

Lab 3 Understanding

These lessons helped me understand how Vue handles logic, structure, and communication in a clean way. Computed properties stood out because they let Vue handle calculations for us instead of putting that logic directly in the template. For example, instead of combining the brand and product name in the HTML, a computed property like title does that work. The same idea applies to image and inStock, which automatically update based on the selected variant. I like that Vue only recalculates these values when the data they depend on changes, which makes the app more efficient.

Breaking the app into components also made a lot more sense after this lesson. Moving everything into a product-display component keeps product-related code organized and reusable. Each product component works independently, even though they all share the same structure. That made it clearer why components are such a big part of Vue apps.

Props helped me understand how data flows from a parent to a child component. Since components are isolated, they can't directly access parent data. Passing the premium value as a prop allows the product component to decide whether shipping is free or not without breaking that separation.

Emitting events finally tied everything together. Because the cart lives in the parent, the product component can't update it directly. Emitting an add-to-cart event lets the component notify the parent when the button is clicked. Passing the product ID with the event makes the cart more realistic.

Overall, these lessons showed me how Vue keeps data flow organized and predictable.

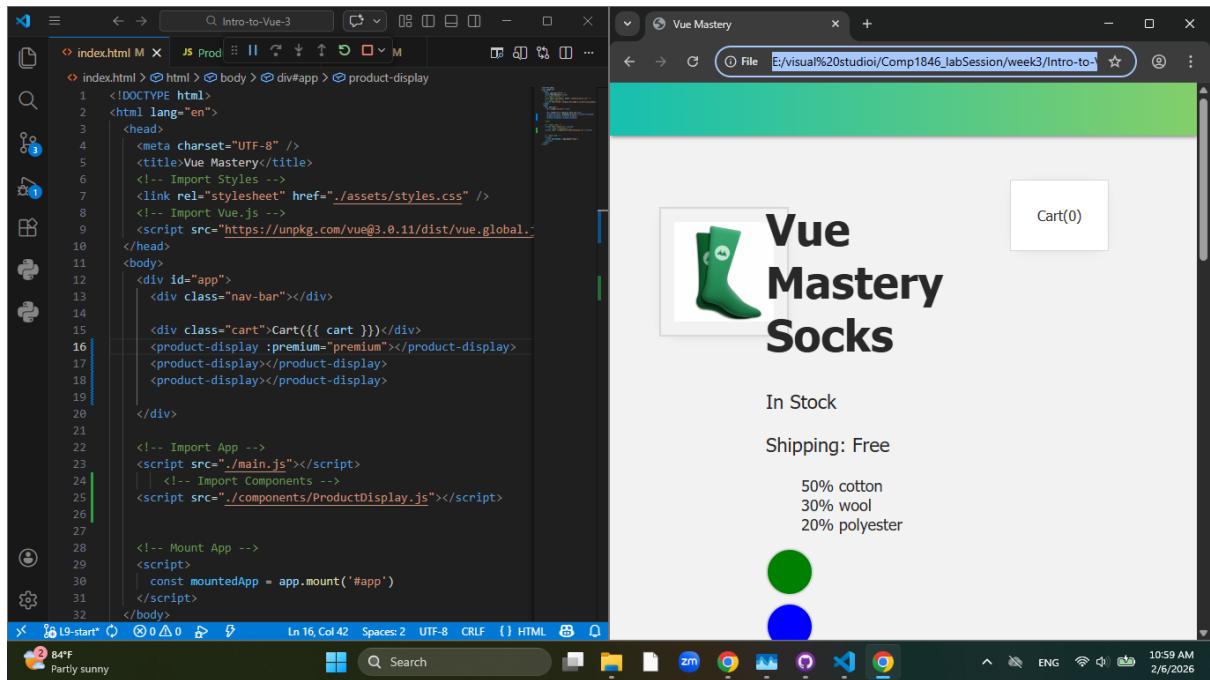
Task of lesson 8 picture:

The screenshot shows a Vue.js application running in a browser. The code editor on the left displays `index.html` and `main.js`. The browser window shows a product page for 'Vue Mastery Socks'. The product image is a blue sock with a small green tag. The title is 'Vue Mastery Socks'. Below the title, it says 'Out of Stock'. To the right is a table showing the fabric composition: 50% cotton, 30% wool, and 20% polyester. There are two colored circles below the table: a green one and a blue one.

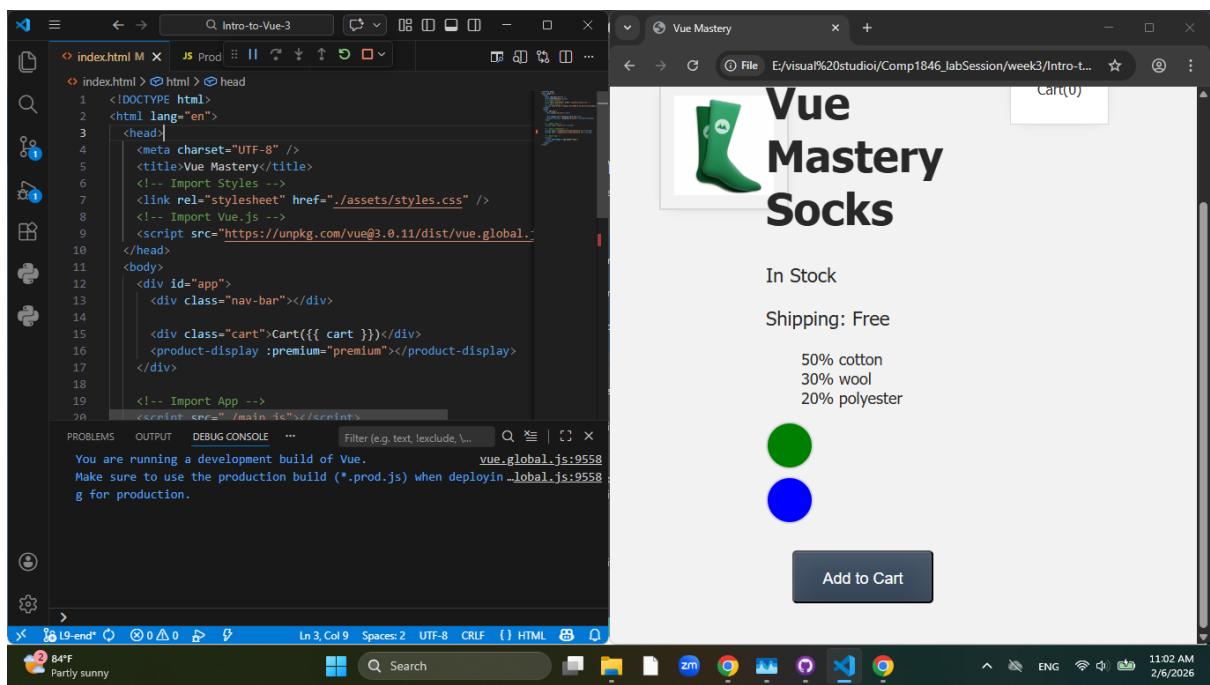
Challenge of lesson 8 picture:

The screenshot shows a Vue.js application running in a browser. The code editor on the left displays `index.html` and `main.js`. The browser window shows a product page for 'Vue Mastery Socks'. The product image is a green sock. The title is 'Vue Mastery Socks'. Below the title, it says 'Vue Mastery Socks is on sale.' Then it says 'In Stock'. To the right is a table showing the fabric composition: 50% cotton, 30% wool, and 20% polyester. There are two colored circles below the table: a green one and a blue one. At the bottom right is a large 'Add to Cart' button.

Task of lesson 9 picture:



Challenge of lesson 9 picture:



Task of lesson 10 picture:

The screenshot shows a development environment with a code editor and a browser window. The code editor displays the file `index.html` with the following content:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <title>Vue Mastery</title>
    <!-- Import Styles -->
    <link rel="stylesheet" href=".assets/styles.css" />
    <!-- Import Vue.js -->
    <script src="https://unpkg.com/vue@3.0.11/dist/vue.global.js" type="text/javascript"></script>
  </head>
  <body>
    <div id="app">
      <div class="cart">Cart({{ cart.length }})</div>
      <product-display :premium="premium" @add-to-cart="updateCart" @remove-from-cart="removeById">
        <!-- Import App -->
        <script src=".main.js"></script>
        <!-- Import Components -->
        <script src=".components/ProductDisplay.js"></script>
        <!-- Mount App -->
        <script>
          const mountedApp = app.mount('#app')
        </script>
      </product-display>
    </div>
    <!-- Import App -->
    <script src=".main.js"></script>
  </body>
</html>
```

The browser window shows a product page for "Vue Mastery Socks". The page includes a green sock icon, the title "Vue Mastery Socks", a "Cart(2)" button, and a "In Stock" message. It also lists the material composition: "50% cotton, 30% wool, 20% polyester". Below the composition are two circular color swatches, one green and one blue. The bottom right of the page has a "Remove Item" button.

Challenge of lesson 10 picture:

The screenshot shows a development environment with a code editor and a browser window. The code editor displays the file `index.html` with the following content:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <title>Vue Mastery</title>
    <!-- Import Styles -->
    <link rel="stylesheet" href=".assets/styles.css" />
    <!-- Import Vue.js -->
    <script src="https://unpkg.com/vue@3.0.11/dist/vue.global.js" type="text/javascript"></script>
  </head>
  <body>
    <div id="app">
      <product-display
        @add-to-cart="updateCart"
        @remove-from-cart="removeById">
        <!-- Import App -->
        <script src=".main.js"></script>
        <!-- Import Components -->
        <script src=".components/ProductDisplay.js"></script>
        <!-- Mount App -->
        <script>
          const mountedApp = app.mount('#app')
        </script>
      </product-display>
    </div>
    <!-- Import App -->
    <script src=".main.js"></script>
  </body>
</html>
```

The browser window shows a simplified product page for "Socks". The page includes the title "Socks", a "In Stock" message, a "Shipping: Free" message, the material composition "50% cotton, 30% wool, 20% polyester", and two circular color swatches (green and blue). At the bottom right, there are "Add to Cart" and "Remove Item" buttons.