

## Day 4 - Advanced Documentation on Dynamic Frontend Components - Comforty

### Objective:

To develop dynamic and responsive frontend components for the marketplace, integrating reusable structures, efficient data fetching, and state management while addressing challenges such as API latency, dynamic routing, and pagination.

### Procedures Undertaken for Component Development and Integration:

#### 1. Initialization and Data Acquisition:

- Established a connection between the frontend and Sanity CMS through the Sanity client, ensuring secure and efficient communication.
- Validated the structural integrity and accessibility of all data models, including ``Products`` and ``Categories``, via API endpoints.
- Engineered reusable and scalable data-fetching functions for essential components such as ``ProductList``, ``CategoryFilter``, and ``SearchBar``.

#### 2. Development of Core Components:

##### Product Listing Component:

- Dynamically rendered product data in a grid layout optimized for responsive design.
- Leveraged card-based interfaces to display key attributes such as product name, pricing, and inventory status.

##### Product Detail Component:

- Utilized dynamic routing within Next.js to generate unique pages for individual product entries.
- Integrated detailed product attributes, including descriptions, pricing, and high-resolution imagery.

## Category Filter Component:

- Dynamically fetched category data from APIs to facilitate product categorization
- Enabled real-time filtering of products based on user-selected categories.
  - **Search Bar**:
- Implemented advanced search functionalities to allow filtering of products via names and associated tags.
  - **Pagination Component**:
- Incorporated intuitive navigation mechanisms such as "previous" and "next" buttons to handle extensive product catalogs efficiently.

### 3. Styling and Adaptive Design:

- Applied Tailwind CSS to achieve a unified, aesthetically pleasing, and mobile-responsive user interface.
- Ensured adaptability of component layouts to various screen sizes through dynamic styling methodologies.

### 4. Global State Management:

- Adopted React Context to establish a global state management system for the cart, wishlist, and order confirmation functionalities
- This approach facilitated seamless communication between components and enhanced data persistence across the application.

## Identified Challenges and Corresponding Solutions:

### 1. Challenge: API Latency and Response Delays

#### Issues:

- Prolonged response times during data fetching hindered component rendering efficiency.
- Encountered CORS-related errors while fetching data due to misconfigured origin settings in Sanity CMS.

### **Solutions:**

- Incorporated a loading state and skeleton UI to provide visual feedback during data retrieval.
- Adjusted CORS configurations in Sanity CMS to whitelist the frontend's origin, enabling uninterrupted data flow.

## **2. Challenge: Errors in Dynamic Routing**

### **Issue:**

- Invalid or missing product IDs resulted in failures during page rendering for product details.

### **Solution:**

- : Introduced robust error handling mechanisms and designed fallback pages to gracefully handle missing or invalid product data.

## **3. Challenge: Complex Filtering and Pagination Integration**

### **Issue:**

- Coordinating multiple filters (e.g., category, price range) with pagination presented challenges in maintaining state consistency.

### **Solution:**

- Implemented URL-based query parameters to synchronize filtering and pagination states across browser reloads.

## **Adopted Best Practices:**

### **Component Reusability:**

- Developed modular and reusable components, including ``ProductCard`` and ``CategoryFilter``, to promote scalability and maintainability.

## Secure Configuration Management:

- Utilized ``.env.local`` for storing sensitive API keys, enhancing overall security and adherence to industry standards.

## Error Mitigation:

- Employed comprehensive error-handling strategies to manage API failures and ensure a seamless user experience.

## Responsive Design Principles:

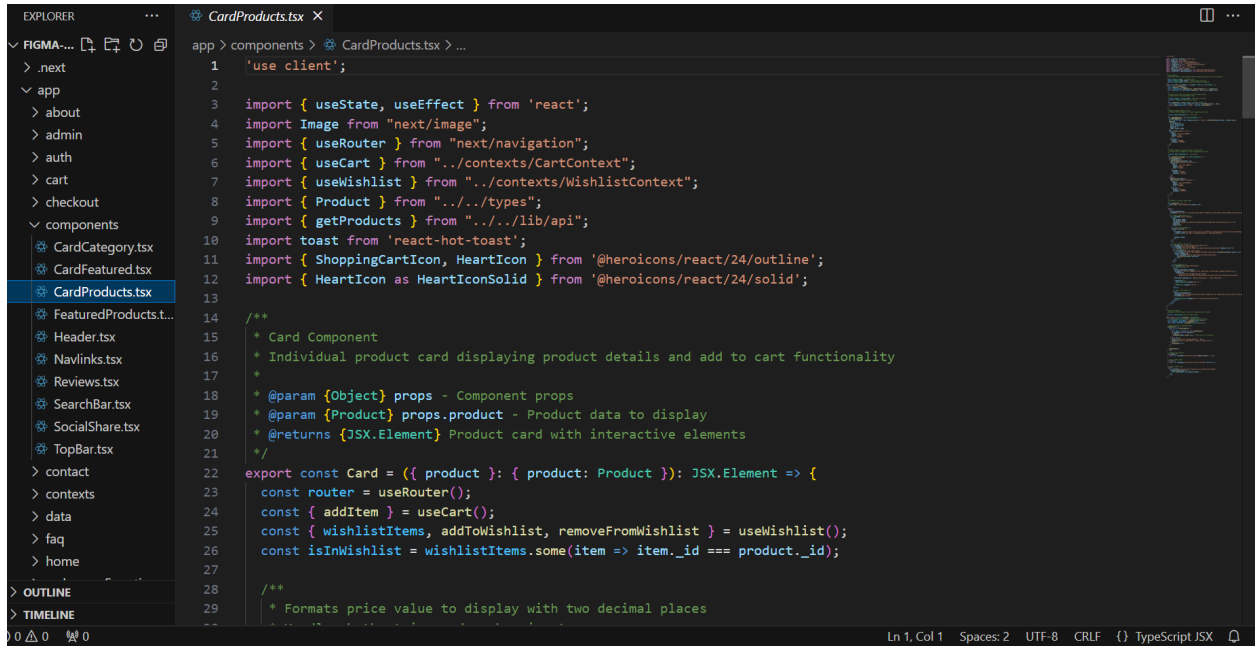
- Thoroughly tested the application across multiple device resolutions to guarantee consistent and accessible user interfaces

## High-Quality Code Standards:

- Adopted descriptive naming conventions and implemented detailed code comments to facilitate readability and future development efforts.

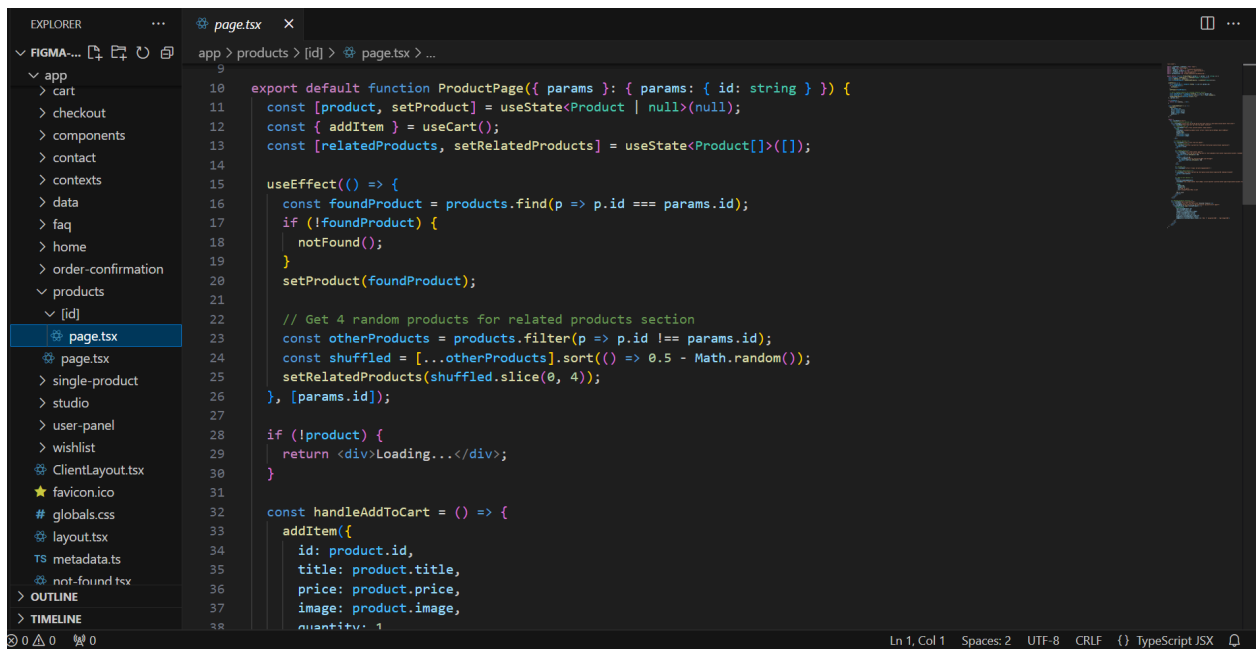
## Screenshots:

## Card Featured:



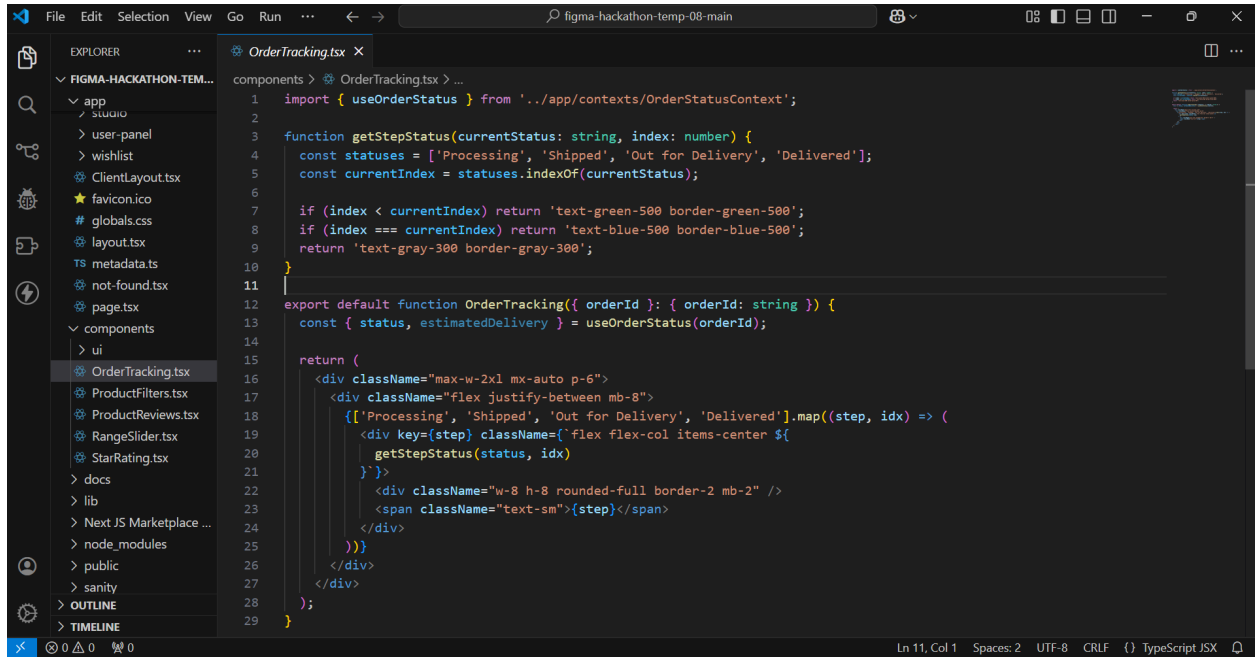
```
1 'use client';
2
3 import { useState, useEffect } from 'react';
4 import Image from 'next/image';
5 import { useRouter } from 'next/navigation';
6 import { useCart } from '../contexts/CartContext';
7 import { useWishlist } from '../contexts/WishlistContext';
8 import { Product } from '../types';
9 import { getProducts } from '../lib/api';
10 import toast from 'react-hot-toast';
11 import { ShoppingCartIcon, HeartIcon } from '@heroicons/react/24/outline';
12 import { HeartIcon as HeartIconSolid } from '@heroicons/react/24/solid';
13
14 /**
15  * Card Component
16  * Individual product card displaying product details and add to cart functionality
17  *
18  * @param {Object} props - Component props
19  * @param {Product} props.product - Product data to display
20  * @returns {JSX.Element} Product card with interactive elements
21  */
22 export const Card = ({ product }: { product: Product }): JSX.Element => {
23   const router = useRouter();
24   const { addItem } = useCart();
25   const { wishlistItems, addToWishlist, removeFromWishlist } = useWishlist();
26   const isInWishlist = wishlistItems.some(item => item._id === product._id);
27
28   /**
29    * Formats price value to display with two decimal places
30    */
31   const formattedPrice = product.price.toFixed(2);
32
33   return (
34     <div>
35       <div>
36         <img alt={product.image} data-bbox={product.image} data-cs={100} data-kind="parent" data-rs={1} data-rs={1} data-rs={1} data-rs={1} data-rs={1} data-rs={1} data-rs={1} data-rs={1} data-rs={1} data-kind="ghost"/>
37         </div>
38       </div>
39       <div>
40         <h3>{product.title}</h3>
41         <p>{formattedPrice}</p>
42         <div>
43           <button>Add to Cart</button>
44           <button>Add to Wishlist</button>
45         </div>
46       </div>
47     </div>
48   );
49 }
```

## Dynamic Routing:



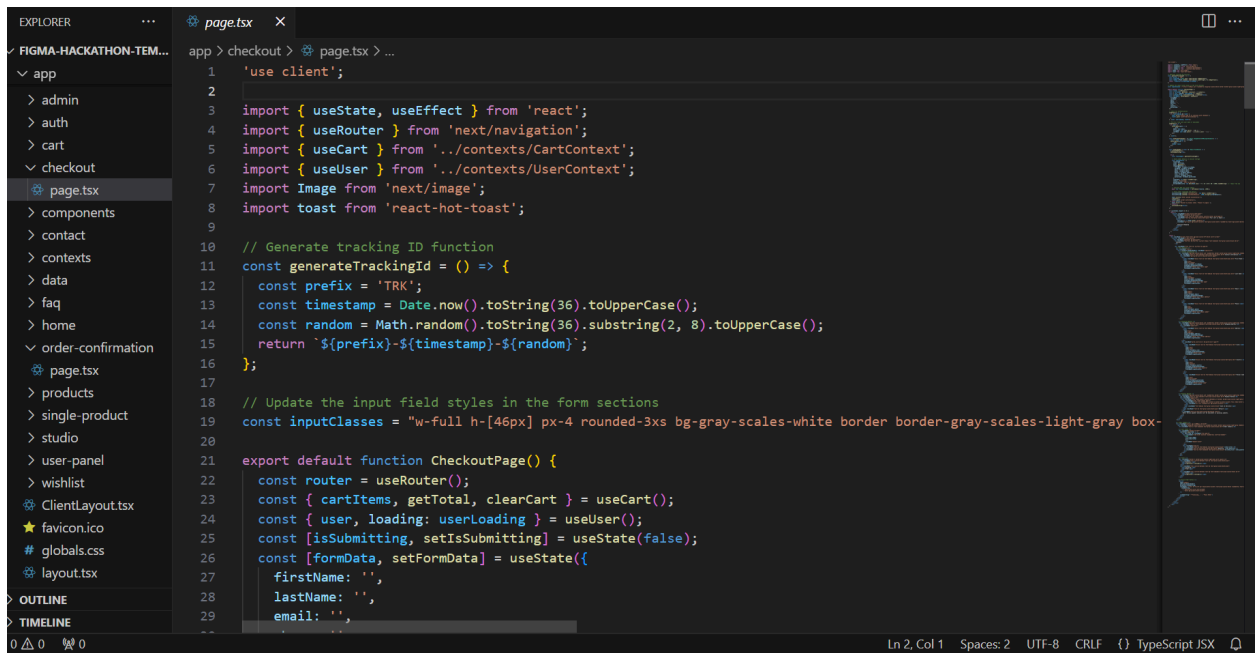
```
10 export default function ProductPage({ params }: { params: { id: string } }) {
11   const [product, setProduct] = useState<Product | null>(null);
12   const { addItem } = useCart();
13   const [relatedProducts, setRelatedProducts] = useState<Product[]>([]);
14
15   useEffect(() => {
16     const foundProduct = products.find(p => p.id === params.id);
17     if (!foundProduct) {
18       notFound();
19     }
20     setProduct(foundProduct);
21
22     // Get 4 random products for related products section
23     const otherProducts = products.filter(p => p.id !== params.id);
24     const shuffled = [...otherProducts].sort(() => 0.5 - Math.random());
25     setRelatedProducts(shuffled.slice(0, 4));
26   }, [params.id]);
27
28   if (!product) {
29     return <div>Loading...</div>;
30   }
31
32   const handleAddToCart = () => {
33     addItem({
34       id: product.id,
35       title: product.title,
36       price: product.price,
37       image: product.image,
38       quantity: 1
39     });
40   };
41
42   return (
43     <div>
44       <div>
45         <img alt={product.image} data-bbox={product.image} data-cs={100} data-kind="parent" data-rs={1} data-rs={1} data-rs={1} data-rs={1} data-rs={1} data-rs={1} data-rs={1} data-rs={1} data-kind="ghost"/>
46         </div>
47       </div>
48       <div>
49         <h3>{product.title}</h3>
50         <p>{product.price}</p>
51         <div>
52           <button>Add to Cart</button>
53           <button>Add to Wishlist</button>
54         </div>
55       </div>
56     </div>
57   );
58 }
```

## Order Tracking:



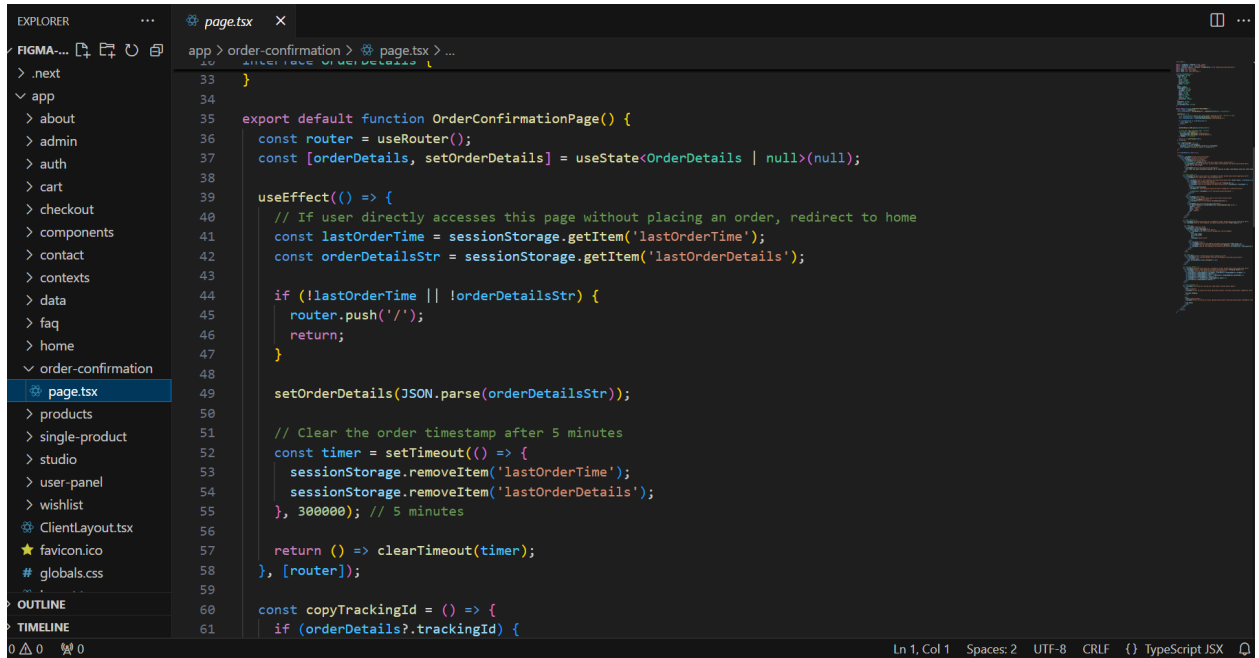
```
1 import { useOrderStatus } from '../app/contexts/OrderStatusContext';
2
3 function getStepStatus(currentStatus: string, index: number) {
4   const statuses = ['Processing', 'Shipped', 'Out for Delivery', 'Delivered'];
5   const currentIndex = statuses.indexOf(currentStatus);
6
7   if (index < currentIndex) return 'text-green-500 border-green-500';
8   if (index === currentIndex) return 'text-blue-500 border-blue-500';
9   return 'text-gray-300 border-gray-300';
10 }
11
12 export default function OrderTracking({ orderId }: { orderId: string }) {
13   const { status, estimatedDelivery } = useOrderStatus(orderId);
14
15   return (
16     <div className="max-w-2xl mx-auto p-6">
17       <div className="flex justify-between mb-8">
18         {['Processing', 'Shipped', 'Out for Delivery', 'Delivered'].map((step, idx) => (
19           <div key={step} className="flex flex-col items-center ${
20             getStepStatus(status, idx)
21           }">
22             <div className="w-8 h-8 rounded-full border-2 mb-2" />
23             <span className="text-sm">{step}</span>
24           </div>
25         ))}
26       </div>
27     </div>
28   );
29 }
```

## Checkout:



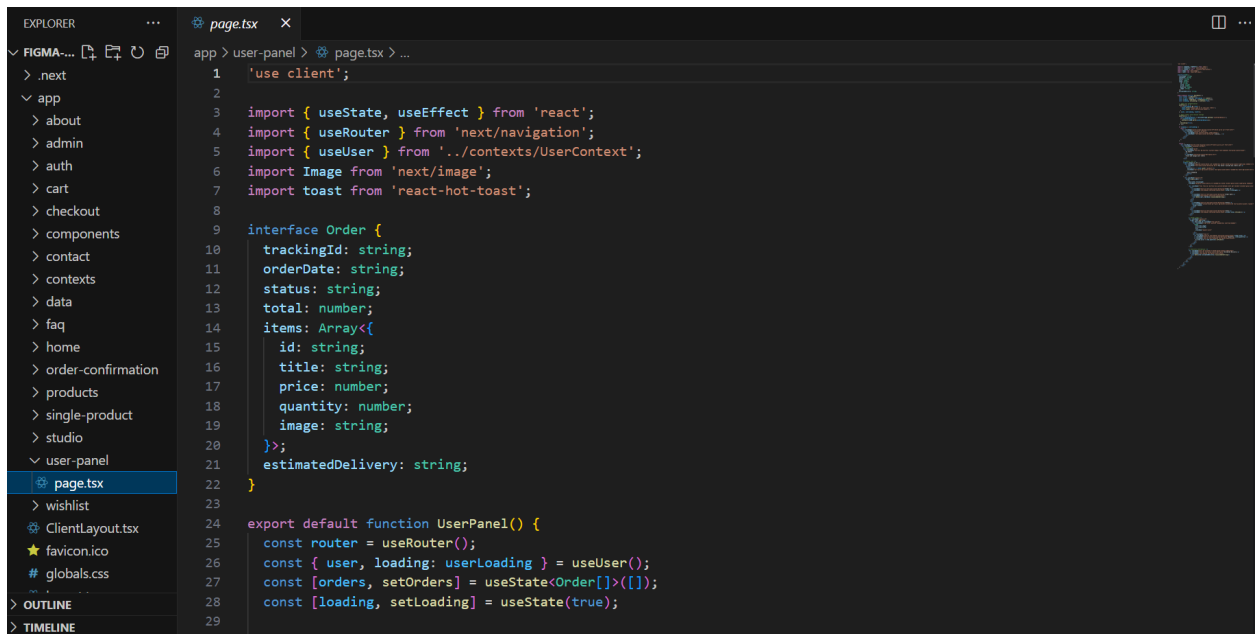
```
1 'use client';
2
3 import { useState, useEffect } from 'react';
4 import { useRouter } from 'next/navigation';
5 import { useCart } from '../contexts/CartContext';
6 import { useUser } from '../contexts/UserContext';
7 import Image from 'next/image';
8 import toast from 'react-hot-toast';
9
10 // Generate tracking ID function
11 const generateTrackingId = () => {
12   const prefix = 'TRK';
13   const timestamp = Date.now().toString(36).toUpperCase();
14   const random = Math.random().toString(36).substring(2, 8).toUpperCase();
15   return `${prefix}-${timestamp}-${random}`;
16 };
17
18 // Update the input field styles in the form sections
19 const inputClasses = "w-full h-[46px] px-4 rounded-3xs bg-gray-scales-white border border-gray-scales-light-gray box-";
20
21 export default function CheckoutPage() {
22   const router = useRouter();
23   const { cartItems, getTotal, clearCart } = useCart();
24   const { user, loading: userLoading } = useUser();
25   const [isSubmitting, setIsSubmitting] = useState(false);
26   const [formData, setFormData] = useState({
27     firstName: '',
28     lastName: '',
29     email: '',
30   });
31 }
```

## Order Confirmation:



```
33 }
34
35 export default function OrderConfirmationPage() {
36   const router = useRouter();
37   const [orderDetails, setOrderDetails] = useState<OrderDetails | null>(null);
38
39   useEffect(() => {
40     // If user directly accesses this page without placing an order, redirect to home
41     const lastOrderTime = sessionStorage.getItem('lastOrderTime');
42     const orderDetailsStr = sessionStorage.getItem('lastOrderDetails');
43
44     if (!lastOrderTime || !orderDetailsStr) {
45       router.push('/');
46       return;
47     }
48
49     setOrderDetails(JSON.parse(orderDetailsStr));
50
51     // Clear the order timestamp after 5 minutes
52     const timer = setTimeout(() => {
53       sessionStorage.removeItem('lastOrderTime');
54       sessionStorage.removeItem('lastOrderDetails');
55     }, 300000); // 5 minutes
56
57     return () => clearTimeout(timer);
58   }, [router]);
59
60   const copyTrackingId = () => {
61     if (orderDetails?.trackingId) {
```

## Basic User Panel:



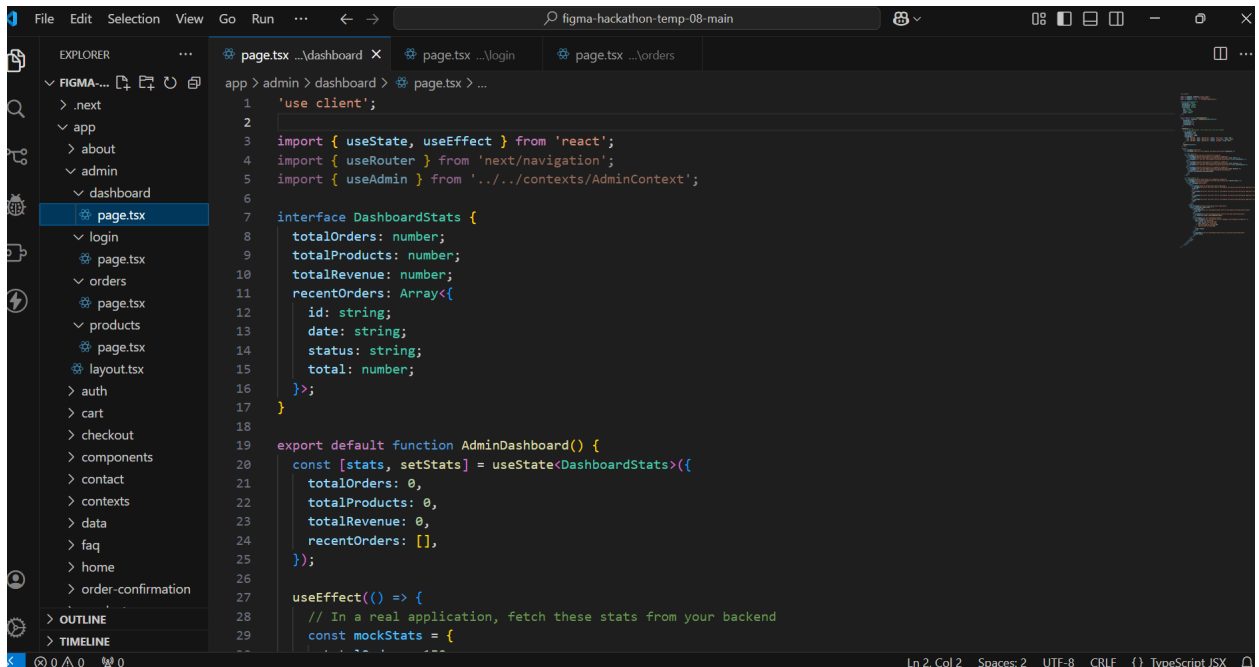
```
1 'use client';
2
3 import { useState, useEffect } from 'react';
4 import { useRouter } from 'next/navigation';
5 import { useUser } from '../contexts/UserContext';
6 import Image from 'next/image';
7 import toast from 'react-hot-toast';
8
9 interface Order {
10   trackingId: string;
11   orderDate: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

## Wishlist:



```
1 'use client';
2
3 import { useWishlist } from '../contexts/WishlistContext';
4 import CardFeatured from '../components/CardFeatured';
5 import { TrashIcon } from '@heroicons/react/24/outline';
6
7 export default function WishlistPage() {
8   const { wishlistItems, removeFromWishlist } = useWishlist();
9
10  return (
11    <section className="w-auto py-12 font-inter">
12      <div className="container mx-auto px-4">
13        <h1 className="text-2xl md:text-4xl lg:text-[32px] font-semibold text-gray-scales-black mb-12">
14          My Wishlist
15        </h1>
16
17        {wishlistItems.length === 0 ? (
18          <div className="flex items-center justify-center py-10">
19            <p className="text-lg text-gray-scales-dark-gray">Your wishlist is empty.</p>
20          </div>
21        ) : (
22          <div className="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-4 gap-6">
23            {wishlistItems.map((item) => (
24              <div key={item.id} className="relative">
25                <CardFeatured
26                  id={item.id}
27                  imageSrc={item.image}
28                  title={item.title}
29                  price={item.price}
30                />
31              <button
32                type="button"
33                className="absolute top-0 right-0 p-2 text-gray-scales-white"
34                onClick={() => removeFromWishlist(item.id)}
35              >
36                <TrashIcon />
37              </button>
38            </div>
39            </div>
40          )
41        )}
42      </div>
43    </section>
44  );
45}
```

## Admin Dashboard:



```
1 'use client';
2
3 import { useState, useEffect } from 'react';
4 import { useRouter } from 'next/navigation';
5 import { useAdmin } from '../contexts/AdminContext';
6
7 interface DashboardStats {
8   totalOrders: number;
9   totalProducts: number;
10  totalRevenue: number;
11  recentOrders: Array<{
12    id: string;
13    date: string;
14    status: string;
15    total: number;
16  }>;
17 }
18
19 export default function AdminDashboard() {
20   const [stats, setStats] = useState<DashboardStats>({
21     totalOrders: 0,
22     totalProducts: 0,
23     totalRevenue: 0,
24     recentOrders: [],
25   });
26
27   useEffect(() => {
28     // In a real application, fetch these stats from your backend
29     const mockStats = {
30       totalOrders: 10,
31       totalProducts: 5,
32       totalRevenue: 100,
33       recentOrders: [
34         { id: '1', date: '2023-10-26', status: 'pending', total: 20 },
35         { id: '2', date: '2023-10-25', status: 'shipped', total: 15 },
36         { id: '3', date: '2023-10-24', status: 'cancelled', total: 10 },
37         { id: '4', date: '2023-10-23', status: 'pending', total: 15 },
38         { id: '5', date: '2023-10-22', status: 'shipped', total: 10 },
39       ],
40     };
41     setStats(mockStats);
42   }, []);
43
44   return (
45     <div>
46       <div>Admin Dashboard</div>
47       <div>
48         <div>Stats</div>
49         <div>Recent Orders</div>
50       </div>
51     </div>
52   );
53 }
```



## Products API Calls:

```
19 /**
20  * Product-related API functions
21  */
22 export const ProductAPI = {
23   /**
24    * Fetch all products with optional pagination
25    */
26   async getAll(page = 1, limit = config.api.itemsPerPage): Promise<Product[]> {
27     const start = (page - 1) * limit;
28     const end = start + limit;
29
30     const query = `*[_type == "products"] | order(_createdAt desc) [{start}...${end}] {
31       _id,
32       title,
33       price,
34       priceWithoutDiscount,
35       description,
36       "image": image.asset->url,
37       badge,
38       inventory,
39       tags,
40       category->
41     }`;
42
43     const [data, error] = await handleError(client.fetch(query), {
44       context: 'ProductAPI.getAll',
45       page,
46       limit,
47     });
```