Day3 – API Integration Report –[urooba Ecommerce]

Webhooks

CORS origins

Tokens

CORS origins

Hosts that can connect to the project API.

ORIGIN	CREDENTIALS	CREATED	
http://localhost:3000	Allowed	yesterday	🗑
http://localhost:3333	Allowed	yesterday	🗑

Tokens

Tokens are used to authenticate apps and scripts to access project data.

NAME	PERMISSIONS	CREATED	
token-temp1.yt	Editor	yesterday	🗑

S Urooba Akram ⌄ temp-1-yt ⌄ ⚡ 29 days left in trial ⓘ Ⓤ

🚀 Getting started ⊞ Overview 👥 Members ▦ Studios 🗄 Datasets 🔒 Access ⋀ Activity 📈 Usage ⚡ Plan ⌗ API ⚙ Settings

Webhooks

CORS origins

Tokens

http://localhost:3333	Allowed	yesterday	🗑

Tokens

Tokens are used to authenticate apps and scripts to access project data.

NAME	PERMISSIONS	CREATED	
token-temp1.yt	Editor	yesterday	🗑

IONAL

```tsx
 8    import { Description } from "@radix-ui/react-dialog";
 9    import sanityClient from "@sanity/client";
10
11    const sanity = sanityClient({
12        projectId: "ijm6r3v6",
13        dataset: "production",
14        apiVersion: "2023-01-01",
15        useCdn: true,
16    });
17
18    interface Product {
19        _id: string;
20        title:string;
21        price:number;
22        description: string;
23        discountPercentage: number;
24        imageUrl: string;
25        productImage: {
26            assest:{
27                _ref:string;
28
29            };
30        };
31        tags: string[];
32    }
33    const ProductCards: React.FC= () => {
34        const [products, setProducts] = useState<Product[]>([]);
35        const [cart, setCart] = useState<Product[]>([]);
36
37        const fetchProducts = async () => {
38            try {
39                const query = `
```

```tsx
33    const ProductCards: React.FC= () => {
37        const fetchProducts = async () => {
39                const query = `
40                *[type == "product"] {
41                    _id,
42                    title,
43                    price,
44                    description,
45                    discountPercentage,
46                    "imageUrl": productImage.assest->url,
47                    tags,
48                }
49                `;
50
51                const data = await sanity.fetch(query);
52                setProducts(data);
53
54            } catch (error) {
55                console.error("Error Fetching Products:" , error);
56            }
57        };
58
59
60    const addCart = (product: Product) => {
61      setCart((prevCart) => [...prevCart, product]);
62      alert( `${product.title} has been added to your cart!`)
63    };
64
65    const truncateDescription = (Description:string) => {
66        return Description.length > 100 ? Description.substring(0,100) + "..." : Description;
67    };
68
```

```
                                                          ← →                    ⌕ website                                    🐙 ˅           ▭ ▯ ▭ ▭   – ◻ ✕

📄  File  Edit  Selection  View  Go  Run  ···

⌕     EXPLORER: WEBSI...  🗋 🗂 ↻ 🗗  ···    v.local       TS product.ts U    TS index.ts U     {} package.json M    JS importData.mjs U   TS next.config.ts U    ⚙ page.tsx ...\product U ✕    ⇄ ▭  ···

⌕     ˅ src                         ●         src › app › product › ⚙ page.tsx › [e] ProductCards › [e] addCart
      ˅ app                         ●          33     const ProductCards: React.FC= () => {
⦚        ˅ components               ●          67   };
            🔲 sh1t.tsx             U          68
            ⚙ size.tsx             U          69     useEffect(() =>{
            ⚙ slider.tsx           U          70         fetchProducts();
         ˅ product                  ●          71       }, []);
            ⚙ page.tsx             U          72
         ˅ products                 ●          73
            › [id]                  ●          74     return(
            ⚙ page.tsx             U          75         <div className="p-4">
            ⚙ sell.tsx             U          76             <h2 className="text-center 🔲text-slate-800 mt-4 mb-4">Products From APIs Data</h2>
         › studio                   ●          77
            ⭐ favicon.ico                     78             <div className="grid grid-cols-1 sm:grid-cols-2 md:grid-cols-3 lg:grid-cols-4 gap-6">
            # globals.css           ●          79             {products.map((product) => (
            ⚙ layout.tsx           M          80                 <div
            ⚙ page.tsx             M          81                 key={product._id}
         ˅ components\ui            ●          82                 className="g-white shadow-md rounded-lgp-4 hover:shadow-lg transition-shadow duration-300">
            ⚙ accordion.tsx        U          83
            ⚙ breadcrumb.tsx       U          84                     <Image
            ⚙ button.tsx           U          85                     src={product.imageUrl}
            ⚙ card.tsx             U          86                     alt={product.title}
            ⚙ carousel.tsx         U          87                     width={300}
            ⚙ checkbox.tsx         U          88                     height={300}
            ⚙ dropdown-menu.tsx    U          89                     className="w-full h-48 object-cover rounded-md"/>
            ⚙ input.tsx            U          90
            ⚙ label.tsx            U          91                     <div className="mt-4">
            ⚙ navigation-menu.tsx  U          92                         <h2 className="text-lg font-semibold">{product.title}</h2>
            ⚙ slider.tsx           U          93                         <p text-slate-800 mt-2 tex-sm>{truncateDescription(product.description)}</p>
                                               94                         <div className="flex justify-between items-center mt-4">
                                               95                         <div>
                                               96                             <p className="🔲text-slate-600 font-bold"> ${product.price}</p>
                                               97                             {product.discountPercentage > 0 && (

⊗      ⚡ master* ⌖ ⊘ ⊗0 ⚠0  ⌨0  β                                                                      Ln 61, Col 49   Spaces: 4   UTF-8   CRLF  {} TypeScript JSX  ⦿ Go Live  ⌲
```

```tsx
const ProductCards: React.FC= () => {
  {products.map((product) => (

                <p className="text-sm text-green-600">
                  {product.discountPercentage}% OFF
                </p>
              )}
            </div>

          </div>
  <div className="mt-2 flex flex-wrap gap-2">
    {product.tags.map((tag, index) =>(

      <span key={index}
        className="text-xs bg-slate-400 text-black rounded-full x-2 py-1">
          {tag}

      </span>


    ))}
  </div>

  {/*Add to cart functionality */}


  <button
  className="mt-4 w-full bg-blue-600 text-white py-2 rounded-md hover:bg-blue-700"
  onClick={() => addCart(product)}
  >
    Add To Cart
```

```tsx
const ProductCards: React.FC= () => {
  {products.map((product) => (
    Add To Cart
  </button>
            </div>
            </div>

    ))}
  </div>
  {/* cart summary */}

      <div className="mt-8 bg-slate-100 p-6 rounded-lg shadow-md">
        <h2 className="text-lg font-black text-red-800">Cart Summary</h2>
        {cart.length > 0 ? (
          <ul className="space-y-4">
            {cart.map((item, index) => (
              <li key={index}
                className="flex justify-between items-center bg-white shadow-sm p-4 rounded-md"
              >
                <div>
                  <p className="font-medium text-slate-900">{item.title}</p>
                  <p className="text-sm text-blue-600">${item.price.toFixed(2)}</p>
                </div>
  <Image
  src={item.imageUrl}
  alt={item.title}
  width={50}
  height={50}
  className="rounded-md"/>
```

TS product.ts U   TS index.ts U   {} package.json M   JS importData.mjs U   TS next.config.ts U   page.tsx ...\product U

- src
  - app
    - components
      - shift.tsx
      - size.tsx   U
      - slider.tsx   U
      - product
        - page.tsx   U
      - products
        - [id]
        - page.tsx   U
        - sell.tsx   U
      - studio
      - favicon.ico
      - globals.css   M
      - layout.tsx   M
      - page.tsx   M
    - components\ui
      - accordion.tsx   U
      - breadcrumb.tsx   U
      - button.tsx   U
      - card.tsx   U
      - carousel.tsx   U
      - checkbox.tsx   U
      - dropdown-menu.tsx   U
      - input.tsx   U
      - label.tsx   U
      - navigation-menu.tsx   U
      - slider.tsx   U

src > app > product > page.tsx > ProductCards > addCart

```
 33     const ProductCards: React.FC= () => {
141                           {cart.map((item, index) => (
154     height={50}
155     className="rounded-md"/>
156
157                        </li>
158
159                     ))}
160
161
162                 </ul>
163         )  :  (
164             <p className="text-black text-center">Your Cart Is Empty Add Products</p>
165
166            )}
167         </div>
168     </div>
169    );
170 };
171
172 export default ProductCards;
173
174
```