

Lesson 5:



The screenshot shows a Vue.js application interface. At the top, there's a navigation bar with icons for file operations and a 'School' tab. Below it is a header with the text '# styles.css ..._lesson5_U'. The main content area displays a product card for a 'sock'. The product image is a pair of green socks with a small logo on the toe. The title 'sock' is displayed in large bold letters. Below the title, the text 'In Stock' is shown. A detailed description follows: '50% cotton', '30% wool', and '20% polyester'. Underneath, color options 'green' and 'blue' are listed. A section titled 'Available Sizes:' lists sizes 'S', 'M', 'L', and 'XL'. To the right of the product card is a code editor window showing the corresponding Vue.js component code.

```
1  # styles.css ..._lesson5_U
2  ...
3  ...
4  ...
5  ...
6  ...
7  ...
8  ...
9  ...
10 ...
11 ...
12 ...
13 ...
14 ...
15 ...
16 ...
17 ...
18 ...
19 ...
20 ...
21 ...
22 ...
23 ...
24 ...
25 ...
26 ...
27 ...
28 ...
29 ...
30 ...
31 ...
32 ...
33 ...
34 ...
35 ...
36 ...
37 ...
38 ...
39 ...
40 ...
41 ...
42 ...
43 ...
44 ...
45 ...
46 ...
47 ...
48 ...
```

Figure 1. Task & Challenge for lesson 5

Lesson 6:



The screenshot shows a Vue.js application interface. At the top, there's a navigation bar with icons for file operations and a 'School' tab. Below it is a header with the text '# index.html ..._lesson6_U'. The main content area displays a product card for a 'sock'. The product image is a pair of green socks with a small logo on the toe. The title 'sock' is displayed in large bold letters. Below the title, the text 'In Stock' is shown. A detailed description follows: '50% cotton', '30% wool', and '20% polyester'. Underneath, color options 'green' and 'blue' are listed. Two buttons are present: 'Add to Cart' and 'Remove from Cart'. Above the 'Add to Cart' button is a small box labeled 'Cart(1)'. To the right of the product card is a code editor window showing the corresponding Vue.js component code.

```
1  ...
2  ...
3  ...
4  ...
5  ...
6  ...
7  ...
8  ...
9  ...
10 ...
11 ...
12 ...
13 ...
14 ...
15 ...
16 ...
17 ...
18 ...
19 ...
20 ...
21 ...
22 ...
23 ...
24 ...
25 ...
26 ...
27 ...
28 ...
29 ...
30 ...
31 ...
32 ...
33 ...
34 ...
35 ...
36 ...
37 ...
38 ...
39 ...
40 ...
41 ...
42 ...
43 ...
44 ...
45 ...
46 ...
47 ...
```

Figure 2. Task & Challenge for lesson 6

Lesson 7:

```

<!-- =====TASK===== -->
<!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"></script>
  </head>
  <body>
    <div id = "app">
      <div class = "nav-bar"></div>
      <div class = "cart">Cart({{cart}})</div>
      <div class = "product-display">
        <div class = "product-container">
          <div class = "product-image">
            <!--CHALLENGE--> 
          </div>
          <div class = "product-info">
            <h1>{{product}}</h1>
            <p v-if = "inStock">In Stock</p>
            <p v-else>Out of Stock</p>
            <ul>
              <li v-for = "detail in details">{{detail}}</li>
            </ul>
            <div v-for="variant in variants"
                :key="variant.id"
                @mouseover="updateImage(variant.image)"
                class="color-circle"
                :style="{'backgroundColor: variant.color'}">
            </div>
          </div>
          <button
            class="button"
            :class="{disabledButton: !inStock}"
            @disabled="!inStock"
            @click="addToCart"
            >Add to Cart
          </button>
        </div>
      </div>
    </div>
  </body>

```

Figure 3. Task & Challenge for lesson 7

Understanding:

In lesson 5, I learned how to display an HTML list from an array of data in the js file. By using v-for, the loop goes through all the values in the array and prints out a list in HTML. It is also possible to assign an id and element for an array; by adding :key with the id in v-for, the loop will print the color element corresponding to that id. This can be practically applied on e-commerce platforms to help sellers quickly update product status, descriptions, and information.

In lesson 6, I learned how to use v-on to create click events and increase the quantity of items in the cart each time the click button is pressed. There are two ways to do this: a simple method that just adds the statement cart += 1 after v-on:click, and a more advanced method requiring a function setup in the js file. Additionally, using the mouseover event, the displayed image can be changed through v-for and an array pre-defined in the js file. The add to cart feature can be applied on all sales websites, as well as other types like movie or story websites (where instead of adding to cart, it can be used to add movies, stories, or music to a playlist).

In lesson 7, there is no need to assign a specific color value directly in the v-for loop; instead, the array defined in js is used. This makes editing element values easier. Also, the disabled attribute can make a button appear disabled when inStock = false. This feature helps clearly show the product's status and prevents users from adding sold-out items to the cart for purchase.