

# HACKATHON DAY-4

---

DYNAMIC FRONTEND COMPONENTS  
& MANY MORE!

# BUILDING DYNAMIC FRONTEND COMPONENTS

## Objective:

The goal for Day 04 is designing and developing frontend components to display marketplace data fetched from Sanity CMS or APIs. In this day we're deep diving into creating dynamic responsive and scalable components for an advanced e-commerce website. From designing sleek product listing pages to implementing dynamic routes for personalized product views, the focus is on crafting user centric interface powered by real-time data.

## KEY DYNAMIC COMPONENTS

### 1 Product List Component

Display products dynamically based on API data. Includes filters for categories, price range and ratings.

Product Name	Price	Image	Stock Status
--------------	-------	-------	--------------

### 2 Product Detail Component

It includes product detail pages using dynamic routing in Next.js.

Product Description	Price	Available Sizes or Colors
---------------------	-------	---------------------------

- 3 Category Component: It includes
- Display categories dynamically fetch from the data source.
  - Enable filtering of products by selecting categories.

4 Search Bar: It includes search functionality to filter products by name or tags.

5 Cart Component: It includes display added items, quantity and total price.

6 Wishlist Component: It includes users to save products for future reference.

7 Checkout Flow Component: It includes a multi-step form for checkout, including fields for Billing and shipping address. Payment details.

8 User Profile Component: It includes display user-specific details like

- Name, email and saved addresses.
- Order history with links to individual orders.

9 Reviews And Ratings Component:

It allows users to view and submit reviews for products. Display average ratings and individual reviews dynamically.

10

## Pagination Component

It includes break down large product lists into manageable pages. It implement previous and next buttons or numbered pagination.

11.

## Filter Panel Component

It includes advanced filtering options such as Price range sliders, Brand selection, Availability

12.

## Related Products Component

It includes fetch data based on tags, categories or customer behaviors.

13

## Footer and Header Component

- Build consistent navigation and branding elements
- Include links to key pages (e.g Home, About, Contact). It ensures responsiveness and accessibility

14

## Notifications Component: It includes

real-time alerts for actions like adding to cart, errors or successful purchases.

15

## Analytics Dashboard Component

It display key performance indicators like sales, user traffic and popular products

16

## Product Comparison Component

It allow users to select and compare multiple products side by side.

17

## Multi-Language Support Component

Add a language switcher to enable multilingual support for the marketplace. It includes fetch translation dynamically for different languages.

18

## Order Tracking Component

It includes real-time updates on their status. Include detail like estimated delivery time and current location.

19

## Help Center Component

It includes searchable question section. Add contact forms or chatbot integration for support.

20

## Subscription Management Component

It allows users to subscribe to products or services. It manages subscription details and renewals.

21

## Admin Dashboard Component

includes an interface for managing products, users and categories.

22

## Discount and Promotion Component

It allows admins to create and manage discount coupons and promotional campaigns.

Display discounts dynamically on relevant product listings.

23

## Social Media Sharing Component

It includes to enable users to share products.

Date: 19-01-25

or promotions directly to social media platforms.

24 **Bulk Upload Component** It allows admin to upload large amounts of products data via CSV or Excel files.

25 **AI Recommendations Component**

It includes displaying recommendations dynamically on the home or product detail pages.

26 **Gift Card and Voucher Component**

It allows users to purchase and redeem gift cards or vouchers. Track gift card balance and use.

27 **Customer Feedback Component**

It creates a form for users to submit feedback about specific products.

28 **Advanced Search Component**

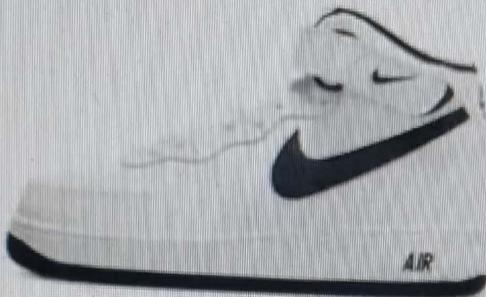
It implements faceted search to combine multiple filters and provide precise results.

## IMPLEMENTATION TIPS

State Management: Use tools like Redux or Context API for managing global state.

Responsive Design: Ensure components adapt to different screen sizes using frameworks like Tailwind CSS.

Performance Optimization: Use React memoization



13295

Nike Zoom Fly 5



11295

Nike Air Force 1 PLT.AF.ORM



8695

## Nike Standard Issue Basketball Jersey



2895

Nike Air Max 270



13295

Nike Zoom Fly 5



EXPLORER ... page.tsx ... \shoes M X TS queries.ts U TS products.ts U TS product.t

PRACTICE-UI... D+ E+ ⌂ ⌂

app components dontmiss essential firstnav footer gearup icon man news shoes

page.tsx M

app > components > shoes > page.tsx > [ ] SHOES > product.map() callback

```
1 "use client"
2
3 import { client } from "@sanity/lib/client";
4 import { urlFor } from "@sanity/lib/image";
5 import { allProducts } from "@sanity/lib/queries";
6 import { Product } from "@types/products";
7 import Image from "next/image";
8 import React, { useEffect, useState } from "react";
9
10 const SHOES = () => {
11
12   const [product, setProduct] = useState<Product[]>([])
13
14   useEffect(() => {
15     async function fetchproduct() {
16       const fetchedProduct : Product[] = await client.fetch(allProducts);
17       setProduct(fetchedProduct)
18     }
19     fetchproduct()
20   },[])
21 }
```

Tabnine | Edit | Explain

const SHOES = () => {  
 const [product, setProduct] = useState<Product[]>([])  
  
 useEffect(() => {  
 async function fetchproduct() {  
 const fetchedProduct : Product[] = await client.fetch(allProducts);  
 setProduct(fetchedProduct)  
 }  
 fetchproduct()  
 },[])

Pieces: Comment | Pieces: Explain  
useEffect(() => {  
 async function fetchproduct() {  
 const fetchedProduct : Product[] = await client.fetch(allProducts);  
 setProduct(fetchedProduct)  
 }  
 fetchproduct()  
},[])

main\* ⌂ Pieces Settings ⌂ 0 ▲ 0 ⌂ 0 CRLF {} TypeScript JSX Pieces Code Lens ⌂ Go Live ⌂ Completions limit reached

PLORER

...

page.tsx ...\\shoes M X

TS queries.ts U

TS products.ts U

TS product.ts U

RACTICE-UIUXHACKATHON

app

components

&gt; dontmiss

&gt; essential

&gt; firstnav

&gt; footer

&gt; gearup

&gt; icon

&gt; man

&gt; news

shoes

page.tsx M

&gt; snav

&gt; tnav

&gt; women

&gt; fonts

&gt; fourcompo

&gt; seccompo

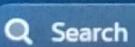
&gt; studio

OUTLINE

TIMELINE

app &gt; components &gt; shoes &gt; page.tsx &gt; [ ] SHOES &gt; [ ] useEffect() callback &gt; [ ] fetchpr

```
1 "use client"  
2  
3 import { client } from "@/sanity/lib/client";  
4 import { urlFor } from "@/sanity/lib/image";  
5 import { allProducts } from "@/sanity/lib/queries";  
6 import { Product } from "@/types/products";  
7 import Image from "next/image";  
8 import React, { useEffect, useState } from "react";  
9  
Tabnine | Edit | Explain  
10 const SHOES = () => {  
11  
12 const [product, setProduct] = useState<Product[]>([])  
13  
Pieces: Comment | Pieces: Explain  
14 useEffect(() => {  
    Pieces: Comment | Pieces: Explain  
15     async function fetchproduct() {  
16         const fetchedProduct : Product[] = await client.fetch(allPr)  
17         setProduct(fetchedProduct)  
18     }  
19     fetchproduct()  
20 },[])  
21
```



File Edit Selection View ... ← → 🔍 practice-uiuxhackathon 🌐

EXPLORER ... page.tsx ...\\shoes 1, M X TS queries.ts U TS products.ts U TS product.ts U TS next

app > components > shoes > page.tsx > [?] SHOES > [?] useEffect() callback > [?] fetchproduct > [?] fe

```
1 "use client"
2
3 import { client } from "@sanity/lib/client";
4 import { urlFor } from "@sanity/lib/image";
5 import { allProducts } from "@sanity/lib/queries";
6
7 const four: string = "https://fontawesome.com/v4.7.0/icon/brands/fa-foursquare";
8
9
10 const SHOES = () => {
11   const [product, setProduct] = useState<Product>(
12     null,
13     () => {
14       useEffect(() => {
15         async function fetchproduct() {
16           const fetchedProduct: Product[] = await client.fetch(`[fourLetter]`);
17           setProduct(fetchedProduct);
18         }
19         fetchproduct();
20       }, []);
21     }
22   );
23
24   return (
25     <div>
26       <h1>{product.name}</h1>
27       <p>${product.description}</p>
28       <img alt="Product image" src={urlFor(product.image).url()} />
29     </div>
30   );
31 }
32
33 export default SHOES;
```

Update import from "@sanity/lib/queri... X [?] four

- [?] FaFoursquare
- [?] FaFoursquare
- [?] ImFoursquare
- [?] PiFourK
- [?] PiFourKBold
- [?] PiFourKDuotone
- [?] PiFourKFill
- [?] PiFourKLight
- [?] PiFourKThin
- [?] SiFoursquare
- [?] SiFoursquarecityguide

Tabnine | Edit | Fix | Explain

Pieces: Comment | Pieces: Explain

useEffect(() => {  
 Pieces: Comment | Pieces: Explain  
 async function fetchproduct() {  
 const fetchedProduct : Product[] = await client.fetch([fourLetter]);  
 setProduct(fetchedProduct);  
 }  
 fetchproduct();  
}, []);

Activate W Go to Settings

main\* 🔍 Pieces Settings ✎ 1 △ 0 ⚡ 0 CRLF ⓘ TypeScript JSX Pieces Code Lens ⚡ Go Live 🌐 Completions limit reached 🌐 tabnine ba

Search



## EXPLORER

...

page.tsx ...shoes



## PRACTICE-UIUXHACKATHON

app &gt; components &gt;



&lt; app

&lt; components



&gt; dontmiss



&gt; essential



&gt; firstnav



&gt; footer

...

&gt; gearup



&gt; icon

&gt; man



&gt; news

&gt; shoes

&gt; snav

&gt; tnav

> women  
> fonts> fourcompo  
> seccompo

&gt; studio

&gt; OUTLINE

&gt; TIMELINE

1, M

● ●

●

●

●

Add import from

const BsFillEm

Tabnine | Edit |

const SHOE

10

11

12

13

14

15

16

17

18

19

20

21

Pieces: Comm

useEffect

Pieces: Cor

async fu

const

setPro

}

fetchprod

},[])



main\*



Pieces Settings



1 △ 0



0

CRLF

{ } Typ

EXPLORER

PRACTICE-UIUXHACKATHON

- > .next
- < app
  - < components
    - > air
    - > dontmiss
    - > essential
    - > firstnav
    - > footer
    - > gearup
    - > icon
    - > man
    - > news
    - < shoes
      - page.tsx 1, M
      - > snav
      - > tnav
      - > women
    - > fonts
    - > fourcompo
    - > seccompo
    - > studio
    - > thirdcompo
    - ★ favicon.ico
    - # globals.css
  - > OUTLINE
  - > TIMELINE

page.tsx app

page.tsx ...shoes 1, M TS queries.ts U TS products.ts U TS pro

app > components > shoes > page.tsx > [●] SHOES > [●] useEffect() callback > [●] fetchproduct > [●] fet

```
1 "use client"
2
3 import { Product } from "@/types/products";
4 import Image from "next/image";
5 import React, { useEffect, useState } from "react";
6
7 Tabnine | Edit | Fix | Explain
8 const SHOES = () => {
9
10   const [product, setProduct] = useState<Product>([])
```

Pieces: Comment | Pieces: Explain  
useEffect(() => {  
 Pieces: Comment | Pieces: Explain  
 async function fetchproduct() {  
 const fetchedProduct : Product[] = await

Add import from "react-icons/ai"

```
11
12   const AiFillAccountBook
13   const AiFillAlert
14   const AiFillAlipayCircle
15   const AiFillAlipaySquare
16   const AiFillAliwangwang
17   const AiFillAmazonCircle
18   const AiFillAmazonSquare
19   const AiFillAndroid
20   const AiFillApi
21   const AiFillApple
22   const AiFillAppstore
23   const AiFillAudio
```

16 const AiFillAccountBook: IconType

18 return (  
19 <div className="max-w-6xl mx-auto px-4">  
20 <Image alt="Placeholder for account book icon" width={100} height={100}/>  
21 </div>

22 );

23 );

25 export default SHOES;

26

main\* ↻ Piec Settings ⌂ 1 ▲ 0 ⌂ 0 In 13, Col 43 Spaces: 2 CRLF ( TypeScript JSX Piec Code Lens Go Live Comp

Edit Selection View Go Run ... ← → ⚡ practice-uiuhackathon 08

EXPLORER PRACTICE-UIUXHACKATHON .next app components air dontmiss essential firstnav footer gearup icon man news shoes page.tsx 1, M snav tnav women fonts fourcompo seccompo studio thirdcompo favicon.ico # globals.css > OUTLINE > TIMELINE

page.tsx app page.tsx ...shoes 1, M TS queries.ts U TS products.ts U TS product.ts U TS next.config.ts M

app > components > shoes > page.tsx > (0) SHOES > product.map() callback

```
9  const SHOES = () => {
10
11
12
13  useEffect(() => {
14    async function fetchproduct() {
15      const fetchedProduct : Product[] = await client.fetch(allProducts)
16      setProdcut(fetchedProduct)
17    }
18    fetchproduct()
19  },[])
20
21
22  return (
23    <div className="max-w-6xl mx-auto px-4 py-8">
24      {product.map((product) => [
25        <div
26          key={product._id}
27          className={`relative aspect-square overflow-hidden rounded-lg transition ${product.outOfStock? "opacity-50" : ""}`}
28        >
29          {product.image && (
30            <Image
31              src={product.image.asset.url}
32              alt={product.title}
33              width={300}
34              height={300}
35            />
```

Activate Windows  
Go to Settings to activate

main Piecess Settings 1 △ 0 🔍 In 25, C49 Sponsor UI 4 ⓘ TypeScript JS Plugins Code Lens Go Live Completions limit reached tabnine basic

Search

[Find a Store](#) | [Help](#) | [Join Us](#) | [Sign In](#)

New & Featured Men Women Kids Sale SNKRS

Search



## Nike Air Force 1 Mid '07

The Nike Air Force 1 Mid '07 is a reimagining of the iconic Air Force 1, first introduced in 1982. This model features a mid-top silhouette that offers added ankle support, making it ideal for those who need extra stability while on the move. Crafted from premium leather, the Air Force 1 Mid '07 offers both style and durability, making it perfect for everyday wear. The classic design paired with modern touches, ensures these sneakers are not only a nod to heritage but also a versatile piece for contemporary fashion. The adjustable ankle strap adds to the customization, allowing you to wear them in your own unique style.

MRP: ₹ 10,695.00

Add To Cart

Activate Windows  
Go to Settings to activate Windows.

[Find a Store](#) | [Help](#) | [Join Us](#) | [Sign In](#)

New & Featured Men Women Kids Sale SNKRS

Search



## Nike Dri-FIT Ready

The Nike Dri-FIT Ready is a versatile fitness top that is ideal for high-intensity workouts. Made with Dri-FIT technology, this top pulls sweat away from your skin, ensuring that you stay cool and dry even during the most strenuous exercises. The lightweight, stretchy fabric allows for unrestricted movement, making it perfect for activities like running, yoga, or weight training. The sleek design and breathable material also make it great for casual wear or lounging around. With its modern fit and stylish appearance, the Nike Dri-FIT Ready top is a must-have for anyone who values both comfort and style in their activewear. Whether you're working out or just relaxing, this top offers all-day comfort and support.

MRP: ₹ 2,495.00

Add To Cart

Activate Windows  
Go to Settings to activate Windows.



New(500)

Shop  
Sports Bra  
Tops & Tees  
Hoodies  
& Sweatshirts  
Jackets  
Trousers & Tights  
Shorts  
Trackuits  
Jumpsuits & Rompers  
Shirts & Dresses  
Socks  
Accessories  
& Equipment

Gender ▾



New & Featured

Men

Women

Kids

Sale

Shoes

Find a Store | Help | Join Us | Sign In

0 items



Hide Filters Sort By



Last In  
**Nike Air Force 1 Mid '07**  
Many Colors  
1 Color  
\$109.99-\$129.99



Last In  
**Nike Court Vision Low Next Nature**  
Many Colors  
1 Color  
\$109.99-\$129.99



Last In  
**Nike Air Max 1 PLATFORM**  
Women's Clothing  
1 Color  
\$109.99-\$129.99



Last In  
**Nike Air Force 1 React**  
Women's Clothing  
1 Color  
\$109.99-\$129.99

Activate Windows

Go to Settings to activate Windows





Nike Standard Issue  
Basketball Jersey  
\$2895



Nike Air Max 270  
\$13295



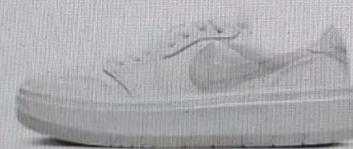
Nike Zoom Fly 5  
\$11295



Nike Air Force 1  
PLT.AF.ORM  
\$8695



Nike Air Force 1 React  
\$13295



Air Jordan 1 Elevate Low  
\$11895



Nike Dri-FIT UV Hyverse  
\$2495



Nike Court Vision Low  
\$5695



Activate Windows  
Go to Settings to activate



Activate Windows  
Go to Settings to activate Windows.

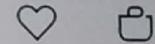


Find a Store | Help | Join Us | Sign In



New & Featured Men Women Kids Sale SNKRS

Q Search



## Hello Nike App

Download the app to access everything Nike. [Get Your Great](#)



Date 19-01-25

techniques to avoid unnecessary re-renders.

### Notes

Test all components thoroughly to ensure proper integration with backend APIs.

Use placeholder or mock data during initial development stages to simulate API responses.

Focus on accessibility to make components usable for all users.

## CONCLUSION

Dynamic frontend components are vital for creating a modern and user-friendly marketplace. By focusing on reusability, responsiveness, and seamless API integration, these components lay the foundation for a polished user experience.