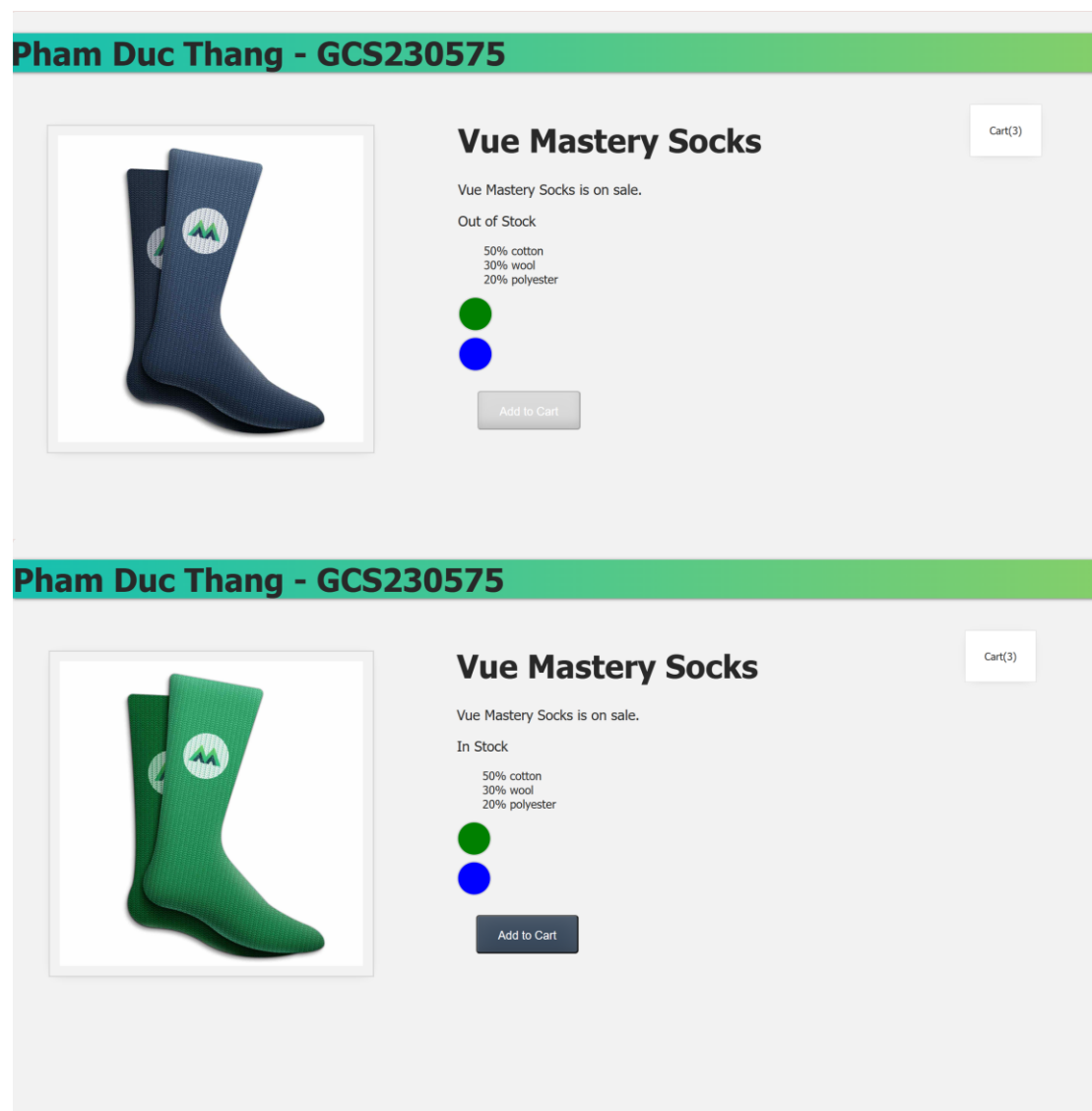
ID: 001407356,

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

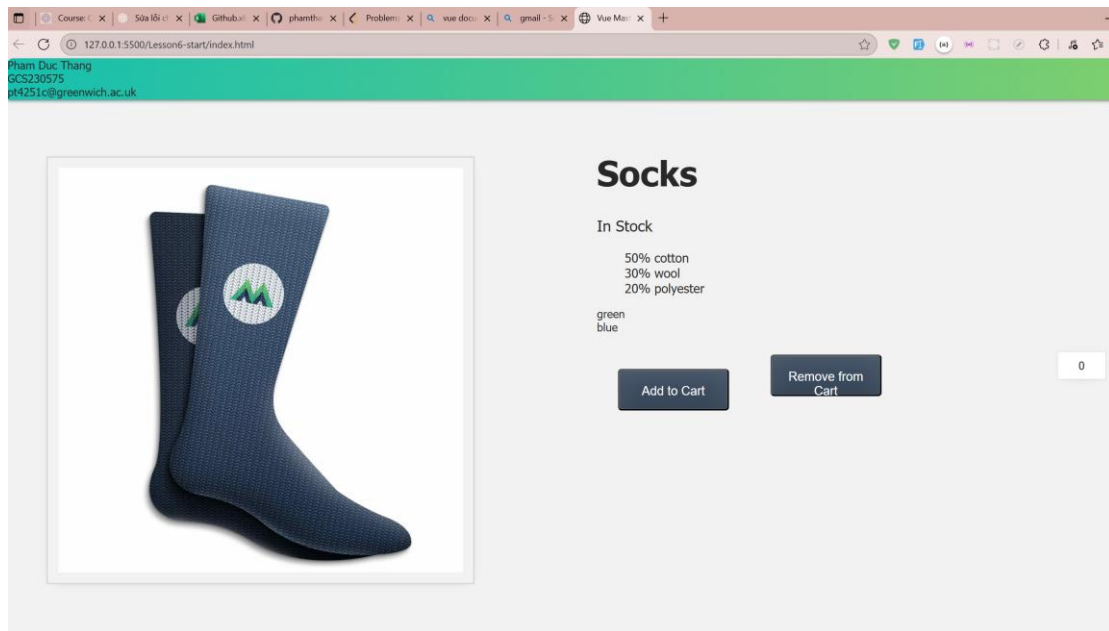
Logbook 3 - Week 3 tasks



Lab 3 – week 3 – lesson 8. Computed Properties

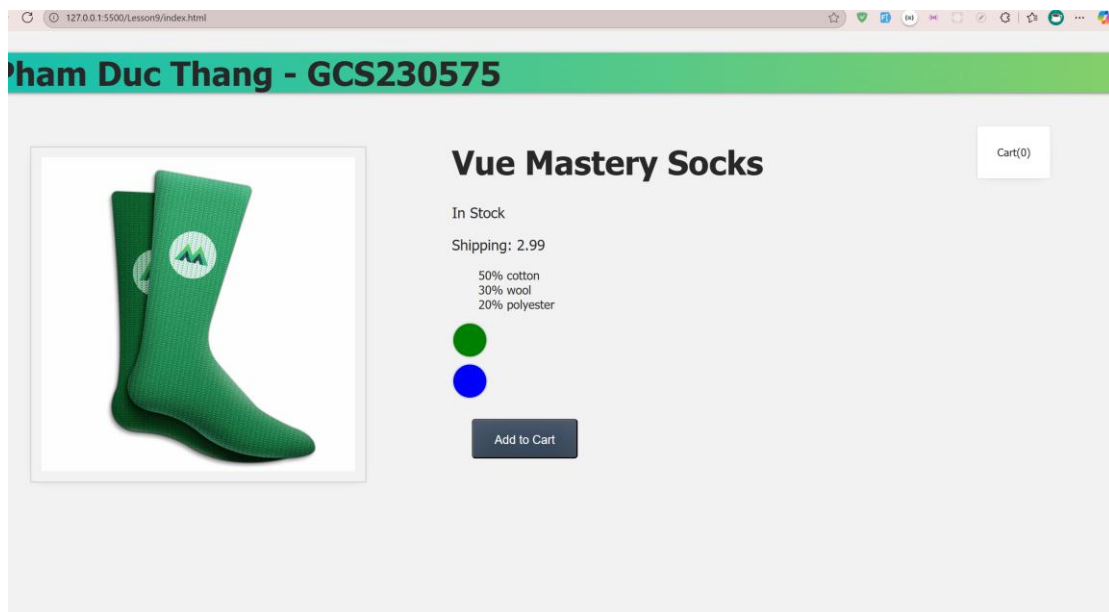
Computed properties are attributes in Vue.js that are calculated based on reactive data. They allow to perform operations or process data without having to call a function each time, helping to keep the code clean and improve performance.

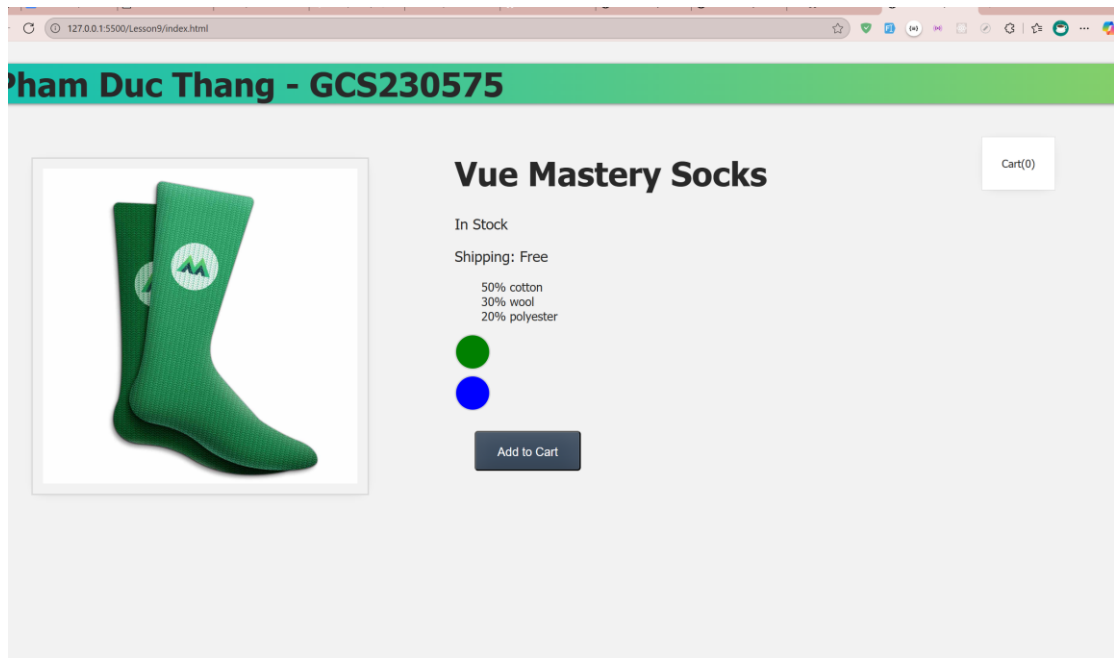
Unlike methods, computed properties are only recalculated when their dependent data changes, which helps optimize performance.



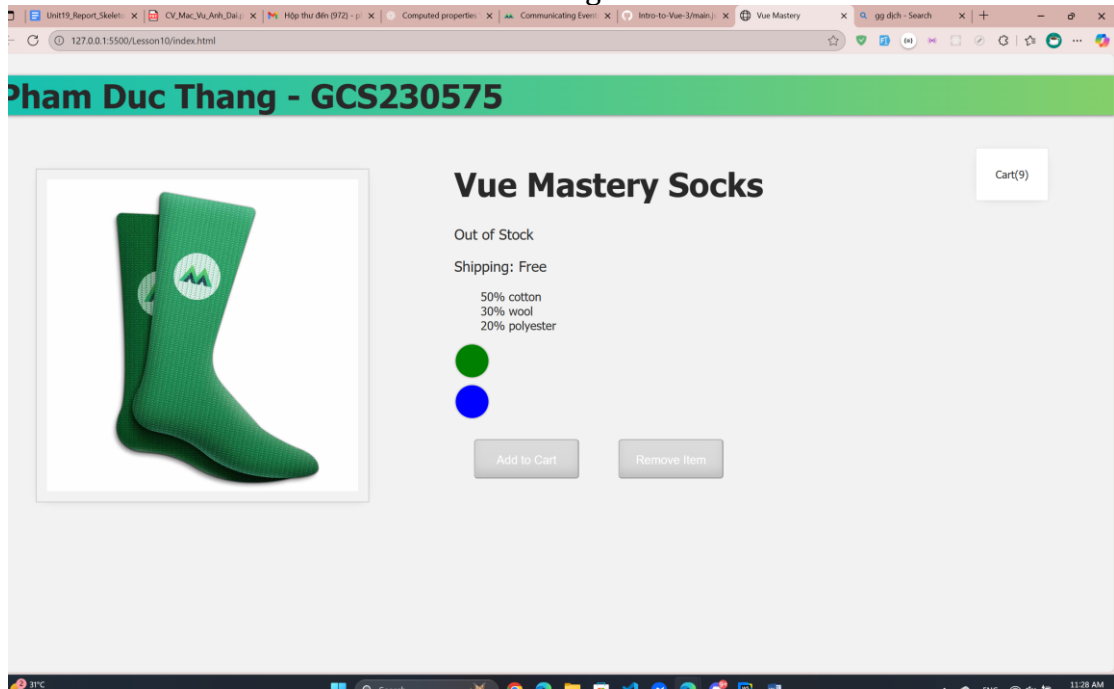
Lab 3 – week 3 – lesson 9 : Components and Props

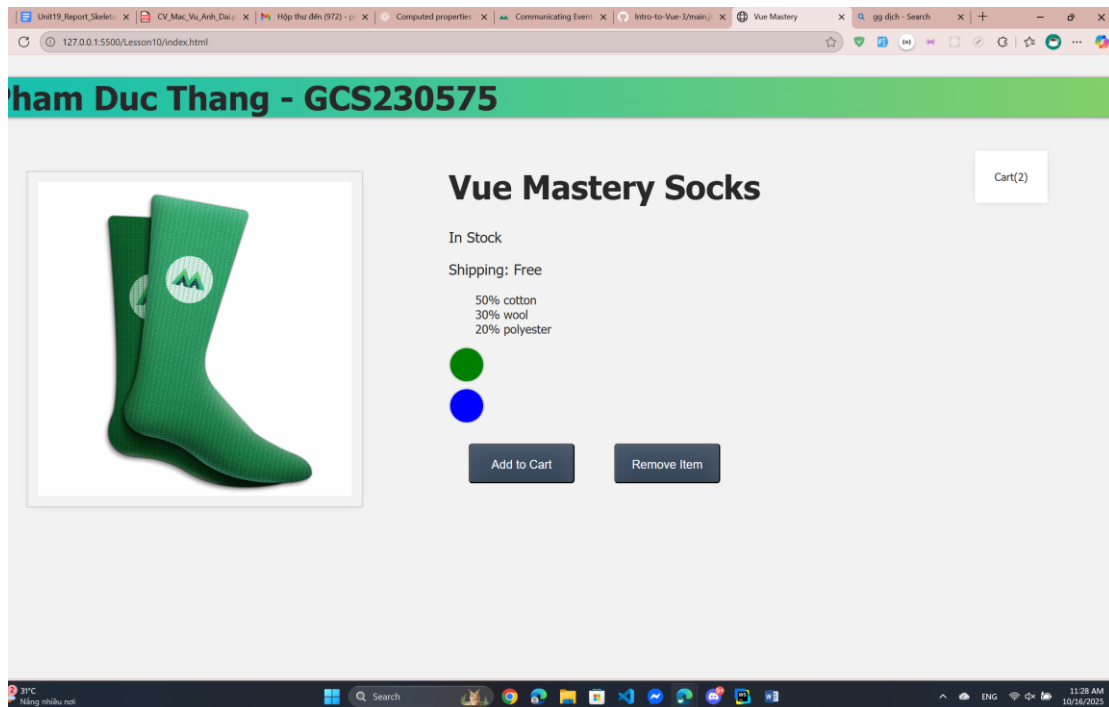
- **Components:** Break complex UI into small, reusable parts to keep `main.js` clean.
- **Props:** Custom attributes to pass data from parent to child components; like React props, `...rest` can be used to gather remaining props.





Lab 3 – week 3 – lesson 10: Communicating Events





- **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**.