

Appendix1:

Task 1:

The screenshot shows the Visual Studio Code interface with two open files: `index.html` and `main.js`.

index.html:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <title>pm76-week3-task1</title>
    <!-- Import Styles -->
    <link rel="stylesheet" href="assets/styles.css" />
    <script src="https://cdn.jsdelivr.net/npm/vue@2.7.1/dist/vue.js"></script>
  </head>
  <body>
    <div id="app" class ="demo">
      <h1>{{product}}</h1>
    </div>
    <!-- Import Vue.js -->
    <!-- Import Js -->
    <script src="main.js"></script>
  </body>
</html>
```

main.js:

```
//var product = "Socks";
var app = new Vue({
  el:'#app',
  data:{
    product : "Socks",
  }
})
```



Socks

The Code for Task 1

Challenge 1 - Intro to Vue

Vue Mastery PRO + Follow

HTML

```
1 <div id="app">
2   <div class="product-image">
3     <img src="" />
4   </div>
5
6   <div class="product-info">
7     <h1>{{ product }}</h1>
8   </div>
9
10 </div>
11
12 <script
src="https://cdn.jsdelivr.net/npm/vue@2.5.13/dist/vue.js">
```

CSS

```
1
```

JS

```
1 //Add a description to the
2 data object with the value "A
3 pair of warm, fuzzy socks".
4 Then display the description
5 using an expression in an p
6 element, underneath the h1.
7
8 var app = new Vue({
9   el: '#app',
10  data: {
11    product: 'COMP1842'
12  }
13})
```

COMP1842

Challenge for task 1

Task 2:

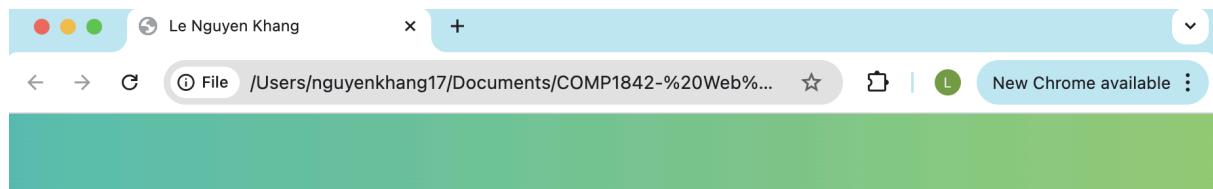
The screenshot shows the VS Code interface with two files open: `index.html` and `main.js`.

`index.html` content:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <title>Le Nguyen Khang</title>
    <!-- Import Styles -->
    <link rel="stylesheet" href="assets/styles.css" />
  </head>
  <body>
    <div class="nav-bar"></div>
    <div id="app">
      <div class="product">
        <div class="product-image">
          
        </div>
        <div class="product-info">
          <h1>{{ product }}</h1>
        </div>
      </div>
    </div>
    <!-- Import Vue.js -->
    <!-- Import Js -->
    <script src="main.js"></script>
  </body>
</html>
```

`main.js` content:

```
//var product = "Socks";
var app = new Vue({
  el: '#app',
  data: {
    product: "Socks",
    image: "./assets/images/socks_green.jpg"
  }
})
```



The code for Task 2

Code editor showing index.html and main.js files.

```
index.html
<html>
  <head>
    <title>Le Nguyen Khang</title>
    <!-- Import Styles -->
    <link rel="stylesheet" href="assets/styles.css" />
    <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="product">
        <div class="product-image">
          
          <!-- Challenge Task 2 -->
          <a v-bind:href="link"> Click the link here </a>
        </div>
      </div>
    </div>
  </body>
</html>
```

```
main.js
var app = new Vue({
  el: '#app',
  data: {
    product: "Socks",
    image: "./assets/images/socks_green.jpg",
    link: "https://youtube.com",
    inventory: 7,
    details: ["80% cotton", "20% polyester", "Gender-neutral"],
    variants: [
      {
        variantId: 2234,
        variantColor: "green",
        variantImage: "./assets/images/socks_green.jpg"
      },
      {
        variantId: 2235,
        variantColor: "blue",
        variantImage: "./assets/images/socks_blue.jpg"
      }
    ]
  }
})
```

Browser preview showing a green sock with a logo and a link.

Socks

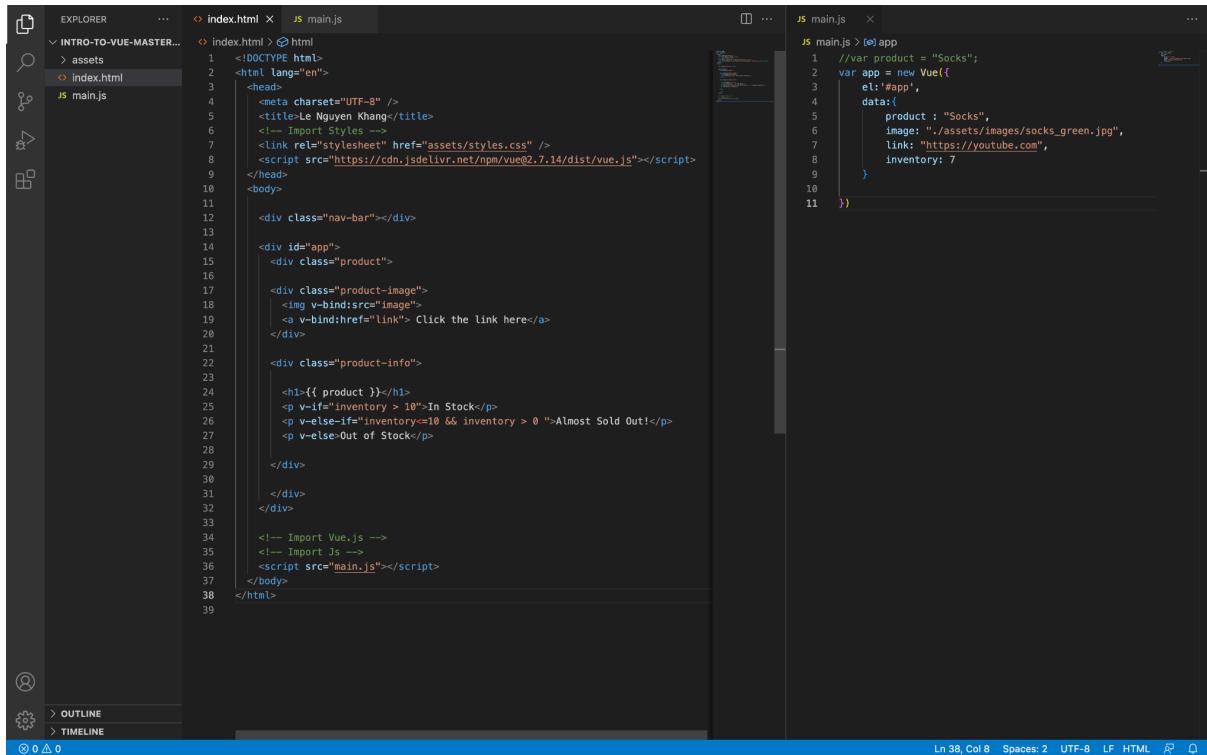


[Click the link here](#)

<https://youtube.com>

Challenge for task 2

Task 3:



The screenshot shows a code editor interface with two files open: `index.html` and `main.js`.

`index.html` content:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <title>Le Nguyen Khang</title>
    <!-- Import Styles -->
    <link rel="stylesheet" href="assets/styles.css" />
  </head>
  <body>
    <div class="nav-bar"></div>
    <div id="app">
      <div class="product">
        <div class="product-image">
          
          <a v-bind:href="link"> Click the link here</a>
        </div>
        <div class="product-info">
          <h1>{{ product }}</h1>
          <p v-if="inventory > 10">In Stock</p>
          <p v-else-if="inventory <= 10 && inventory > 0 ">Almost Sold Out!</p>
          <p v-else>Out of Stock</p>
        </div>
      </div>
    </div>
    <!-- Import Vue.js -->
    <!-- Import Js -->
    <script src="main.js"></script>
  </body>
</html>
```

`main.js` content:

```
//var product = "Socks";
var app = new Vue({
  el:'#app',
  data:{
    product : "Socks",
    image: "/assets/images/socks_green.jpg",
    link: "https://youtube.com",
    inventory: 7
  }
})
```



Socks

Almost Sold Out!

[Click the link here](#)

The code for Task 3

The screenshot shows the VS Code interface with two tabs open: 'index.html' and 'main.js'. The 'index.html' tab displays the HTML structure for a product page, including a navigation bar, a product section with an image and details, and a footer. The 'main.js' tab shows the Vue.js script defining a component with data properties like 'product' (Socks), 'image' (green socks image), and 'variants' (two variants: green and blue). It also includes logic for challenge tasks and sales.

```

<!-- Import Styles -->
<link rel="stylesheet" href="assets/styles.css" />
<script src="https://cdn.jsdelivr.net/npm/vue@2.7.14/dist/vue.js"></script>

<div class="nav-bar"></div>
<div id="app">
  <div class="product">
    <div class="product-image">
      
      <!-- Challenge Task 2 -->
      <a v-bind:href="link" href="https://youtube.com"> Click the link here </a>
    </div>
    <div class="product-info">
      <h1>{{ product }}</h1>
      <!-- Challenge Task 3 -->
      <p v-if="inventory">On Sale!</p>
      <!-- v-if="inventory > 10 > On Stock</p>
    </div>
  </div>
</div>

```

```

var app = new Vue({
  el: '#app',
  data: {
    product: "Socks",
    image: "./assets/images/socks_green.jpg",
    //Challenge Task 2
    link: "https://youtube.com",
    // Challenge Task 3
    inventory: true,
    details: ["80% cotton", "20% polyester", "Gender-neutral"],
    variants: [
      {
        variantId: 2234,
        variantColor: "green",
        variantImage: "./assets/images/socks_green.jpg",
      },
      {
        variantId: 2235,
        variantColor: "blue",
        variantImage: "./assets/images/socks_blue.jpg",
      }
    ],
  }
});

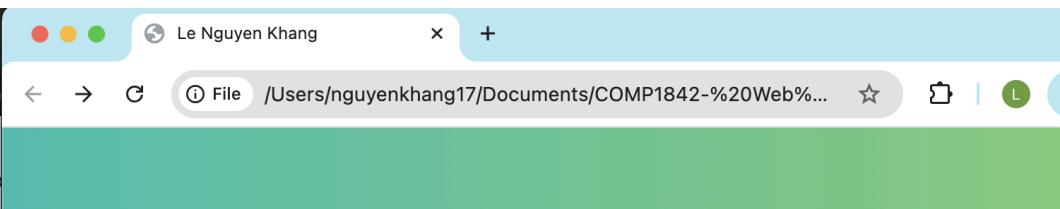
```



[Click the link here](#)

Challenge for Task 3

Task 4:



The screenshot shows a web browser window titled "Le Nguyen Khang". The address bar displays the URL: "/Users/nguyenkhang17/Documents/COMP1842-%20Web%...". The page content is a product listing for "Socks". The image shows two green socks with a white circular logo featuring a mountain range. The text on the page includes "Almost Sold Out", "80% cotton", "20% polyester", and "Gender-neutral". The browser interface includes standard navigation buttons (back, forward, search) and a toolbar.

index.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <title>Le Nguyen Khang</title>
    <!-- Import Styles -->
    <link rel="stylesheet" href="/assets/styles.css" />
    <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="product">
        <div class="product-image">
          
          <!-- <a v-bind:href="#"> Click the link here! </a -->
        </div>
        <div class="product-info">
          <h1>{{ product }}</h1>
          <p v-if="inventory > 10">In Stock</p>
          <p v-else-if="inventory<=10 && inventory > 0">Almost Sold Out</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.variantId">
            <p>{{ variant.variantColor }}</p>
          </div>
        </div>
        <!-- Import Vue.js -->
        <!-- Import JS -->
        <script src="main.js"></script>
      </div>
    </body>
```

main.js

```
var app = new Vue({
  el: '#app',
  data: {
    product: "Socks",
    image: "/assets/images/socks_green.jpg",
    // link: "https://youtube.com",
    inventory: 7,
    details: ["80% cotton", "20% polyester", "Gender-neutral"],
    variants: [
      {
        variantId: 2234,
        variantColor: "green"
      },
      {
        variantId: 2235,
        variantColor: "blue"
      }
    ]
})
```



Socks

Almost Sold Out

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

green

blue

The code for Task 4

```

index.html
<div class="product-image">
  
  <!-- Challenge Task 2 -->
  <a v-bind:href="link" Click the link here>
</div>

<div class="product-info">
  <h1>{{ product }}</h1>
  <!-- Challenge Task 3 -->
  <p v-if="inventory>10">On Sale!</p>
  <p v-if="inventory > 10">In Stock</p>
  <p v-else-if="inventory<=10 && inventory > 0">Almost Sold Out</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.variantId" >
    <p @mouseover="updateProduct(variant.variantImage)">{{ variant }}</p>
  </div>
  <!-- Challenge Task 4 -->
  <ul>
    <li v-for="size in sizes">{{ size }}</li>
  </ul>

```

```

main.js
1 var app = new Vue({
2   el: '#app',
3   data: {
4     product: "Socks",
5     image: "./assets/images/socks_green.jpg",
6     // challenge Task 2
7     // link: "https://youtube.com",
8     // Challenge Task 3
9     // inventory: true,
10    inventory: 7,
11    details: ["80% cotton", "20% polyester", "Gender-neutral"],
12    variants: [
13      {
14        variantId: 2234,
15        variantColor: "green",
16        variantImage: "./assets/images/socks_green.jpg",
17      },
18      {
19        variantId: 2235,
20        variantColor: "blue",
21        variantImage: "./assets/images/socks_blue.jpg",
22      }
23    ],
24    // Challenge Task 4
25    sizes: ["S", "M", "L", "XL"]
26  }
27}
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43

```

Le Nguyen Khang /Users/nguyenkhang17/Documents/COMP1842-%20Web%... New Chrome available



Socks

Almost Sold Out

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

green

blue

- S
- M
- L
- XL

Challenge for Task 4

Understanding for Appendix 1:

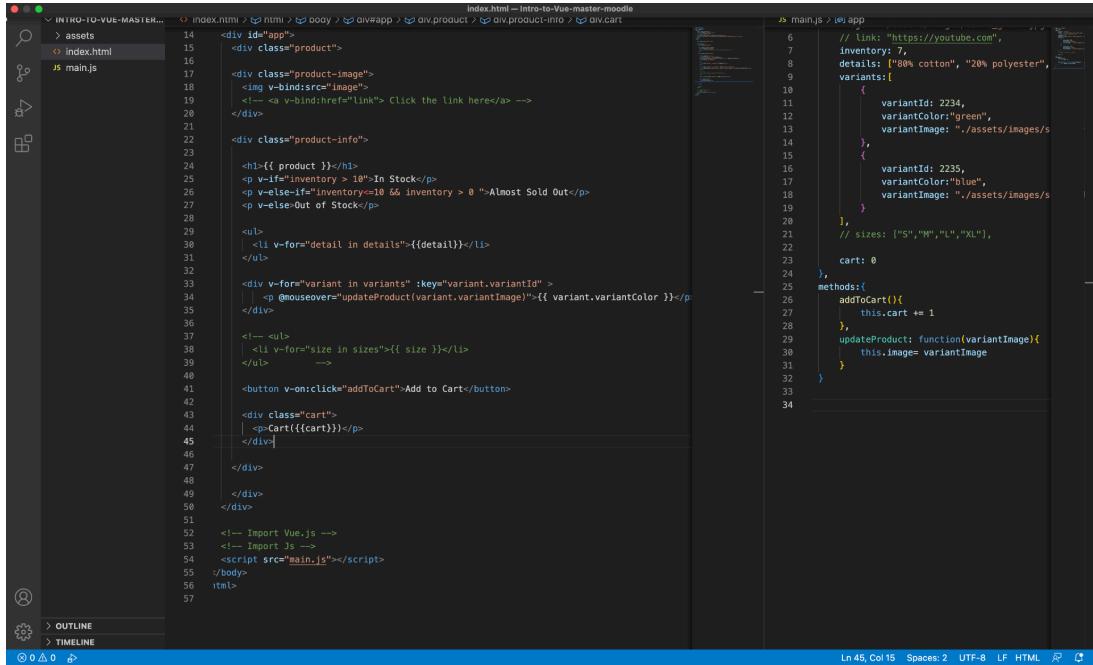
In Vue.js, creating a Vue instance connects data to the DOM, making the interface reactive. This means any changes in the JavaScript data are instantly reflected in the UI without manual updates. Vue's data binding with `{{ }}` allows dynamic data display in HTML, while v-bind enables binding attributes like `src` and `alt` in an image tag to Vue's data. Using shorthand like `:src="image"` simplifies the process, ensuring that when the image source or data changes, the UI automatically updates without reloading the page. This reactivity makes Vue ideal for building interactive, responsive web applications.

For conditional rendering, Vue uses directives like `v-if`, `v-else`, and `v-else-if` to control element display based on conditions. For example, displaying "In Stock" when `inStock` is true or showing an alternative when false. Additionally, `v-show` efficiently toggles element visibility by altering the CSS `display` property without removing elements from the DOM.

Vue's v-for directive makes list rendering straightforward by looping over data arrays and dynamically generating HTML for each item. This reduces the need for hardcoding and makes the application more scalable and maintainable. As data changes, Vue automatically updates the rendered lists, ensuring that the UI stays synchronised with the data model. Overall, Vue simplifies the development of dynamic and interactive interfaces while improving code maintainability and efficiency.

Appendix 2:

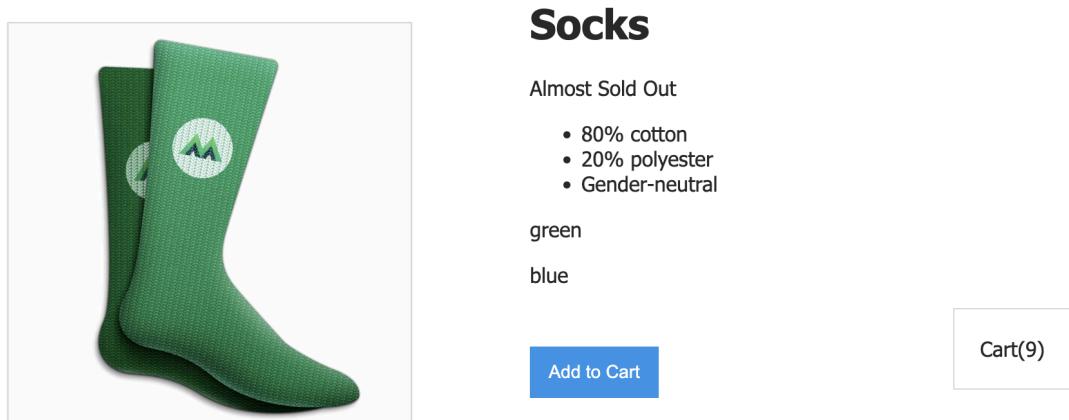
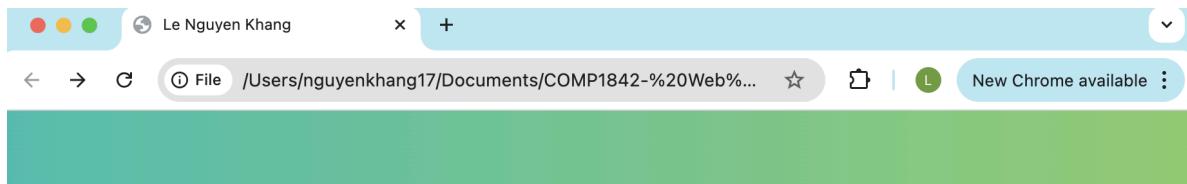
Task 5:



The screenshot shows a code editor with two tabs open: 'index.html' and 'main.js'. The 'index.html' tab displays the structure of a Vue.js application, including components like 'app', 'product', 'product-image', 'product-info', and 'cart'. The 'main.js' tab shows the corresponding JavaScript code, defining a Vue component 'app' with properties for 'variant' and 'cart', and methods for 'addCart()' and 'updateProduct()'. The code uses ES6 syntax and Vue.js directives.

```
index.html - intro-to-Vue-master-moodle
<div id="app">
  <div class="product">
    <div class="product-image">
      
      <a v-bind:href="link" Click the link here></a>
    </div>
    <div class="product-info">
      <h1>{{ product }}</h1>
      <p v-if="inventory > 10">In Stock</p>
      <p v-else-if="inventory<=10 && inventory > 0 ">Almost Sold Out</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.variantId" >
        <p @mouseover="updateProduct(variant.variantImage)">{{ variant.variantColor }}</p>
      </div>
      <!-- -->
      <ul v-for="size in sizes">{{ size }}</li>
      </ul>
      <button v-on:click="addToCart">Add to Cart</button>
    </div>
    <div class="cart">
      <p>Cart({{cart}})</p>
    </div>
  </div>
</div>
<!-- Import Vue.js -->
<!-- Import Js -->
<script src="#main.js"></script>
</body>
</html>
```

```
main.js - [app]
// link: "https://youtube.com",
inventory: 7,
details: ["80% cotton", "20% polyester"],
variants: [
  {
    variantId: 2234,
    variantColor: "green",
    variantImage: "./assets/images/socks-green.jpg"
  },
  {
    variantId: 2235,
    variantColor: "blue",
    variantImage: "./assets/images/socks-blue.jpg"
  }
],
// sizes: ["S", "M", "L", "XL"],
cart: 0
},
methods: {
  addToCart(){
    this.cart += 1
  },
  updateProduct(variantImage){
    this.image= variantImage
  }
}
```



The code for Task 5

```
index.html
<!-- Challenge Task 3 -->
<!-- <p v-if="inventory>10>Sale!</p> -->
<p v-if="inventory > 10">In Stock</p>
<p v-else-if="inventory>=10 && inventory > 8 ">Almost Sold Out</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.variantId" >
|   <p @mouseover="updateProduct({variant.variantImage})">{{ variant.variантId }}</p>
</div>

<!-- Challenge Task 4 -->
<!-- <ul>
|   <li v-for="size in sizes">{{ size }}</li>
</ul> -->

<button v-on:click="addToCart">Add to Cart</button>
<!-- Challenge Task 5 -->
<button v-on:click="decrementCart">Remove from Cart</button>
<div class="cart">
|   <p>Cart({{cart}})</p>
</div>

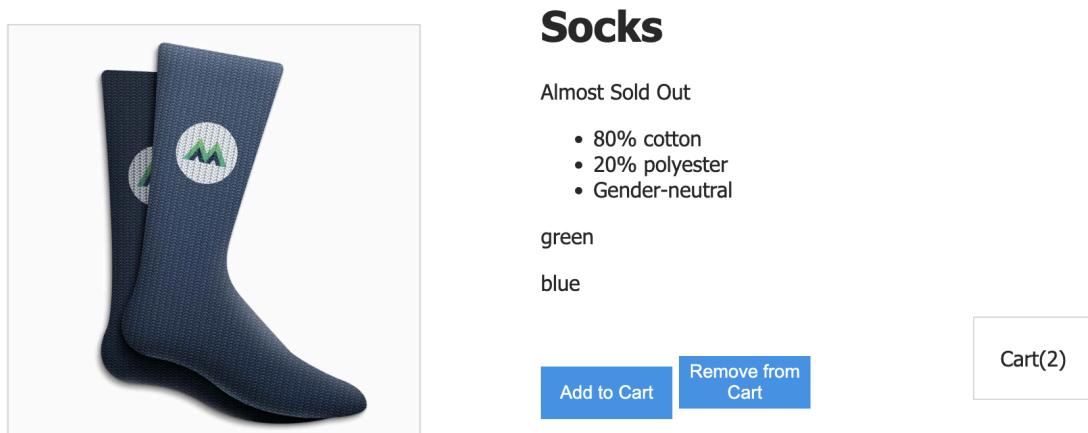
</div>
</div>

<!-- Import Vue.js -->
<!-- Import Js -->
<script src="main.js"></script>
</body>
</html>
```

```
JS main.js
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
49
50
51
52
53
54
55
56
57
58
59
60
61
62

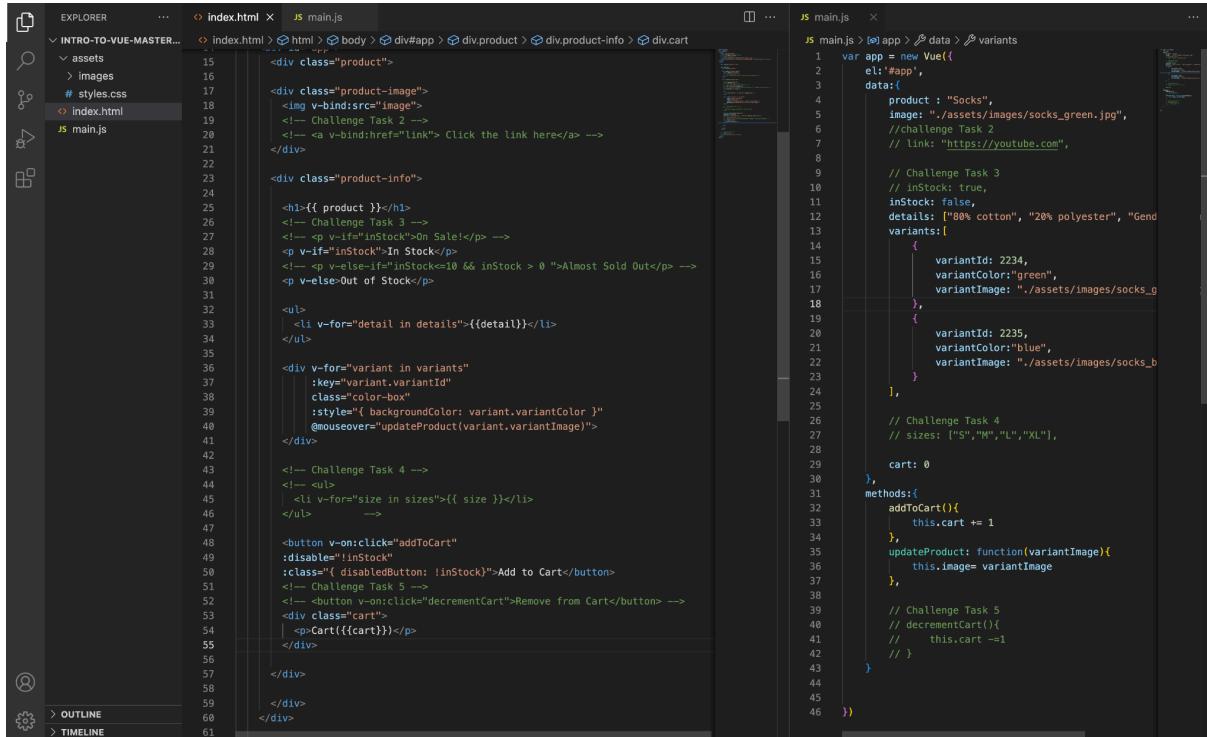
variantId: 2234,
variantColor: "green",
variantImage: "./assets/images/socks_green.jpg",
},
{
variantId: 2235,
variantColor: "blue",
variantImage: "./assets/images/socks_blue.jpg",
}

// Challenge Task 4
// sizes: ["S", "M", "L", "XL"],
// cart: 0
),
methods:{
    addToCart(){
        this.cart += 1
    },
    updateProduct: function(variantImage){
        this.image= variantImage
    },
    // Challenge Task 5
    decrementCart(){
        this.cart -=1
    }
}
```



Challenge for Task 5

Task 6:



The screenshot shows a code editor with two tabs open: 'index.html' and 'main.js'. The 'index.html' tab displays a Vue.js template for a product page. The 'main.js' tab shows the corresponding JavaScript code defining a Vue instance and its data, methods, and computed properties.

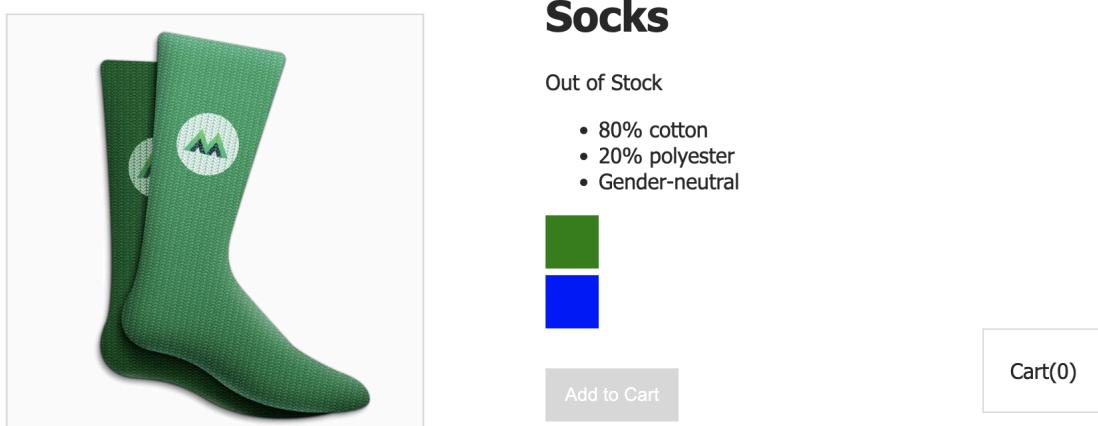
```
index.html
<div class="product">
  <div class="product-image">
    
  </div>
  <!-- Challenge Task 2 -->
  <a v-bind:href="link"> Click the link here</a>
</div>

<div class="product-info">
  <h1>{{ product }}</h1>
  <!-- Challenge Task 3 -->
  <p v-if="inStock">On Sale!</p>
  <p v-if="inStock">In Stock</p>
  <!-- cp v-else-if="inStock==10 && inStock > 0 ">Almost Sold Out</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.variantId</key>
    <div class="color-box" :style="{ backgroundColor: variant.variantColor }">
      @mouseover="updateProduct(variant.variantImage)">
    </div>
  </div>
  <!-- Challenge Task 4 -->
  <!-- ul-->
  <li v-for="size in sizes">{{ size }}</li>
</ul>
  <!-- Challenge Task 5 -->
  <!-- button v-on:click="addCart">Add to Cart</button>
  <!-- Challenge Task 6 -->
  <!-- button v-on:click="decrementCart">Remove from Cart</button>
</div>
</div>
</div>
</div>
```

```
main.js
var app = new Vue({
  el: '#app',
  data: {
    product: "Socks",
    image: "./assets/images/socks_green.jpg",
    //Challenge Task 2
    // Link: "https://youtube.com",
    // Challenge Task 3
    // inStock: true,
    inStock: false,
    details: ["80% cotton", "20% polyester", "Gender-neutral"],
    variants: [
      {
        variantId: 2234,
        variantColor: "green",
        variantImage: "./assets/images/socks_green.jpg"
      },
      {
        variantId: 2235,
        variantColor: "blue",
        variantImage: "./assets/images/socks_blue.jpg"
      }
    ],
    // Challenge Task 4
    // sizes: ["S","M","L","XL"],
    sizes: ["S", "M", "L", "XL"],
    cart: 0
  },
  methods: {
    addCart() {
      this.cart += 1
    },
    updateProduct(variantImage) {
      this.image = variantImage
    },
    // Challenge Task 5
    // decrementCart(){
    //   this.cart -= 1
    // }
  }
})
```



The code for Task 6

EXPLORER ... index.html # styles.css

```

INTRO-TO-VUE-MASTER...
  assets
    > images
      # styles.css
    > index.html
  JS main.js

index.html
1  !DOCTYPE html>
2  <html lang="en">
3  <head>
4    <meta charset="UTF-8" />
5    <title>Le Nguyen Khang</title>
6    <!-- Import Styles -->
7    <link rel="stylesheet" href="#{assets}/styles.css" />
8    <script src="https://cdn.jsdelivr.net/npm/vue@2.7.14/dist/vue.js"></script>
9
10 </head>
11
12 <div class="nav-bar"></div>
13
14 <div id="app">
15   <div class="product">
16     <div class="product-image">
17       
18       <!-- Challenge Task 2 -->
19       <!-- <a v-bind:href="link"> Click the link here</a> -->
20     </div>
21
22     <div class="product-info">
23
24       <h1>{{ product }}</h1>
25       <!-- Challenge Task 3 -->
26       <!-- <p v-if="inStock">On Sale!</p> -->
27       <p v-if="inStock">In Stock</p>
28       <!-- <p v-else-if="inStock<=10 && inStock > 0 ">Almost Sold Out</p> -->
29
30       <!-- Challenge Task 6 -->
31       <p v-else :class="{ outOfStock: !inStock }" >Out of Stock</p>
32
33     <ul>
34       <li v-for="detail in details">{{detail}}</li>
35     </ul>
36
37     <div v-for="variant in variants"
38       :key="variant.variantId"
39       class="color-box"
40       :style="{ backgroundColor: variant.variantColor }"
41       @mouseover="updateProduct(variant.variantImage)">
42
43   </div>

```

styles.css

```

assets > # styles.css > #.activeTab
59   .disabledButton {
60     background-color: #d8d8d8;
61   }
62
63   .review-form {
64     width: 400px;
65     padding: 20px;
66     margin: 40px;
67     border: 1px solid #d8d8d8;
68   }
69
70   input {
71     width: 100%;
72     height: 25px;
73     margin-bottom: 20px;
74   }
75
76   textarea {
77     width: 100%;
78     height: 60px;
79   }
80
81   .tab {
82     margin-left: 20px;
83     cursor: pointer;
84   }
85
86   .activeTab {
87     color: #1E80B0;
88     text-decoration: underline;
89   }
90
91   /* Challenge for Task 6 */
92   .outOfStock {
93     text-decoration: line-through;
94   }

```



Socks

Out of Stock

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




Add to Cart

Cart(0)

Challenge for Task 6

Task 7:

```

EXPLORER ... index.html x JS main.js # styles.css ...
INTRO-TO-VUE-MASTER... index.html > HTML > BODY > DIV#APP > DIV.PRODUCT > DIV.PRODUCT-INFO > DIV.CARD
  | <!-- <a v-bind:href="link"> Click the link here</a> -->
  </div>
<div class="product-info">
  <h1>{{ title }}</h1>
  <!-- Challenge Task 3 -->
  <!-- <p v-if="inStock">On Sale!</p> -->
  <p v-if="inStock">In Stock</p>
  <!-- <p v-else-if="inStock<=10 && inStock > 0 ">Almost Sold Out</p>
  <!-- Challenge Task 6 -->
  <!-- <p v-else :class="( outOfStock: !inStock ) >Out of Stock</p>
  <p v-else>Out of Stock</p>

  <ul>
    <li v-for="detail in details">{{detail}}</li>
  </ul>

  <div v-for="(variant, index) in variants"
    :key="variant.variantId"
    class="color-box"
    :style="{ backgroundColor: variant.variantColor }"
    @mouseover="updateProduct(index)">
    </div>
    <!-- Challenge Task 4 -->
    <!-- <ul>
      <li v-for="size in sizes">{{ size }}</li>
    </ul>
    -->

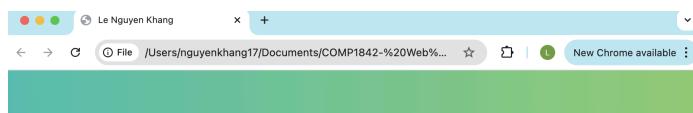
    <button v-on:click="addToCart"
      :disabled="!inStock"
      :class="{ disabledButton: !inStock }">Add to Cart</button>
    <!-- Challenge Task 5 -->
    <!-- <button v-on:click="decrementCart">Remove from Cart</button>
    <div class="cart">
      <p>Cart({{cart}})</p>
    </div>
  </div>
</div>
</div>

```

```

JS main.js > [app]
variants: [
  {
    variantId: 2234,
    variantColor: "green",
    variantImage: "./assets/images/socks_green.jpg",
    variantQuantity: 10
  },
  {
    variantId: 2235,
    variantColor: "blue",
    variantImage: "./assets/images/socks_blue.jpg",
    variantQuantity: 0
  }
],
// Challenge Task 4
// sizes: ["S", "M", "L", "XL"],
cart: 0,
methods: {
  addToCart() {
    this.cart += 1
  },
  updateProduct(index) {
    this.selectedVariant = index
    console.log(index)
  },
},
computed: {
  title() {
    return this.brand + ' ' + this.product
  },
  image() {
    return this.variants[this.selectedVariant].variantImage
  },
  inStock() {
    return this.variants[this.selectedVariant].variantQuantity
  }
}

```



Vue Mastery Socks

In Stock

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



Add to Cart

Cart(0)



Vue Mastery Socks

Out of Stock

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



Add to Cart

Cart(0)

The code for Task 7

```

    var app = new Vue({
      el: '#app',
      data: {
        brand: 'Vue Mastery',
        product: "Socks",
        selectedVariant: 0,
        //Challenge Task 7
        onSale: true,
        //Challenge Task 2-
      },
      details: ["80% cotton", "20% polyester", "Gender-neutral"],
      variants: [
        { ... },
        { ... }
      ],
      // Challenge Task 4-
      cart: 0
    },
    methods: {
      addToCart() {
        ...
      },
      updateProduct: function(index) {
        ...
      }
    },
    // Challenge Task 5
    // decrementCart(){
    //   this.cart -=1
    // }
  },
  computed: {
    // Challenge Task 7
    title() {
      if (this.onSale) {
        return this.brand + ' ' + this.product + ' is on Sale!';
      } else {
        return this.brand + ' ' + this.product;
      }
      // return this.brand + ' ' + this.product
    },
    image() {
      return this.variants[this.selectedVariant].variantImage
    },
    inStock() {
      ...
    }
  }
}

```

The screenshot shows the VS Code interface with the file 'main.js' open. The code defines a Vue.js application with components for a socks product, including details like '80% cotton', '20% polyester', and 'Gender-neutral'. It also includes methods for adding items to a cart and updating the product. A computed property 'title' checks if the product is on sale and returns the appropriate message. An 'image' computed property returns the URL of the selected variant's image. The 'inStock' method is currently commented out.



Vue Mastery Socks is on Sale!

In Stock

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



Add to Cart

Cart(0)

Challenge for Task 7

Understanding for Appendix 2:

Vue's `v-on` directive is a powerful tool for handling DOM events such as clicks or mouseovers, enabling developers to trigger methods that enhance the interactivity of applications. Using the shorthand `@`, event types like `@click` can easily trigger methods such as `addToCart`, which dynamically updates the data model by referencing properties like `this.cart`. This integration keeps the application reactive and ensures that user interactions are instantly reflected in the UI.

In working with class and style binding, I learned how to dynamically link data-driven styles and classes to HTML elements. For instance, using `v-bind:style` allowed me to change a `div`'s background color based on product variants, while class binding grayed out buttons when a product was out of stock. These bindings ensure that the UI adapts automatically to changes in data, offering a seamless user experience without manual DOM manipulation.

Additionally, Vue's computed properties calculate values dynamically, such as combining product brand and name into a single string, streamlining the logic and reducing redundancy in the data model. By leveraging Vue's reactivity and directives like `v-for` for list rendering, I saw how Vue simplifies complex UI updates, providing cleaner, more maintainable code for scalable applications.