

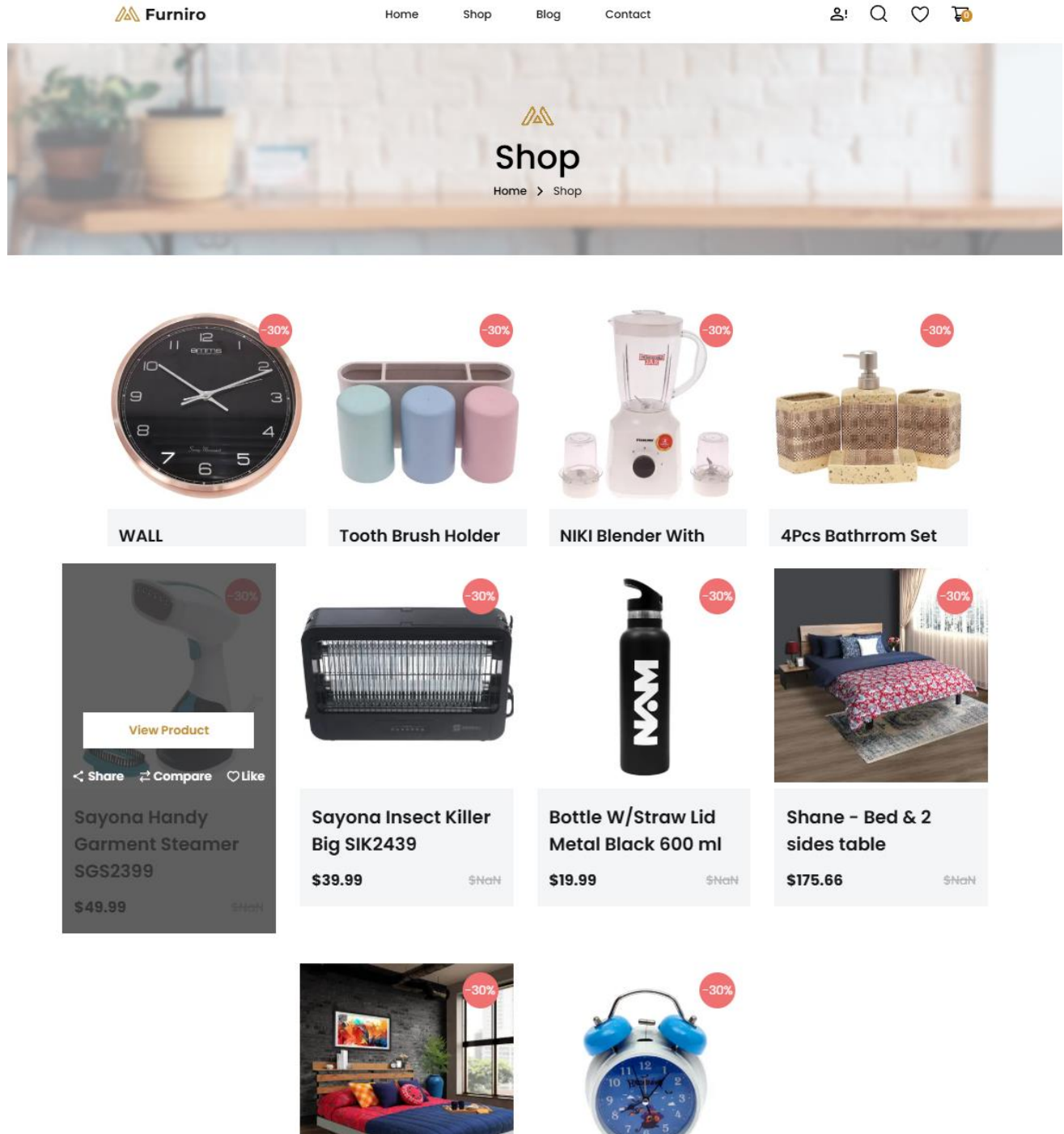


# Day 4 - Dynamic Frontend Components - Hiperstar

# MARKETPLACE BUILDER HACKATHON 2025

## DAY 4: Dynamic Frontend Components - Hiperstar

### 1. Functional Deliverables:





**High Quality**  
crafted from top  
materials



**Warranty  
Protection**  
Over 2 years



**Free Shipping**  
Order over 150 \$



**24 / 7 Support**  
Dedicated support

## Furniro.

400 University Drive Suite 200 Coral  
Gables,  
FL 33134 USA

### Links

[Home](#)

[Shop](#)

[Contact](#)

### Help

[Payment Options](#)

[Returns](#)

[Privacy Policies](#)

### Newsletter

[Enter Your Email Address](#)

[SUBSCRIBE](#)

2023 furino. All rights reserved



[Home](#)

[Shop](#)

[Blog](#)

[Contact](#)



[Home](#) > [Shop](#) >

[Tooth Brush Holder W/ Tumbler Set.E1615](#)



# Tooth Brush Holder W/ Tumbler Set.E1615

\$14.99

Introducing our stylish and functional Tissue Holder, the perfect addition to any home or office space. With its sleek design and durable construction, our Tissue Holder keeps your tissues neatly organized and easily accessible. Enhance your decor wh...

- 1 +

[Add To Cart](#)

[Comparison](#)


SKU : 45f0b

Category: Bath

Share : [f](#) [in](#) [t](#)


Introducing our stylish and functional Tissue Holder, the perfect addition to any home or office space. With its sleek design and durable construction, our Tissue Holder keeps your tissues neatly organized and easily accessible. Enhance your decor while adding convenience with our premium Tissue Holder. Buy now for a clutter-free and elegant solution! Delivery time is 5-7 working days

Related Products




WALL  
CLOCK.METAL..AMB-  
641

\$64.17




Tooth Brush Holder  
W/ Tumbler  
Set.E1615

\$14.99



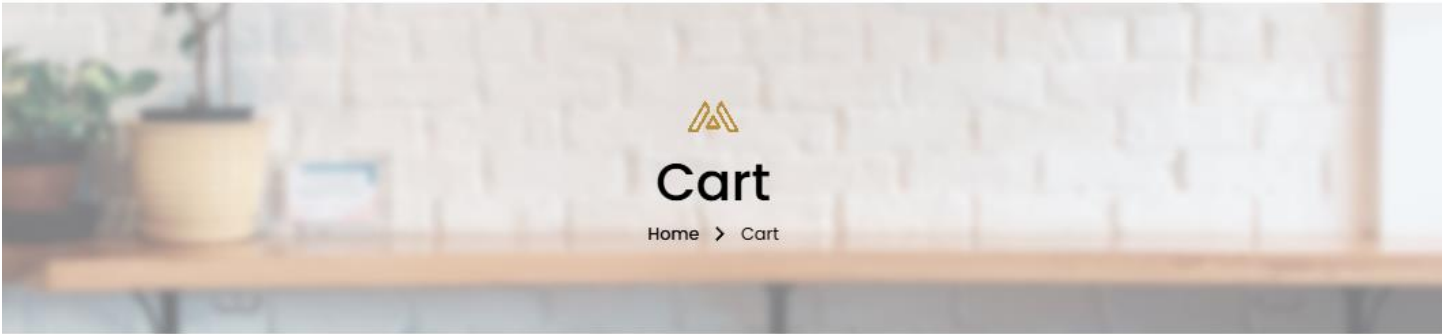
NIKI Blender With  
Mill NB1705/1900A

\$49.99





4Pcs Bathrrom Set  
BT-009

\$19.99



Cart

[Home](#) > [Cart](#)

Image	Product	Price	Quantity	Subtotal	
	Tooth Brush Holder W/ Tumbler Set.E1615	14.99	1	14.99	

Cart Totals

Subtotal \$14.99

Total \$14.99

Check Out



# Checkout

[Home](#) > [Checkout](#)

## Billing Details

First Name

Last Name

Company Name (Optional)

Company Name (Optional)

Country / Region

City

Street Address

Zip code

## Product

Tooth Brush Holder W/ Tumbler Set.E1615 x1

Total

## Subtotal

\$14.99

**\$14.99**☐ Direct Bank Transfer

Make your payment directly into our bank account. Please use your Order ID as the payment reference. Your order will not be shipped

☐ Direct Bank Transfer

Make your payment directly into our bank account. Please use your Order ID as the payment reference. Your order will not be shipped until the funds have cleared in our account.

☒ Cash on Delivery

Your personal data will be used to support your experience throughout this website, to manage access to your account, and for other purposes described in our [privacy policy](#).

[Place Order](#)

## 2. Code Deliverables:

### Product Cart Component:

```
"use client"
import React, { useState } from 'react'
import { urlFor } from '@/utils/sanityImageBuilder'
import Image from 'next/image';
import { useRouter } from 'next/navigation';
import { productData } from '@/types';
function ProductCard({ data }: {data:productData}) {
  const router = useRouter()
  const productImageUrl = data?.main_image
    ? urlFor(data.main_image)?.url()
    : null;
  const [showPopup, setShowPopup] = useState(false)
  return (
    <div className='w-[285px] relative' onMouseLeave={() => setShowPopup(false)}
onMouseEnter={() => setShowPopup(true)}>
      <div className='flex justify-center items-center bg-[#E97171] absolute top-5
right-5 rounded-full w-[48px] h-[48px]'>
        <p className='text-white text-[16px]'>-30%</p>
      </div>
      <productImageUrl && (
        <Image src={productImageUrl} width={300} height={300} className="m-auto"
style={{width:"300px",height:"300px",objectFit:"contain"}} alt='Dining'/>
      )>
      <div className='bg-[#F4F5F7] p-4'>
        <h2 className='font-semibold text-[24px] py-1'>{data?.product_name}</h2>
        <p className='text-[16px] font-medium text-[#898989] py-1'></p>
        <div className='flex flex-wrap items-center justify-between py-1'>
          <p className='font-bold text-[20px]'>${data?.price}</p>
          <span className='text-[#B0B0B0] text-[16px] font-
regular'><s>${Number(data?.price+ (data?.price * 0.10)).toFixed(2)}</s></span>
        </div>
      </div>
      <div className='flex flex-col justify-center items-center absolute top-0
left-0 bg-[#3A3A3A]/80 w-full h-full'>
        <button onClick={() => router.push(`/shop/${data?.slug.current}`)}
className='bg-white text-[#B88E2F] text-[16px] font-semibold w-[80%] px-14 py-3 my-0.5'>View
Product</button>
        <div className='flex flex-wrap justify-between items-center gap-5 mt-6
px-2'>
          <div className='flex items-center cursor-pointer'>
            <svg width="18" height="18" viewBox="0 0 16 16" fill="none"
xmlns="http://www.w3.org/2000/svg">
              <path d="M10 9.66671C9.47467 9.66671 9 9.87337 8.644
10.2047L3.94 7.46671C3.97333 7.31337 4 7.16004 4 7.00004C4 6.84004 3.97333 6.68671 3.94
6.53337L8.64 3.79337C9 4.12671 9.47333 4.33337 10 4.33337C11.1067 4.33337 12 3.44004 12
2.33337C12 1.22671 11.1067 0.333374 10 0.333374C8.89333 0.333374 8 1.22671 8 2.33337C8
2.49337 8.02667 2.64671 8.06 2.80004L3.36 5.54004C3 5.20671 2.52667 5.00004 2
```

```

5.00004C0.893333 5.00004 0 5.89337 0 7.00004C0 8.10671 0.893333 9.00004 2 9.00004C2.52667
9.00004 3 8.79337 3.36 8.46004L8.05867 11.2054C8.02112 11.3563 8.00143 11.5112 8 11.6667C8
12.0623 8.1173 12.4489 8.33706 12.7778C8.55682 13.1067 8.86918 13.3631 9.23463
13.5145C9.60009 13.6658 10.0022 13.7054 10.3902 13.6283C10.7781 13.5511 11.1345 13.3606
11.4142 13.0809C11.6939 12.8012 11.8844 12.4448 11.9616 12.0569C12.0387 11.6689 11.9991
11.2668 11.8478 10.9013C11.6964 10.5359 11.44 10.2235 11.1111 10.0038C10.7822 9.784 10.3956
9.66671 10 9.66671Z" fill="white" />
      </svg>
      <p className='text-white font-semibold text-[16px] pl-
0.5'>Share</p>
    </div>
    <div className='flex items-center cursor-pointer'>
      <svg width="18" height="18" viewBox="0 0 16 16" fill="none"
xmlns="http://www.w3.org/2000/svg">
        <path d="M9.08004 6L10.08 7L13.52 3.55L10 0L9.00004 1L10.8
2.8H1.00004V4.2H10.82L9.08004 6ZM4.86004 8L3.86004 7L0.420044 10.5L3.91004 14L4.91004
13L3.10004 11.2H13V9.8H3.10004L4.86004 8Z" fill="white" />
      </svg>
      <p className='text-white font-semibold text-[16px] pl-
0.5'>Compare</p>
    </div>
    <div className='flex items-center cursor-pointer'>
      <svg width="18" height="18" viewBox="0 0 17 16" fill="none"
xmlns="http://www.w3.org/2000/svg">
        <path d="M7.99973 13.0361C-5.33333 5.66669 3.99999 -2.33331
7.99973 2.72539C12 -2.33331 21.3333 5.66669 7.99973 13.0361Z" stroke="white"
strokeWidth="1.8" />
      </svg>
      <p className='text-white font-semibold text-[16px] pl-
0.5'>Like</p>
    </div>
  </div>
</div>
)}
</div>
)
}

export default ProductCard

```

## Product Details Component

```

"use client"
import React, { useEffect, useState } from 'react'
import AddToCartButton from './AddToCartButton'
import { useStore } from '@store'
import RelatedProducts from './RelatedProducts'
import { urlFor } from '@utils/sanityImageBuilder'
import Image from 'next/image'
import { productData } from '@types'

```



```

const generateKey = () => `${Date.now()}-${Math.random().toString(36).substr(2, 9)}`;

type variation = {
  _key: string,
  variation_name: string,
  variation_options: string[]
}

const ProductDetail = ({ slug }: { slug: string }) => {
  const { products } = useStore()
  const [quantity, setQuantity] = useState(1)
  const [mainImage, setMainImage] = useState<string | null>(null)
  const [sideImages, setSideImages] = useState<string[]>([])
  const [currentVariation, setCurrentVariation] = useState<{ _key: string, variation_name:
string, variation_option: string }[]>([])
  const filteredProduct = products.filter((product: productData) => {
    return product.slug.current == slug
  })

  const handleCurrentVariation = (var_name: string, var_option: string) => {
    setCurrentVariation((prevVariations) =>
      prevVariations.map((variation) =>
        variation.variation_name === var_name
          ? { ...variation, variation_option: var_option }
          : variation
      )
    );
  }

  const handleMainImage = (selectedImage: string) => {
    if (mainImage !== null) {
      setSideImages((prevState: string[]) =>
        [...prevState.filter((image: string) => image !== selectedImage), mainImage]
      );
    }
    setMainImage(selectedImage);
  };

  console.log(filteredProduct[0]?.product_name)

  useEffect(() => {
    if (products.length > 0) {
      if (filteredProduct[0].variation_details != null) {
        setCurrentVariation((prevVariation) =>
          [...prevVariation,
...filteredProduct[0]?.variation_details?.map((variation: variation) => {
            return {
              _key: generateKey(),
              variation_name: variation.variation_name,
              variation_option: variation.variation_options[0]
            }
          )
        )
      }
    }
  }, [products, filteredProduct])
}

```



```

    }
  })]
)
}
const main_image = filteredProduct[0]?.main_image ?
urlFor(filteredProduct[0]?.main_image)?.url() : null;
setMainImage(main_image!)
const sideImages = filteredProduct[0]?.product_images?.map((product_image:
string) => {
  return urlFor(product_image)?.url()
})
setSideImages(sideImages)
}, [products])

return (
  <div>
    <div className='bg-[#F9F1E7] py-5'>
      <div className='flex flex-wrap items-center gap-x-8 w-[90%] 2xl:w-[1240px]
mx-auto'>
        <div className='flex items-center gap-x-5'>
          <p className='text-[#9F9F9F] text-[16px]'>Home</p>
          <svg width="8" height="14" viewBox="0 0 8 14" fill="none"
xmlns="http://www.w3.org/2000/svg">
            <path d="M0 12L5 7L0 2L1 0L8 7L1 14L0 12Z" fill="black" />
          </svg>
        </div>
        <div className='flex items-center gap-x-5'>
          <p className='text-[#9F9F9F] text-[16px]'>Shop</p>
          <svg width="8" height="14" viewBox="0 0 8 14" fill="none"
xmlns="http://www.w3.org/2000/svg">
            <path d="M0 12L5 7L0 2L1 0L8 7L1 14L0 12Z" fill="black" />
          </svg>
        </div>
        <svg width="2" height="37" viewBox="0 0 2 37" fill="none"
xmlns="http://www.w3.org/2000/svg">
          <line x1="1" x2="1" y2="37" stroke="#9F9F9F" strokeWidth="2" />
        </svg>
        <p className='text-[16px]'>{filteredProduct[0]?.product_name}</p>
      </div>
    </div>
    <div className='w-[90%] 2xl:w-[1240px] mx-auto'>
      <div className='flex max-lg:flex-col gap-x-5 py-8'>
        <div className='flex max-md:flex-col-reverse max-md:items-center max-
lg:justify-center gap-x-3 lg:w-[50%]'>
          <div className='max-md:flex max-sm:justify-center overflow-x-scroll
max-md:gap-5 max-md:mt-4'>
            {

```

```

        sideImages?.map((image, index) => {
            return (
                <Image onClick={() => handleMainImage(image)}
key={index} src={image} className='bg-gray-100 my-1' width={100} height={100} style={{ width:
"100px", height: "100px", objectFit: "contain" }} alt={filteredProduct[0]?.product_name} />
            )
        })
    }
</div>
<div className='lg:flex-1'>
    {
        mainImage && (
            <Image
                src={mainImage}
                width={500}
                height={500}
                style={{ width: "500px", height: "500px", objectFit:
"contain" }}
                alt={filteredProduct[0]?.product_name}
            />
        )
    }
</div>
</div>
<div className='w-[100%] lg:w-[50%]'>
    <h1 className='text-[42px]'>{filteredProduct[0]?.product_name}</h1>
    <p className='text-[24px] font-medium text-
[#9F9F9F]'>${filteredProduct[0]?.price}</p>
    <p className='text-[16px] leading-[19.5px] sm:w-[450px] py-
2'>{filteredProduct[0]?.description.slice(0, 250)}...
    </p>
    {
        filteredProduct[0]?.variation_details != null ?
        filteredProduct[0]?.variation_details.map((variation:
variation, index: number) => {
            return (
                <React.Fragment key={index}>
                    {
                        variation.variation_name == "Color" ?
                            <div className='py-3'>
                                <p className='text-[16px] text-
[#9F9F9F]'>{variation.variation_name}</p>
                                <div className='flex gap-x-4 mt-4'>
                                    {
                                        variation.variation_options.m
ap((options: string, index: number) => {
                                            const isActive =
currentVariation.some(({ variation_name, variation_option }) => {
                                                return variation_name
== variation.variation_name && variation_option == options

```

```

    })
    return <button
onClick={() => handleCurrentVariation(variation.variation_name, options)} key={index}
className={`w-[30px] h-[30px] rounded-full ${isActive ? "border-4 border-gray-600" : "border
border-gray-600"}`} style={{ backgroundColor: options }}></button>
    })
  }
</div>
</div>
:
<div className='py-3'>
  <p className='text-[16px] text-
[#9F9F9F]'>{variation.variation_name}</p>
  <div className='flex gap-x-2 mt-4'>
    {
      variation.variation_options.m
    }
  </div>
</div>
ap((options: string, index: number) => {
  const isActive =
currentVariation.some(({ variation_name, variation_option }) => {
  return variation_name
== variation.variation_name && variation_option == options
  })
  return options != "" ?
<button onClick={() => handleCurrentVariation(variation.variation_name, options)} key={index}
className={`rounded-lg ${isActive ? "bg-[#B88E2F] text-white" : "bg-gray-300 text-black"} px-
2 h-[30px]`}>{options}</button> : null
  })
  }
</div>
</div>
}
</React.Fragment>
)
}) : null
}

    <div className='flex flex-wrap max-sm:flex-col max-sm:items-start
gap-x-2 justify-between items-center py-3'>
      <div className='flex justify-around items-center w-[123px] h-
[64px] border border-[#9F9F9F] rounded-lg'>
        <span onClick={() => { if (quantity > 1) {
setQuantity(quantity - 1) } }} className='text-[16px] cursor-pointer'></span>
        <p className='text-[16px]'>{quantity}</p>
        <span className='text-[16px] cursor-pointer' onClick={() =>
setQuantity(quantity + 1)}></span>
      </div>
      <div className='flex-1 max-sm:mt-3'>

```

```

        <AddToCartButton productData={filteredProduct[0]}
variation={currentVariation} quantity={quantity} />
        <button className='text-[18px] max-sm:mr-2 max-sm:mt-2 sm:mx-
2 border border-black px-4 py-4 rounded-xl'>Comparison</button>
    </div>
</div>
<div className='my-10 h-[0.5px] bg-black/40'>
</div>
<div className=''>
    <div className='flex flex-wrap py-1'>
        <div className='flex justify-between w-[80px]'>
            <p className='text-[#9F9F9F] text-[16px]'>SKU</p>
            <span className='text-[#9F9F9F] text-[16px]'>:</span>
        </div>
        <p className='flex-1 ml-2 text-[#9F9F9F] text-
[16px]'>{filteredProduct[0]?._id.slice(0, 5)}</p>
    </div>
    <div className='flex flex-wrap py-1'>
        <div className="flex justify-between w-[80px]">
            <p className='text-[#9F9F9F] text-[16px]'>Category</p>
            <span className='text-[#9F9F9F] text-[16px]'>:</span>
        </div>
        <p className='flex-1 ml-2 text-[#9F9F9F] text-
[16px]'>{filteredProduct[0]?.category.category_name}</p>
    </div>
    <div className='flex flex-wrap py-1'>
        <div className='flex justify-between w-[80px]'>
            <p className='text-[#9F9F9F] text-[16px]'>Share</p>
            <span className='text-[#9F9F9F] text-[16px]'>:</span>
        </div>
        <div className='flex flex-1 ml-2 gap-x-3 items-center'>
            <svg width="20" height="20" viewBox="0 0 20 20"
fill="none" xmlns="http://www.w3.org/2000/svg">
                <g clipPath="url(#clip0_117_920)">
                    <path fillRule="evenodd" clipRule="evenodd" d="M0
10.0558C0 15.0275 3.61083 19.1617 8.33333 20V12.7775H5.83333V10H8.33333V7.7775C8.33333 5.2775
9.94417 3.88917 12.2225 3.88917C12.9442 3.88917 13.7225 4 14.4442
4.11083V6.66667H13.1667C11.9442 6.66667 11.6667 7.2775 11.6667 8.05583V10H14.3333L13.8892
12.7775H11.6667V20C16.3892 19.1617 20 15.0283 20 10.0558C20 4.525 15.5 0 10 0C4.5 0 0 4.525 0
10.0558Z" fill="black" />
                </g>
            </defs>
            <clipPath id="clip0_117_920">
                <rect width="20" height="20" fill="white" />
            </clipPath>
        </defs>
    </svg>
    <svg width="20" height="20" viewBox="0 0 20 20"
fill="none" xmlns="http://www.w3.org/2000/svg">
        <g clipPath="url(#clip0_117_924)">

```

```

<path fillRule="evenodd" clipRule="evenodd"
d="M0.833252 2.365C0.833252 1.95877 0.994624 1.56919 1.28187 1.28195C1.56911 0.994702 1.9587
0.83333 2.36492 0.83333H17.6333C17.8346 0.833001 18.034 0.872383 18.22 0.949219C18.4061
1.02606 18.5752 1.13884 18.7176 1.28111C18.8601 1.42338 18.973 1.59235 19.0501
1.77834C19.1271 1.96433 19.1667 2.16368 19.1666 2.365V17.6333C19.1668 17.8347 19.1273 18.0341
19.0504 18.2202C18.9735 18.4063 18.8606 18.5753 18.7183 18.7178C18.5759 18.8602 18.4069
18.9731 18.2209 19.0502C18.0348 19.1272 17.8354 19.1668 17.6341 19.1667H2.36492C2.16371
19.1667 1.96447 19.127 1.77858 19.05C1.5927 18.973 1.42381 18.8601 1.28157 18.7178C1.13933
18.5754 1.02653 18.4065 0.949604 18.2206C0.87268 18.0346 0.833143 17.8354 0.833252
17.6342V2.365ZM8.08992 7.82333H10.5724V9.07C10.9308 8.35333 11.8474 7.70833 13.2249
7.70833C15.8658 7.70833 16.4916 9.13583 16.4916 11.755V16.6067H13.8191V12.3517C13.8191 10.86
13.4608 10.0183 12.5508 10.0183C11.2883 10.0183 10.7633 10.9258 10.7633
12.3517V16.6067H8.08992V7.82333ZM3.50659
16.4925H6.17992V7.70833H3.50659V16.4917V16.4925ZM6.56242 4.84333C6.56746 5.07222 6.52673
5.29982 6.44262 5.51276C6.35851 5.7257 6.23271 5.91969 6.07261 6.08336C5.91251 6.24702
5.72133 6.37706 5.5103 6.46584C5.29926 6.55461 5.07262 6.60035 4.84367 6.60035C4.61472
6.60035 4.38808 6.55461 4.17704 6.46584C3.966 6.37706 3.77483 6.24702 3.61473 6.08336C3.45463
5.91969 3.32883 5.7257 3.24472 5.51276C3.16061 5.29982 3.11988 5.07222 3.12492
4.84333C3.13481 4.39404 3.32024 3.96649 3.64149 3.65224C3.96274 3.33798 4.39427 3.16201
4.84367 3.16201C5.29307 3.16201 5.7246 3.33798 6.04585 3.65224C6.3671 3.96649 6.55253 4.39404
6.56242 4.84333Z" fill="black" />
</g>
<defs>
<clipPath id="clip0_117_924">
<rect width="20" height="20" fill="white" />
</clipPath>
</defs>
</svg>
<svg width="25" height="25" viewBox="0 0 25 25"
fill="none" xmlns="http://www.w3.org/2000/svg">
<path d="M12.5 1.5625C6.45996 1.5625 1.5625 6.45996
1.5625 12.5C1.5625 18.54 6.45996 23.4375 12.5 23.4375C18.54 23.4375 23.4375 18.54 23.4375
12.5C23.4375 6.45996 18.54 1.5625 12.5 1.5625ZM17.7563 9.80713C17.7637 9.92188 17.7637
10.0415 17.7637 10.1587C17.7637 13.7427 15.0342 17.8711 10.0464 17.8711C8.5083 17.8711
7.08252 17.4243 5.88135 16.6553C6.10107 16.6797 6.31104 16.6895 6.53565 16.6895C7.80518
16.6895 8.97217 16.2598 9.90234 15.5322C8.71094 15.5078 7.70996 14.7266 7.36816
13.6523C7.78564 13.7134 8.16162 13.7134 8.59131 13.6035C7.97785 13.4789 7.42645 13.1457
7.0308 12.6606C6.63515 12.1755 6.41964 11.5684 6.4209 10.9424V10.9082C6.77979 11.1108 7.20215
11.2354 7.64404 11.2524C7.27256 11.0049 6.96792 10.6695 6.75711 10.276C6.5463 9.88244 6.43585
9.443 6.43555 8.99658C6.43555 8.49121 6.56738 8.02979 6.8042 7.62939C7.48511 8.46762 8.33479
9.15318 9.29801 9.64152C10.2612 10.1299 11.3164 10.41 12.395 10.4639C12.0117 8.62061 13.3887
7.12891 15.0439 7.12891C15.8252 7.12891 16.5283 7.45605 17.0239 7.9834C17.6367 7.86865
18.2227 7.63916 18.7451 7.33154C18.5425 7.95898 18.1177 8.48877 17.5537 8.82324C18.1006
8.76465 18.6279 8.61328 19.1162 8.40088C18.7476 8.94287 18.2861 9.42383 17.7563 9.80713Z"
fill="black" />
</svg>
</div>
</div>
</div>
</div>

```

```

    </div>

    <div className='my-8'>
      <div className='flex flex-wrap gap-x-10 justify-center'>
        <h2 className='text-[24px] font-medium text-center'>Description</h2>
        <h2 className='text-[24px] font-medium text-center text-
[#9F9F9F]'>Additional Information</h2>
        <h2 className='text-[24px] font-medium text-center text-
[#9F9F9F]'>Reviews [5]</h2>
      </div>
      <div className='w-[80%] mx-auto space-y-5 py-6'>
        <p className='text-[16px] text-
[#9F9F9F]'>{filteredProduct[0]?.description}</p>
      </div>
    </div>
    <RelatedProducts />
  </div>
</div>
)
}

export default ProductDetail

```