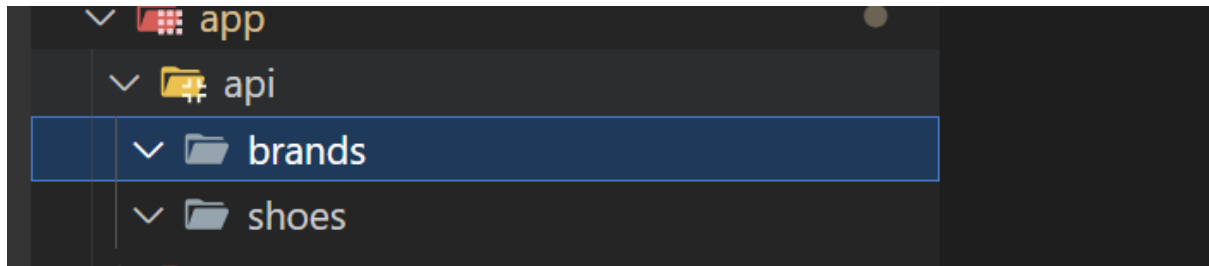


Nama : Farid Hambali

NPM : 22312112



Membuat folder api yang berisi brands dan shoes

```
app > api > brands > route.ts > GET
1 import { NextResponse } from 'next/server';
2 export async function GET() {
3   const brands = [
4     { id: 'adidas', name: 'Adidas', logo: '/images/adidas-logo.png' },
5     { id: 'ortuseight', name: 'Ortuseight', logo: '/images/ortus-logo.png' },
6     { id: 'salomon', name: 'Salomon', logo: '/images/salomon-logo.png' },
7     { id: 'nike', name: 'Nike', logo: '/images/nike-logo.png' },
8     { id: 'newbalance', name: 'New Balance', logo: '/images/newbalance-logo.png' },
9     { id: 'on', name: 'On', logo: '/images/on-logo.png' },
10  ];
11
12  return NextResponse.json(brands);
13 }
```

API untuk data merek sepatu

```

app > api > shoes > route.ts > GET
1  import { NextResponse } from 'next/server';
2
3  export async function GET(request: Request) {
4    try {
5      const { searchParams } = new URL(request.url);
6      const brand = searchParams.get('brand'); // Mendapatkan parameter "brand" dari URL
7
8      // Data sepatu dengan harga dan ukuran
9      const shoes: Record<string, Array<{ id: string; name: string; image: string; price: number; sizes: string[] }>> = {
10        adidas: [
11
12        ],
13        ortuseight: [
14
15        ],
16        nike: [
17
18        ],
19        newbalance: [
20
21        ],
22        salomon: [
23
24        ],
25        on: [
26
27        ]
28      ];
29
30      const shoeData = brand && shoes[brand] ? shoes[brand] : [];
31
32      return NextResponse.json(shoeData);
33    } catch (error) {
34      console.error('Error handling request:', error);
35      return NextResponse.json({ error: 'An error occurred while processing your request.' }, { status: 500 });
36    }
37  }
38
39  }
40

```

API untuk data sepatu dari setiap merek

```

page.tsx app  route.ts ...brands  route.ts ...shoes  page.tsx ...[id] U X  Untitled-1
app > brands > [id] > page.tsx > Shoe
1  "use client";
2  import { useEffect, useState } from 'react';
3
4  type Shoe = {
5    id: string;
6    name: string;
7    image: string;
8    price: number;
9    sizes: string[];
10 };

```

Halaman dinamis untuk sepatu per merek

```

app > brands > [id] > page.tsx > BrandDetail
11
12 export default function BrandDetail({ params }: { params: { id: string } }) {
13   const [shoes, setShoes] = useState<Shoe[]>([]);
14   const [selectedShoe, setSelectedShoe] = useState<Shoe | null>(null);
15   const [isModalOpen, setIsModalOpen] = useState<boolean>(false);
16
17   useEffect(() => {
18     const fetchShoes = async () => {
19       const res = await fetch(`/api/shoes?brand=${params.id}`);
20       const data = await res.json();
21       setShoes(data);
22     };
23
24     fetchShoes();
25   }, [params.id]);
26
27   const handleShoeClick = (shoe: Shoe) => {
28     setSelectedShoe(shoe);
29     setIsModalOpen(true); // Show the modal
30   };
31
32   const closeModal = () => {
33     setIsModalOpen(false);
34   };

```

Membuat komponen untuk menampilkan detail merek berdasarkan id