

Day 4 - Dynamic Frontend Components

- [www.WoodenAspire.com]

Objective:

On Day 4, we will concentrate on designing and creating dynamic frontend components to showcase marketplace data retrieved from Sanity CMS or APIs. This step highlights the importance of modular and reusable component design, as well as real-world techniques for developing scalable and responsive web applications.

Key Learning Outcomes:

- 1. Build dynamic frontend components to display data from Sanity CMS or APIs.**
- 2. Implement reusable and modular components.**
- 3. Understand and apply state management techniques.**
- 4. Learn the importance of responsive design and UX/UI best practices.**
- 5. Prepare for real-world client projects by replicating professional workflow**

Overview of System Functions:

The project integrates the following key functionalities:

- 1. Product Listing Page.**
- 2. Individual Product Detail Page.**
- 3. Cart Page.**

4. Search Bar.
5. Checkout.
6. Header.
- 7.Footer.

Each feature is integral to constructing a responsive and scalable marketplace platform.






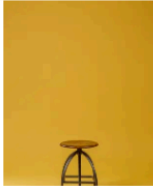
Key Components to Build:

Product Listing Component.

Code:

```
ProductGrid.tsx M X
src > app > components > ProductGrid.tsx > Product
1  "use client"
2  import React, { useEffect, useState } from "react"; 6.9k (gzipped: 2.7k)
3  import Image from "next/image"; 16.5k (gzipped: 6.1k)
4  import ProductCard from "../components/Cards";
5  import { fetchProducts } from "@utils/fetch";
6
7  const Product = () => {
8    const [products, setProducts] = useState([]);
9    const [loading, setLoading] = useState(true); // End loading
10   useEffect(() => {
11     const getProducts = async () => {
12       setLoading(true); // Start loading when fetching starts
13       try {
14         const data = await fetchProducts(); // Call the utility function
15         setProducts(data); // Set products in the state
16       } catch (error) {
17         console.error("Error fetching products:", error);
18       } finally {
19         setLoading(false);
20       }
21     };
22   });
23   getProducts();
24   return (
25     <div className="px-4 py-8 md:px-12">
26       <h1 className="text-2xl md:text-3xl font-bold text-center mb-6">Our Products</h1>
27
28       <div className="grid grid-cols-3 justify-center"> { /* Use flex-wrap and justify-center for wrapping and centering the cards */ }
29
30       {products.map((product, index) => (
31         <div key={index}>
32           <ProductCard product={product} />
33         </div>
34       ))}
35     </div>
36
37     <div className="flex justify-center mt-8">
38       <button className="px-6 py-2 bg-[#B88E2F] text-white rounded-md">
39         Show More
40       </button>
41     </div>
42   </div>
43 );
44
45 export default Product;
```

Output Comes:

Our Products		
<div></div> <div>Rustic Vase Set rustic vase home decorvintage interior design \$210</div>	<div></div> <div>Amber Haven amber luxury cozy elegant furniture \$150</div>	<div></div> <div>Bold Nest bold nest furniture modern contemporary \$260</div>
<div></div> <div>Cloud Haven Chair cloud chair comfy home decormodern furniture \$230</div>	<div></div> <div>Bright Space bright space minimalistic modern decor \$180</div>	<div></div> <div>The Dandy chair chair elegant vintage classic furniture \$150</div>

Individual Product Detail Component.

Code:

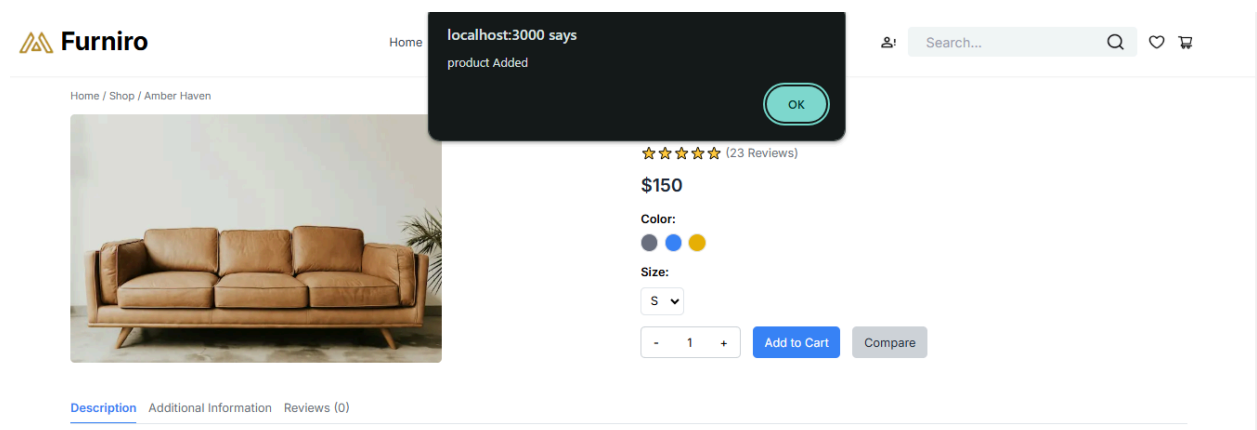
Cart Component.

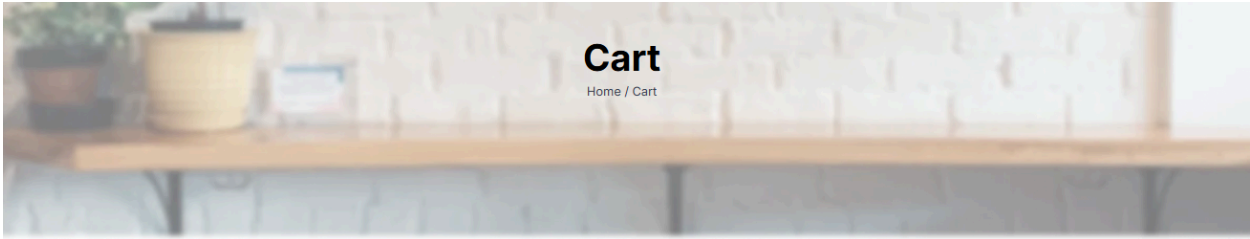
Code:

```
src > app > cart > page.tsx > Cartpage > cart.map() callback
1  "use client"
2  import React, { useEffect, useState } from 'react'; 6.9k (gzipped: 2.7k)
3  import Link from 'next/link'; 32.5k (gzipped: 9.6k)
4  import delete_icon from "@app/images/delete_icon.png"
5  import Header from "../components/Header"
6  import CartHero from "../components/CartHero"
7  import Achievement from "../components/Achievement"
8  import Footer from "../components/Footer"
9  import { useCart } from "@context/cartContext";
10
11 import Image from 'next/image'; 16.5k (gzipped: 6.1k)
12 import { useSearchParams } from 'next/navigation'; 14.2k (gzipped: 3k)
13 import { fetchProducts } from '@utils/fetch';
14 import { ProductType } from '@types/product.types';
15 import { product } from '@sanity/schemaTypes/product';
16
17 const Cartpage = () => {
18   const { cart, removeFromCart } = useCart();
19   const [products, setProducts] = useState<ProductType[]>(cart);
20
21   useEffect(() => {
22     setProducts(cart);
23   }, [cart]);
24
25   return (
26     <div className="bg-white">
27       <Header />
28       <CartHero />
29       <div className="container mx-auto px-4 sm:px-6 lg:px-8">
30         <div className="grid grid-cols-3 gap-8">
31           <div className="col-span-2">
32             <div className="bg-[#F9F1E7] grid grid-cols-4 text-center p-4">
33               <h6 className="font-bold text-black">Product</h6>
34               <h6 className="font-bold text-black">Price</h6>
35               <h6 className="font-bold text-black">Quantity</h6>
36               <h6 className="font-bold text-black">Subtotal</h6>

```




Output Comes:





Cart

Home / Cart

Product	Price	Quantity	Subtotal
Amber Haven	\$150	<input type="text" value="1"/>	\$150 
Bold Nest	\$260	<input type="text" value="1"/>	\$260 
Rustic Vase Set	\$210	<input type="text" value="1"/>	\$210 

Cart Totals	
Subtotal	620.00
Total	620.00
<input type="button" value="Check Out"/>	



High Quality
crafted from top materials



Waranty Protection
Over 2 years



Free Shipping
Order over 150 \$



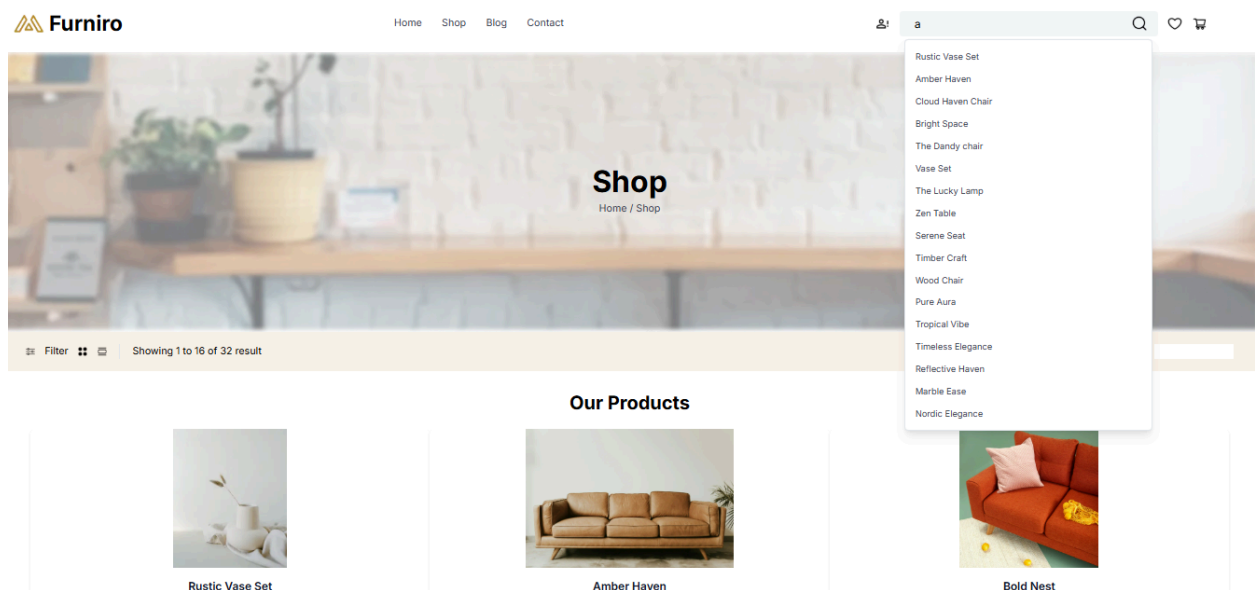
24 / 7 Support
Dedicated Support

Search Bar Component.

Code:

```
SearchBar.jsx
src > app > components > SearchBar > SearchBar
1  "use client";
2  import React, { useEffect, useState } from "react"; 6.9K (gzipped: 2.7k)
3  import Image from "next/image"; 16.5K (gzipped: 6.1k)
4  import SearchIcon from "../images/search-icon.png";
5  import Link from "next/link"; 32.5K (gzipped: 9.6k)
6  import { fetchProducts } from "@utils/fetch";
7  import { ProductType } from "@types/product.types";
8
9  const SearchBar = () => {
10   const [query, setQuery] = useState("");
11   const [products, setProducts] = useState<ProductType[]>([]);
12   const [loading, setLoading] = useState(true);
13
14
15   useEffect(() => {
16     const getProducts = async () => {
17       setLoading(true); // Start loading when fetching starts
18       try {
19         const data = await fetchProducts(); // Call the utility function
20         setProducts(data); // Set products in the state
21       } catch (error) {
22         console.error("Error fetching products:", error);
23       } finally {
24         setLoading(false); // End loading
25       }
26     };
27
28     getProducts();
29   }, []);
30
31   const handleInputChange = (e: { target: { value: React.SetStateAction<string>; }; }) => {
32     setQuery(e.target.value);
33   };
34
35   return (
36     <div className="bg-slate-100 flex flex-col w-full max-w-xs sm:max-w-sm md:max-w-md lg:max-w-lg rounded-md px-2 relative">
37       <input type="text" value={query} onChange={handleInputChange} />
38     </div>
39   );
40 }
```

Output Comes:



CheckOut Component.

Code:

```
src > app > checkout > @ pages > @ BillingPage > @ handleBuyAndPlaceOrder
1 "use client"
2 import React, { useEffect } from "react"
3 import Header from "../../components/Header"
4 import CheckoutHero from "../../components/CheckoutHero"
5 import Achievement from "../../components/Achievement"
6 import Footer from "../../components/Footer"
7
8 import { ProductType } from "../../types/product.types"
9 import { useCart } from "@context/cartContext"
10
11 const BillingPage = () => {
12   const { cart, removeFromCart } = useCart()
13   const [products, setProducts] = useState<ProductType[]>(cart)
14   const [isProcessing, setIsProcessing] = useState(false)
15   const [paymentSuccess, setPaymentSuccess] = useState(false)
16
17   useEffect(() => {
18     setProducts(cart)
19   }, [cart])
20
21   const handleBuyAndPlaceOrder = () => {
22
23     setIsProcessing(true)
24     setTimeout(() => {
25       setIsProcessing(false)
26       setPaymentSuccess(true)
27       setTimeout(() => {
28         setPaymentSuccess(false)
29       }, 3000)
30     }, 3000)
31   }, 2000)
32
33   return (
34     <div>
35       <Header />
36       <CheckoutHero />
37       <div className="min-h-screen bg-gray-50 p-4 sm:p-6 lg:p-10">
38         <div className="max-w-7xl mx-auto bg-white shadow-md rounded-md p-6 md:p-10 grid grid-cols-1 lg:grid-cols-3 gap-6">
39           <div className="lg:col-span-2">
40             <div className="text-2xl font-sans font-weight-bold mb-6">Billing details</div>
41             <form className="grid grid-cols-1 sm:grid-cols-2 gap-4">
42               <div>
43                 <label className="block text-sm font-medium text-gray-700">First Name</label>
44                 <input
45                   type="text"
46                   className="w-full border-gray-300 rounded-md shadow-sm focus:ring-indigo-500 focus:border-indigo-500 sm:text-sm"
47                 />
48             </form>
49           </div>
50         </div>
51       </div>
52     </div>
53   )
54 }
```

Output Comes:

Output Comes:



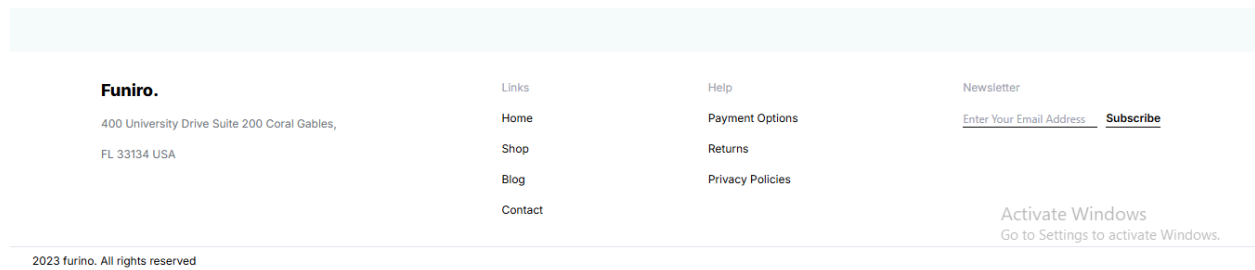
Footer Component.

Code:

```
page.jsx U  Footer.jsx X
src > app > components > Footer.jsx ...
1  import Link from "next/link"; 32.5k (gzipped: 9.6k)
2  import React from "react" 6.9k (gzipped: 2.7k)
3
4  const Footer = () => {
5    return (
6      <footer>
7        <div className="px-5 py-10 flex flex-col md:flex-row justify-around gap-8 md:gap-0">
8          <div className="space-y-5 text-center md:text-left">
9            <h1 className="text-2xl font-extrabold">Furniro.</h1>
10           <p className="text-gray-500">400 University Drive Suite 200 Coral Gables,</p>
11           <p className="text-gray-500">FL 33134 USA</p>
12         </div>
13
14         <div className="space-y-5 text-center md:text-left">
15           <p className="text-gray-400">Links</p>
16           <p>Home</p>
17           <p>Shop</p>
18           <p>Blog</p>
19           <p>Contact</p>
20         </div>
21
22         <div className="space-y-5 text-center md:text-left">
23           <p className="text-gray-400">Help</p>
24           <p>Payment Options</p>
25           <p>Returns</p>
26           <p>Privacy Policies</p>
27         </div>
28
29         <div className="space-y-5 text-center md:text-left">
30           <p className="text-gray-400">Newsletter</p>
31           <div className="flex flex-col md:flex-row gap-2 space-x-3 md:gap-0">
32             <input
33               type="email"
34               className="font-sans bg-transparent border-b-2 border-black text-center md:text-left"
35               placeholder="Enter Your Email Address"/>
36             <Link href="#" className="font-semibold md:border-b-2 border-black">Subscribe</Link>
37           </div>
38         </div>
39       </div>
40     );
  }
```

Activate Windows
Go to Settings to activate Windows.

Output Comes:



API Integration.

Code:

```
const [products, setProducts] = useState([]);
const [loading, setLoading] = useState(true); // End loading
useEffect(() => {
  const getProducts = async () => {
    setLoading(true); // Start loading when fetching starts
    try {
      const data = await fetchProducts(); // Call the utility function
      setProducts(data); // Set products in the state
    } catch (error) {
      console.error("Error fetching products:", error);
    }
  }
  getProducts();
}, []);
```

```
    } finally {  
      setLoading(false);  
    }  
  };  
  
  getProducts();  
}, []);
```

Fetch.ts.

Code:

```
import { client } from '../sanity/lib/client'  
  
export async function fetchProducts() {  
  const query = `  
    *[_type == "product"]{  
      _id,  
      _type,  
      title,  
      price,  
      "productImage": productImage.asset->url,  
      tags,  
      discountPercentage,  
      description,  
      isNew  
    }  
  `;  
  
  const products = await client.fetch(query)  
  return products  
}
```

Dynamic Routing.

Dynamic routing is used to make web applications more flexible, efficient, and user-friendly. It allows the application to handle different pages or components dynamically without creating a separate route for each one.

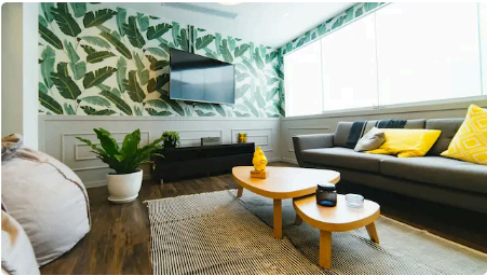
Next.js Dynamic Routes File:

- 1) Routes like `/product/:id` allow the app to handle many pages using a single route, where `:id` changes based on the product.
- 2) Create a file named `[id]` under the `shop` folder directory.
- 3) When you visit a URL like `page/product/[id]`, the `id` parameter is available in the file for rendering content dynamically.

Code:

```
src > app > shop > [id] > page.tsx M X
1 "use client"
2 import React, { useEffect, useState } from "react";
3 import Image from "next/image";
4 import Link from "next/link";
5 import Header from "../../components/Header";
6 import Achievement from "../../components/Achievement";
7 import Footer from "../../components/Footer";
8
9 import { useParams } from "next/navigation";
10 import ProductGrid from "@app/components/ProductGrid";
11 import { fetchProducts } from "@utils/fetch";
12 import { CartItem, ProductType } from "@types/product.types";
13 import { useCart } from "@context/cartContext";
14
15 const ProductPage = () => {
16   const { addToCart, cart, updateQuantity } = useCart();
17   const [products, setProducts] = useState<ProductType[]>([]);
18   const [loading, setLoading] = useState(true);
19
20   useEffect(() => {
21     const getProducts = async () => {
22       setloading(true);
23       try {
24         const data = await fetchProducts();
25         setProducts(data);
26       } catch (error) {
27         console.error("Error fetching products:", error);
28       } finally {
29         setLoading(false);
30       }
31     };
32     getProducts();
33   }, []);
34
35   const { id } = useParams(); // id 2
36
37   127
38   128
39   129
40   130
41   131
42   132
43   133
44   134
45   135
46   136
47   137
48   138
49   139
50   140
51   141
52   142
53   143
54   144
55   145
56   146
57   147
58   148
59   149
60   150
61   151
62   152
63   153
64   154
65   155
66   156
67   157
68   158
69   159
70   160
71   161
72   162
73   163
74   164
75   165
76   166
77   167
78   168
79   169
80   170
81   171
82   172
83   173
84   174
85   175
86   176
87   177
88   178
89   179
90   180
91   181
92   182
93   183
94   184
95   185
96   186
97   187
98   188
99   189
100  190
101  191
102  192
103  193
104  194
105  195
106  196
107  197
108  198
109  199
110  200
111  201
112  202
113  203
114  204
115  205
116  206
117  207
118  208
119  209
120  210
121  211
122  212
123  213
124  214
125  215
126  216
127  217
128  218
129  219
130  220
131  221
132  222
133  223
134  224
135  225
136  226
137  227
138  228
139  229
140  230
141  231
142  232
143  233
144  234
145  235
146  236
147  237
148  238
149  239
150  240
151  241
152  242
153  243
154  244
155  245
156  246
157  247
158  248
159  249
160  250
161  251
162  252
163  253
164  254
165  255
166  256
167  257
168  258
169  259
170  260
171  261
172  262
173  263
174  264
175  265
176  266
177  267
178  268
179  269
180  270
181  271
182  272
183  273
184  274
185  275
186  276
187  277
188  278
189  279
190  280
191  281
192  282
193  283
194  284
195  285
196  286
197  287
198  288
199  289
200  290
201  291
202  292
203  293
204  294
205  295
206  296
207  297
208  298
209  299
210  300
211  301
212  302
213  303
214  304
215  305
216  306
217  307
218  308
219  309
220  310
221  311
222  312
223  313
224  314
225  315
226  316
227  317
228  318
229  319
230  320
231  321
232  322
233  323
234  324
235  325
236  326
237  327
238  328
239  329
240  330
241  331
242  332
243  333
244  334
245  335
246  336
247  337
248  338
249  339
250  340
251  341
252  342
253  343
254  344
255  345
256  346
257  347
258  348
259  349
260  350
261  351
262  352
263  353
264  354
265  355
266  356
267  357
268  358
269  359
270  360
271  361
272  362
273  363
274  364
275  365
276  366
277  367
278  368
279  369
280  370
281  371
282  372
283  373
284  374
285  375
286  376
287  377
288  378
289  379
290  380
291  381
292  382
293  383
294  384
295  385
296  386
297  387
298  388
299  389
300  390
301  391
302  392
303  393
304  394
305  395
306  396
307  397
308  398
309  399
310  400
311  401
312  402
313  403
314  404
315  405
316  406
317  407
318  408
319  409
320  410
321  411
322  412
323  413
324  414
325  415
326  416
327  417
328  418
329  419
330  420
331  421
332  422
333  423
334  424
335  425
336  426
337  427
338  428
339  429
340  430
341  431
342  432
343  433
344  434
345  435
346  436
347  437
348  438
349  439
350  440
351  441
352  442
353  443
354  444
355  445
356  446
357  447
358  448
359  449
360  450
361  451
362  452
363  453
364  454
365  455
366  456
367  457
368  458
369  459
370  460
371  461
372  462
373  463
374  464
375  465
376  466
377  467
378  468
379  469
380  470
381  471
382  472
383  473
384  474
385  475
386  476
387  477
388  478
389  479
390  480
391  481
392  482
393  483
394  484
395  485
396  486
397  487
398  488
399  489
400  490
401  491
402  492
403  493
404  494
405  495
406  496
407  497
408  498
409  499
410  500
411  501
412  502
413  503
414  504
415  505
416  506
417  507
418  508
419  509
420  510
421  511
422  512
423  513
424  514
425  515
426  516
427  517
428  518
429  519
430  520
431  521
432  522
433  523
434  524
435  525
436  526
437  527
438  528
439  529
440  530
441  531
442  532
443  533
444  534
445  535
446  536
447  537
448  538
449  539
450  540
451  541
452  542
453  543
454  544
455  545
456  546
457  547
458  548
459  549
460  550
461  551
462  552
463  553
464  554
465  555
466  556
467  557
468  558
469  559
470  560
471  561
472  562
473  563
474  564
475  565
476  566
477  567
478  568
479  569
480  570
481  571
482  572
483  573
484  574
485  575
486  576
487  577
488  578
489  579
490  580
491  581
492  582
493  583
494  584
495  585
496  586
497  587
498  588
499  589
500  590
501  591
502  592
503  593
504  594
505  595
506  596
507  597
508  598
509  599
510  600
511  601
512  602
513  603
514  604
515  605
516  606
517  607
518  608
519  609
520  610
521  611
522  612
523  613
524  614
525  615
526  616
527  617
528  618
529  619
530  620
531  621
532  622
533  623
534  624
535  625
536  626
537  627
538  628
539  629
540  630
541  631
542  632
543  633
544  634
545  635
546  636
547  637
548  638
549  639
550  640
551  641
552  642
553  643
554  644
555  645
556  646
557  647
558  648
559  649
560  650
561  651
562  652
563  653
564  654
565  655
566  656
567  657
568  658
569  659
570  660
571  661
572  662
573  663
574  664
575  665
576  666
577  667
578  668
579  669
580  670
581  671
582  672
583  673
584  674
585  675
586  676
587  677
588  678
589  679
590  680
591  681
592  682
593  683
594  684
595  685
596  686
597  687
598  688
599  689
600  690
601  691
602  692
603  693
604  694
605  695
606  696
607  697
608  698
609  699
610  700
611  701
612  702
613  703
614  704
615  705
616  706
617  707
618  708
619  709
620  710
621  711
622  712
623  713
624  714
625  715
626  716
627  717
628  718
629  719
630  720
631  721
632  722
633  723
634  724
635  725
636  726
637  727
638  728
639  729
640  730
641  731
642  732
643  733
644  734
645  735
646  736
647  737
648  738
649  739
650  740
651  741
652  742
653  743
654  744
655  745
656  746
657  747
658  748
659  749
660  750
661  751
662  752
663  753
664  754
665  755
666  756
667  757
668  758
669  759
670  760
671  761
672  762
673  763
674  764
675  765
676  766
677  767
678  768
679  769
680  770
681  771
682  772
683  773
684  774
685  775
686  776
687  777
688  778
689  779
690  780
691  781
692  782
693  783
694  784
695  785
696  786
697  787
698  788
699  789
700  790
701  791
702  792
703  793
704  794
705  795
706  796
707  797
708  798
709  799
710  800
711  801
712  802
713  803
714  804
715  805
716  806
717  807
718  808
719  809
720  810
721  811
722  812
723  813
724  814
725  815
726  816
727  817
728  818
729  819
730  820
731  821
732  822
733  823
734  824
735  825
736  826
737  827
738  828
739  829
740  830
741  831
742  832
743  833
744  834
745  835
746  836
747  837
748  838
749  839
750  840
751  841
752  842
753  843
754  844
755  845
756  846
757  847
758  848
759  849
760  850
761  851
762  852
763  853
764  854
765  855
766  856
767  857
768  858
769  859
770  860
771  861
772  862
773  863
774  864
775  865
776  866
777  867
778  868
779  869
780  870
781  871
782  872
783  873
784  874
785  875
786  876
787  877
788  878
789  879
790  880
791  881
792  882
793  883
794  884
795  885
796  886
797  887
798  888
799  889
800  890
801  891
802  892
803  893
804  894
805  895
806  896
807  897
808  898
809  899
810  900
811  901
812  902
813  903
814  904
815  905
816  906
817  907
818  908
819  909
820  910
821  911
822  912
823  913
824  914
825  915
826  916
827  917
828  918
829  919
830  920
831  921
832  922
833  923
834  924
835  925
836  926
837  927
838  928
839  929
840  930
841  931
842  932
843  933
844  934
845  935
846  936
847  937
848  938
849  939
850  940
851  941
852  942
853  943
854  944
855  945
856  946
857  947
858  948
859  949
860  950
861  951
862  952
863  953
864  954
865  955
866  956
867  957
868  958
869  959
870  960
871  961
872  962
873  963
874  964
875  965
876  966
877  967
878  968
879  969
880  970
881  971
882  972
883  973
884  974
885  975
886  976
887  977
888  978
889  979
890  980
891  981
892  982
893  983
894  984
895  985
896  986
897  987
898  988
899  989
900  990
901  991
902  992
903  993
904  994
905  995
906  996
907  997
908  998
909  999
910  1000
911  1001
912  1002
913  1003
914  1004
915  1005
916  1006
917  1007
918  1008
919  1009
920  1010
921  1011
922  1012
923  1013
924  1014
925  1015
926  1016
927  1017
928  1018
929  1019
930  1020
931  1021
932  1022
933  1023
934  1024
935  1025
936  1026
937  1027
938  1028
939  1029
940  1030
941  1031
942  1032
943  1033
944  1034
945  1035
946  1036
947  1037
948  1038
949  1039
950  1040
951  1041
952  1042
953  1043
954  1044
955  1045
956  1046
957  1047
958  1048
959  1049
960  1050
961  1051
962  1052
963  1053
964  1054
965  1055
966  1056
967  1057
968  1058
969  1059
970  1060
971  1061
972  1062
973  1063
974  1064
975  1065
976  1066
977  1067
978  1068
979  1069
980  1070
981  1071
982  1072
983  1073
984  1074
985  1075
986  1076
987  1077
988  1078
989  1079
990  1080
991  1081
992  1082
993  1083
994  1084
995  1085
996  1086
997  1087
998  1088
999  1089
1000 1090
1001 1091
1002 1092
1003 1093
1004 1094
1005 1095
1006 1096
1007 1097
1008 1098
1009 1099
1010 1100
1011 1101
1012 1102
1013 1103
1014 1104
1015 1105
1016 1106
1017 1107
1018 1108
1019 1109
1020 1110
1021 1111
1022 1112
1023 1113
1024 1114
1025 1115
1026 1116
1027 1117
1028 1118
1029 1119
1030 1120
1031 1121
1032 1122
1033 1123
1034 1124
1035 1125
1036 1126
1037 1127
1038 1128
1039 1129
1040 1130
1041 1131
1042 1132
1043 1133
1044 1134
1045 1135
1046 1136
1047 1137
1048 1138
1049 1139
1050 1140
1051 1141
1052 1142
1053 1143
1054 1144
1055 1145
1056 1146
1057 1147
1058 1148
1059 1149
1060 1150
1061 1151
1062 1152
1063 1153
1064 1154
1065 1155
1066 1156
1067 1157
1068 1158
1069 1159
1070 1160
1071 1161
1072 1162
1073 1163
1074 1164
1075 1165
1076 1166
1077 1167
1078 1168
1079 1169
1080 1170
1081 1171
1082 1172
1083 1173
1084 1174
1085 1175
1086 1176
1087 1177
1088 1178
1089 1179
1090 1180
1091 1181
1092 1182
1093 1183
1094 1184
1095 1185
1096 1186
1097 1187
1098 1188
1099 1189
1100 1190
1101 1191
1102 1192
1103 1193
1104 1194
1105 1195
1106 1196
1107 1197
1108 1198
1109 1199
1110 1200
1111 1201
1112 1202
1113 1203
1114 1204
1115 1205
1116 1206
1117 1207
1118 1208
1119 1209
1120 1210
1121 1211
1122 1212
1123 1213
1124 1214
1125 1215
1126 1216
1127 1217
1128 1218
1129 1219
1130 1220
1131 1221
1132 1222
1133 1223
1134 1224
1135 1225
1136 1226
1137 1227
1138 1228
1139 1229
1140 1230
1141 1231
1142 1232
1143 1233
1144 1234
1145 1235
1146 1236
1147 1237
1148 1238
1149 1239
1150 1240
1151 1241
1152 1242
1153 1243
1154 1244
1155 1245
1156 1246
1157 1247
1158 1248
1159 1249
1160 1250
1161 1251
1162 1252
1163 1253
1164 1254
1165 1255
1166 1256
1167 1257
1168 1258
1169 1259
1170 1260
1171 1261
1172 1262
1173 1263
1174 1264
1175 1265
1176 1266
1177 1267
1178 1268
1179 1269
1180 1270
1181 1271
1182 1272
1183 1273
1184 1274
1185 1275
1186 1276
1187 1277
1188 1278
1189 1279
1190 1280
1191 1281
1192 1282
1193 1283
1194 1284
1195 1285
1196 1286
1197 1287
1198 1288
1199 1289
1200 1290
1201 1291
1202 1292
1203 1293
1204 1294
1205 1295
1206 1296
1207 1297
1208 1298
1209 1299
1210 1300
1211 1301
1212 1302
1213 1303
1214 1304
1215 1305
1216 1306
1217 1307
1218 1308
1219 1309
1220 1310
1221 1311
1222 1312
1223 1313
1224 1314
1225 1315
1226 1316
1227 1317
1228 1318
1229 1319
1230 1320
1231 1321
1232 1322
1233 1323
1234 1324
1235 1325
1236 1326
1237 1327
1238 1328
1239 1329
1240 1330
1241 1331
1242 1332
1243 1333
1244 1334
1245 1335
1246 1336
1247 1337
1248 1338
1249 1339
1250 1340
1251 1341
1252 1342
1253 1343
1254 1344
1255 1345
1256 1346
1257 1347
1258 1348
1259 1349
1260 1350
1261 1351
1262 1352
1263 1353
1264 1354
1265 1355
1266 1356
1267 1357
1268 1358
1269 1359
1270 1360
1271 1361
1272 1362
1273 1363
1274 1364
1275 1365
1276 1366
1277 1367
1278 1368
1279 1369
1280 1370
1281 1371
1282 1372
1283 1373
1284 1374
1285 1375
1286 1376
1287 1377
1288 1378
1289 1379
1290 1380
1291 1381
1292 1382
1293 1383
1294 1384
12
```

Fetch Product Details Dynamically:



Tropical Vibe

★★★★★ (23 Reviews)

\$550

Color:



Size:

S ▾

- 1 +

Add to Cart

Compare

Description Additional Information Reviews (0)

Escape to paradise with TropicalVibe—a collection that brings the vibrant energy of the tropics right into your home. Inspired by lush landscapes, warm sunshine, and vibrant colors, TropicalVibe offers a refreshing, tropical-inspired aesthetic that creates a relaxed and inviting atmosphere in any space. Designed for those who dream of a laid-back, vacation-like ambiance, TropicalVibe captures the essence of island living with its lively prints, natural textures, and bold colors. Whether you're looking to add a touch of the tropics to your living room, bedroom, or outdoor patio, TropicalVibe seamlessly infuses any area with a sense of adventure and tranquility. Crafted from high-quality materials, each piece in the TropicalVibe collection is both durable and stylish. The collection features elements such as leafy

Prepared By: Muhammad Nehal Nadeem.

Documentation:

This documentation explains a simple and effective way to build dynamic and responsive marketplace components. By using Sanity CMS for managing the backend and modular frontend development techniques, the application becomes scalable, efficient, and offers a great user experience. It is a helpful guide to create strong and easy-to-use marketplace applications, particularly those who are new to modern web development.

Each feature, like showing products or placing orders, is crucial for building a marketplace that works well in the real world. Sanity CMS helps by making content management simple and flexible. At the same time, using modular components makes the frontend easier to develop and update, ensuring the app runs smoothly and can be easily maintained over time.

In conclusion, this documentation emphasizes the importance of a well-structured approach to building a dynamic and scalable marketplace. By integrating Sanity CMS for seamless backend management and utilizing modular frontend techniques, the application ensures both efficiency and flexibility. With each component contributing to a professional and user-centered platform.

Final Checklist:

- ☒ **Frontend Component Development.**

- ☒ **Styling and Responsiveness.**
- ☒ **Code Quality.**
- ☒ **Documentation and Submission.**
- ☒ **Final Review.**

Prepared By: Muhammad Nehal Nadeem.