



COMP1842 Lab Task

Week 3 - 26/09/2024

COS1104 Nguyen Dinh Hoang Son GCS220616

Introduction

Task 1

Lesson 1: The Vue Instance

Overview

In this lesson, I learned the fundamental concepts of Vue.js by building a simple product page. The key takeaway is how Vue allows us to display data dynamically on a webpage and how it provides reactivity to ensure the page updates when the data changes.

Challenge Solution

javascript

```
var app = new Vue({  
  el: '#app',  
  data: {  
    product: 'Socks',  
    image: './assets/images/socks.jpg',  
    altText: 'A pair of socks',
```

```
    link: 'https://en.wikipedia.org/wiki/Sock'  
  }  
});
```

Code Line 9. A `link` property is added to the Vue instance's `data`, containing a URL.

html

```
<a v-bind:href="link">More details</a>
```

Code Line 10. Using `v-bind:href`, the `href` attribute of the anchor (`a`) tag is bound to the `link` data property. This makes the anchor's URL dynamic based on the data.

Conclusion

In this lesson, I learned how to:

- Use `v-bind` to dynamically bind data to HTML attributes like `src`, `alt`, and `href`.
- Apply the shorthand (`:`) for cleaner code.
- Make attributes responsive to data changes within the Vue instance.

By completing the challenge, I bound a dynamic link to an anchor tag, further demonstrating my understanding of attribute binding in Vue.js.

Lesson 3: Conditional Rendering

Overview

In this lesson, I learned how to conditionally display elements in Vue using directives like `v-if`, `v-else`, and `v-show`. This allows for dynamic changes in the UI based on the state of the application.

Challenge Solution

To complete the challenge, I added an `onSale` property to conditionally render a message when the product is on sale.

javascript

```
data: {  
  product: 'Socks',  
  inStock: true,  
  onSale: true  
}
```

Code Line 16. Adding the `onSale` property to the data object.

html

```
<span v-if="onSale">On Sale!</span>
```

Code Line 17. Using `v-if` to conditionally render a span that displays "On Sale!" if the `onSale` property is `true`.

Lesson 4: List Rendering

Overview

In this lesson, I learned how to use Vue's `v-for` directive to display lists of data on a webpage. This allows dynamic rendering of lists by looping through arrays or objects and displaying each element as HTML.

Challenge Solution

To meet the challenge, we can add an array of sizes and use `v-for` to render a list of available sizes.

javascript

```
var app = new Vue({  
  el: '#app',  
  data: {  
    product: 'Socks',  
    details: ['80% cotton', '20% polyester', 'Gender-neutral'],  
    variants: [  
      { variantId: 2234, color: 'green' },  
      { variantId: 2235, color: 'blue' }  
    ],  
    sizes: ['S', 'M', 'L']  
  }  
})
```

```
}  
});
```

Code Line 20: The array `sizes` is added to the data object, which contains the available sizes of the product.

```
html  
<ul>  
  <li v-for="size in sizes">{{ size }}</li>  
</ul>
```

Code Line 21: This loop iterates over the `sizes` array and renders each size in a list item (`li`).

Conclusion

In this lesson, I learned how to use the `v-for` directive to render arrays and object arrays in Vue.js. This directive allows for looping through data, displaying each item dynamically on the page, and using dot notation to access specific properties in objects. I also learned the importance of using a unique key for each element when rendering lists.

Task 2

Lesson 5: Event Handling

Overview

In this lesson, I learned how to handle DOM events in Vue.js using the `v-on` directive. This directive allows us to listen for various user interactions and trigger corresponding methods, enhancing the interactivity of our web application.

Challenge Solution

To solve the challenge, I created a new button to decrement the cart value, along with a method to handle the event.

```
html
```

```
<button @click="removeFromCart">Remove from Cart</button>
```

Code Line 30: A button is added to trigger the `removeFromCart` method when clicked.

javascript

```
methods: {  
  removeFromCart() {  
    if (this.cart > 0) {  
      this.cart -= 1;  
    }  
  }  
}
```

Code Line 31: The `removeFromCart` method decreases the cart value by 1, ensuring that it doesn't go below 0.

Conclusion

In this lesson, I learned how to handle events in Vue.js using the `v-on` directive and its shorthand `@`. Events like `click` and `mouseover` can trigger methods, which can take arguments to perform more complex tasks. Additionally, I explored how `this` refers to the current Vue instance's data and how it can be used to dynamically update content.

Lesson 6: Class & Style Binding

Overview

In this lesson, I learned how to dynamically style HTML elements by binding Vue.js data to their class and style attributes. This allows for more interactive and responsive UI elements that reflect changes in the underlying data.

Challenge Solution: Binding a Class to the "Out of Stock" Text

To meet the challenge, we can dynamically bind a class to the "Out of Stock" text that adds a `line-through` style when the product is out of stock.

html

```
<p :class="{ outOfStockText: !inStock }">Out of Stock</p>
```

Code Line 36: The `outOfStockText` class is applied to the `<p>` tag when the product is out of stock, adding a `line-through` effect.

CSS

```
.outOfStockText {  
  text-decoration: line-through;  
}
```

Code Line 37: The `outOfStockText` class uses CSS to apply the `line-through` effect.

Conclusion

In this lesson, I learned how to use Vue.js for both style and class binding. This allows us to dynamically update the style and appearance of elements based on reactive data. We explored binding the `background-color` of elements using `v-bind:style` and conditionally applying CSS classes using `v-bind:class`. Additionally, I learned how to manage disabled states for buttons using both attribute and class bindings.

Lesson 7: Computed Properties

Overview

In this lesson, I learned how to use **computed properties** in Vue.js. These properties calculate values based on existing data and update automatically when the underlying data changes. This makes them ideal for combining or processing reactive data without duplicating logic.

Challenge Solution: `onSale` Computed Property

To meet the challenge, we add a boolean data property `onSale` and create a computed property that shows a message when the product is on sale.

javascript

```
computed: {  
  title() {  
    return this.brand + ' ' + this.product
```

```

    },
    image(){
        return this.variants[this.selectedVariant].variantImage
    },
    inStock(){
        return
this.variants[this.selectedVariant].variantQuantity
    },
    sale_message() {
        if (this.onSale) {
            return this.brand + ' ' + this.product + ' are on sale!'
        }
        return this.brand + ' ' + this.product + ' are not on
sale'
    }
}

```

Code Line 41: The computed property `sale` checks if `onSale` is true. If the product is on sale, it appends "is on Sale!".

html

```

<h1>{{ sale_message }}</h1>

```

Code Line 42: The `<h1>` tag now displays the `sale` computed property, which dynamically reflects whether the product is on sale.

Conclusion

In this lesson, I learned that computed properties in Vue.js calculate values based on reactive data and are automatically updated when their dependencies change. They are cached for efficiency and should be used for pure functions that don't mutate data. By leveraging computed properties, we can make the application cleaner and more scalable.

Appendices

Lesson 1: Introduction to Vue.js

```
Challenge 1 - Intro to Vue
Vue Mastery 100% 100%

HTML CSS JS

1 <div id="app"> <!-- vue instance will control everything within this div -->
2
3 <div class="product-image">
4   <img src="" /> <!-- Placeholder for dynamically binding product image -->
5 </div>
6
7 <div class="product-info">
8   <h1{{ product }}> <!-- We will dynamically insert the product name here -->
9   <h2{{ description }}> <!-- We will dynamically insert the product description here -->
10 </div>
11
12 </div>
13
14 </div>
15
16 <script src="https://cdn.jsdelivr.net/npm/vue@2.6.11/dist/vue.js"></script> <!-- Importing -->
17
18 </script>
```

Socks

A pair of warm, fuzzy socks.

Lesson 2: Event Handling

```
Challenge 2 - Intro to Vue
Vue Mastery 100% 100%

HTML CSS JS

1 <div id="app"> <!-- vue instance will control everything within this div -->
2
3 <div class="product">
4   <div class="product-image">
5     <img src="" /> <!-- Dynamically bind the image source using vue's v-bind syntax -->
6   </div>
7
8   <div class="product-info">
9     <h1{{ product }}> <!-- We will dynamically insert the product name here -->
10     <a href="" link="" target="">More details</a> <!-- Dynamically bind the href attribute to the link data -->
11   </div>
12
13 </div>
14
15 </div>
16
17 </script>
```



Socks

[More details](#)

Lesson 3: Conditional Rendering

Challenge 3 - Intro to Vue

```

HTML
<div class="nav-bar"></div>
<div id="app"> <!-- Vue instance will control everything within this div -->
<div class="product">
  <div class="product-image">
    <img src="" /> <!-- Dynamically bind the image source using vue's v-bind syntax -->
  </div>
  <div class="product-info">
    <div {{ product }}> <!-- We will dynamically insert the product name here -->
    <p v-if="inStock">In Stock</p> <!-- Conditionally render "In Stock" if inStock is true -->
  </div>
</div>
</div>
CSS
body {
  font-family: tahoma;
  color: #282828;
  margin: 0px;
}
.nav-bar {
  background: linear-gradient(to top, #000000, #000000);
  height: 50px;
  margin-bottom: 10px;
}
.product {
  display: flex;
}
JS
var app = new Vue({
  el: 'app', // Link this Vue instance to the HTML element with id "app"
  data: {
    products: 'Socks', // Data property for the product name
    image: 'https://www.vuumastery.com/images/challenges/vsocks-green-onwhite.jpg', // Data property for the product image url
    inStock: true, // Data property to indicate if the product is in stock
    onSale: true // Data property to indicate if the product is on sale
  }
})

```



Socks

In Stock
On Sale!

Lesson 4: Lists and Rendering with v-for

Challenge 4 - Intro to Vue

```

HTML
<div class="nav-bar"></div>
<div id="app"> <!-- the root element for the vue instance -->
<div class="product">
  <div class="product-image">
    <img src="" /> <!-- Dynamically bind the image source using vue's v-bind syntax -->
  </div>
  <div class="product-info">
    <div {{ product }}> <!-- We will dynamically insert the product name here -->
    <p v-if="inStock">In Stock</p> <!-- Conditionally render "In Stock" if inStock is true -->
  </div>
</div>
CSS
body {
  width: 100%;
  padding: 10px;
  border: 1px solid #000;
}
input {
  width: 100%;
  height: 30px;
  margin-bottom: 10px;
}
textarea {
  width: 100%;
  height: 80px;
}
JS
var app = new Vue({
  el: 'app', // Link this Vue instance to the HTML element with id "app"
  data: {
    products: 'Socks', // Data property for the product name
    image: 'https://www.vuumastery.com/images/challenges/vsocks-green-onwhite.jpg', // Data property for the product image url
    inStock: true, // Data property to indicate if the product is in stock
    details: ['80% cotton', '20% polyester', 'Gender-neutral'], // Array of product details
    variants: [ // Array of product variants
      {
        variantId: 001,
        variantColor: 'green'
      },
      {
        variantId: 002,
        variantColor: 'blue'
      }
    ]
  }
})

```



Socks

In Stock

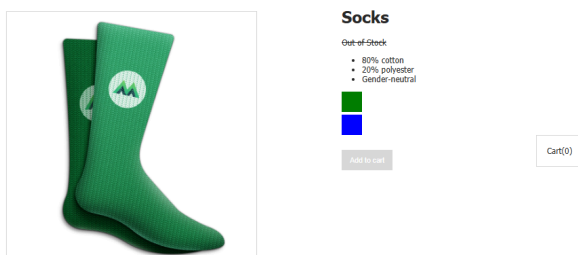
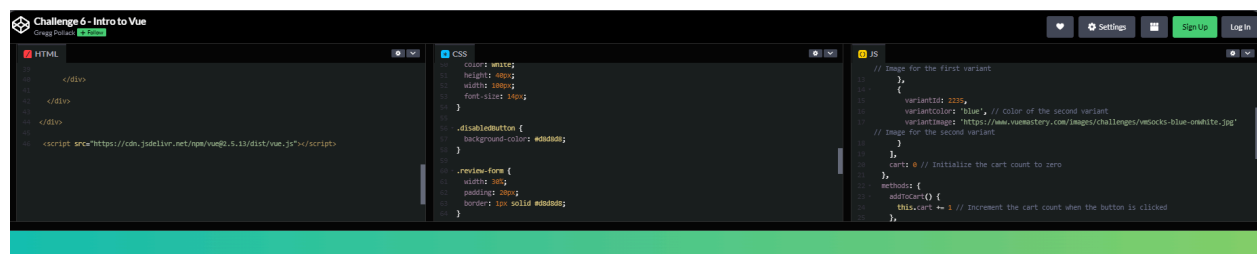
- 80% cotton
- 20% polyester
- Gender-neutral

green

blue

- S
- M
- L

Lesson 5: Computed Properties



Lesson 7: Computed Properties

Challenge 7 - Intro to Vue

HTML

JS

```
1 <div v-for="detail in details" :key="detail" />
2 </div>
3
4 <div class="color-box">
5   <div v-for="(variant, index) in variants"
6     :key="variant.variantId"
7     :style="{ backgroundColor: variant.variantColor }"
8     @mouseover="updateProduct(index)"
9   >
10
11 </div>
12
13 <button v-on:click="addToCart"
14   :disabled="!inStock"
15   :track="true" :disablebutton="true" />
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
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```



Vue Mastery Socks

In Stock

Vue Mastery Socks are on sale!

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



Add to cart

Cart(0)