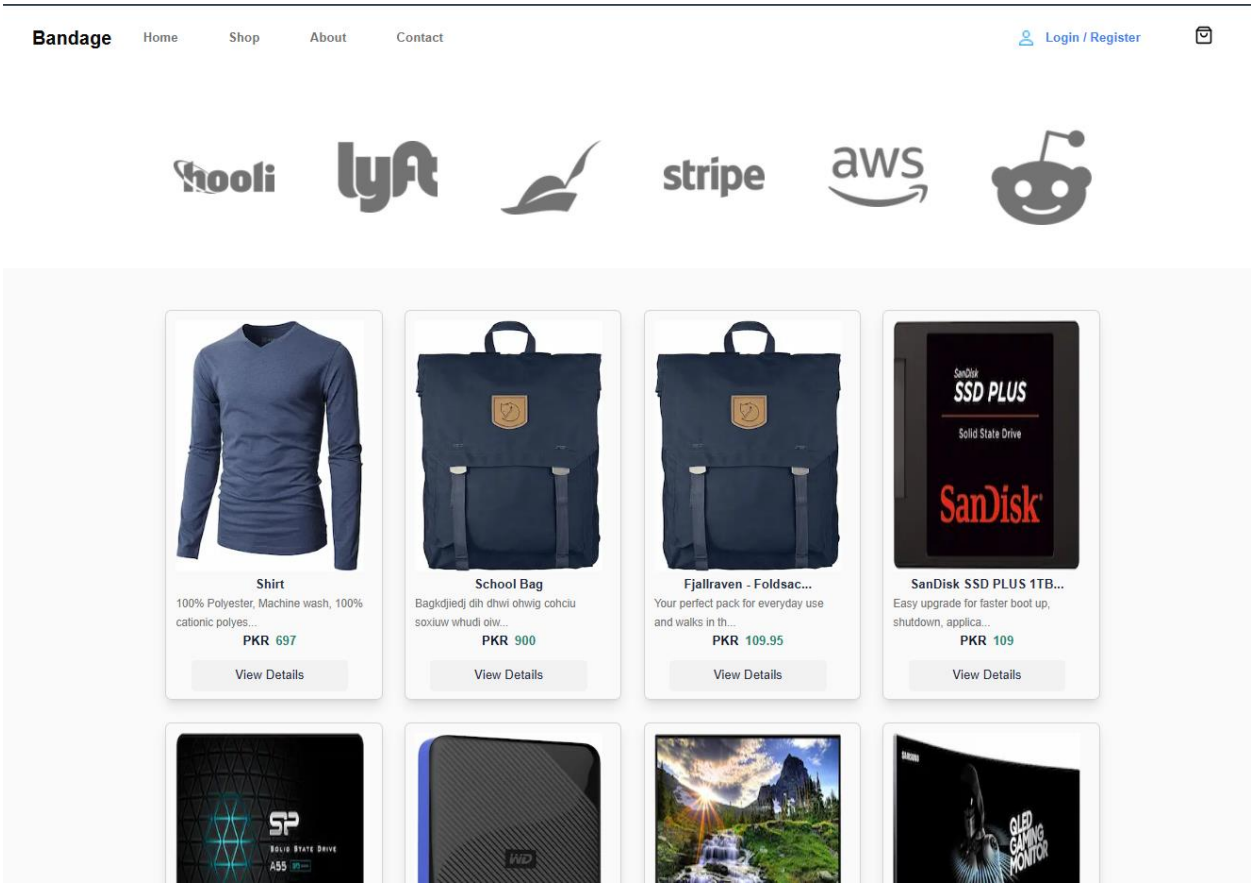
Dynamic Frontend Components [Brands Bazar]

On the Day 4 of marketplace builder hackathon I have built frontend components that were needed according to my project needs. So far I have built:

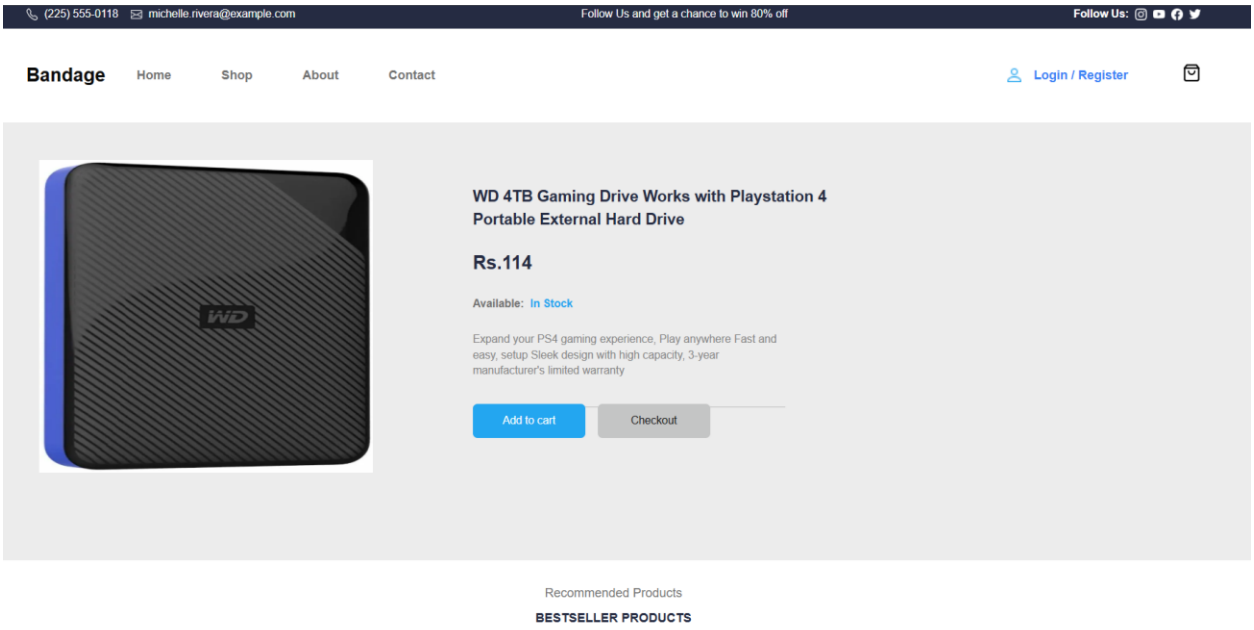
Product Listing:

After fetching data dynamically from our sanity.io on day3 of our hackathon, I made a proper listing of products. These are completely dynamic and functional so the user gets the best user experience.



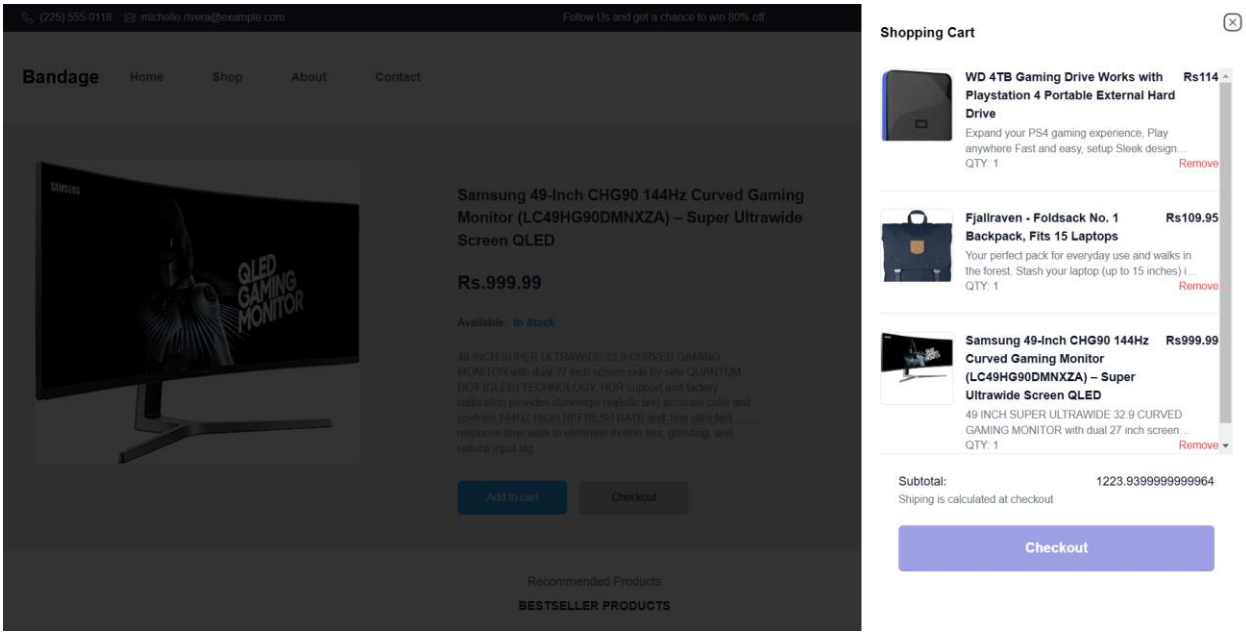
Product Detail or Slug page:

I created a slug page to display every product's detail dynamically. The slug page gets data on the basis of the generated slug of each product and then displays that data on the UI.



Add to cart:

I created the UI for the add to cart page using "**shadcn**". I used the slider component of shadcn to the UI. For the functionality of the cart I have used an external library "**use shopping cart**". It made the process quite easy and more time is saved. I have also made a Sub-total calculator which again was made using some built-in components of the use shopping cart library. In the cart you can see the quantity counter and a remove button which are some essential functions for the cart.



Code deliverables of components:

Product Listing:

featuredproduct.tsx M

src > app > components > featuredproduct.tsx > BestSeller > products.map() callback

```tsx
import React from "react";
import { client } from "../../sanity/lib/client";
import Image from "next/image";
import Link from "next/link";
function truncateText(text: string, maxLength: number) {
  if (text.length > maxLength) {
    return text.slice(0, maxLength) + "...";
  }
  return text;
}

const getFtProducts = async () => {
  const product = await client.fetch(`
    *[_type == "product"][0..7]{
  _id,
    name,
    description,
    price,
    "currentslug" : slug.current,
    "image_url": image.asset->url,
  }
    `);
  return product;
};
export default async function BestSeller() {
  const products = await getFtProducts();
  return (
    <div className="flex py-7 flex-col items-center">
      <h1 className="leading-[30px] text-[#737373]">Recommended Products</h1>
      <h2 className="leading-[32px] font-bold text-[#252B42] ">
        BESTSELLER PRODUCTS
      </h2>


      <div className="pt-6 gap-7 grid grid-cols-1 sm:grid-cols-2 lg:grid-cols-3 xl:grid-cols-4">
        {products.map(
          (product: {
            id: string;
            name: string;
            price: number;
            image_url: string;
            description: string;
            currentslug: string;
          }) => (
            <div key={product.id} className="rounded-lg border-[1.5px] border-gray-300 shadow-lg p-3">

            <div className="w-[250px]  h-[470px]  " >
              <Image
                classNam="h-80 "
                alt={product.name}
                src={product.image_url}
                width={239}
                height={420}
                />
              <div className="w-[239px] h-[188px] flex flex-col pt-1 items-center  ">
                <h1 className="font-bold text-[16px] text-[#252B42] leading-[24px]">
                  {truncateText(product.name, 20)}
                </h1>
                <p className="text-[14px] leading-[24px] font-bold text-[#737373]">
                  {truncateText(product.description, 50)}
                </p>
                <div className="flex gap-1 items-center">
                  <p className="font-bold text-[16px] flex gap-2 text-[#23856D] ">

                      <span className="font-bold text-[16px] text-slate-900">PKR</span>
                      {product.price}
                  </p>
                </div>
                <div className="py-3 flex gap-2">

                  <Link href={`/product/${product.currentslug}`}>
                  <button classNam="bg-[#F2F2F2] hover:bg-[#dbdada] active:bg-[#dbdada] font-medium text-[#252B42] w-[200px] h-[40px] rounded-md">
                    View Details

                  </button>
                  </Link>
                </div>
              </div>
            </div>
              </div>
          )
        )}
      </div>
    </div>
  );
}
```

Slug:

```
export default function Product({ params }: { params: { slug: string } }) {
  const [product, setProduct] = useState<Product[]>([]);
  const [bestSellerComponent, setBestSellerComponent] =
    useState<JSX.Element | null>(null);

  // Fetch product data based on slug
  useEffect(() => {
    async function fetchData() {
      const res = await getData(params.slug);
      setProduct(res || []);
    }
    fetchData();
  }, [params.slug]);

  // Load BestSeller component dynamically
  useEffect(() => {
    const fetchBestSeller = async () => {
      const component = await BestSeller();
      setBestSellerComponent(component);
    };
```

```
<div>
  {product.length > 0 ? (
    product.map((products) => (
      <div
        key={products._id}
        className="w-[100vw] flex flex-col lg:flex-row pl-3  gap-56 bg-[#ECECEC] h-auto py-20"
      >
        {/* Product Image */}
        <div className="relative md:left-24 -top-8 ">
          <Image
            className="lg:w-[700px] lg:h-[400px] md:w-[400px] md:h-[300px] w-[300px] h-[300px]"
            src={products.image_url}
            alt={products.name}
            width={500}
            height={400}
          />

        </div>

        {/* Product Details */}
        <div className=" flex flex-col  w-[100vw]  gap-6  items-center lg:items-start  lg:-top-0 relative -top-28  ">
          <p className="text-[20px]  md:w-[30rem] font-semibold leading-[30px] text-[#252B42]">
            {products.name}
          </p>
          <div className="font-bold w-[105px] text-[#252B42] text-[24px] leading-8">
            Rs.{products.price}
          </div>
          <div className="flex gap-2 w-40 font-bold text-[14px] leading-6">
            <p className="text-[#737373]">Available:</p>
            <p className="text-[#23A6F0]">In Stock</p>
          </div>
          <div className="w-[270px] md:w-[400px] h-[60px] text-[#858585] pb-24 border-b-[1px] border-[#BDBDBD] text-[14px] leading-[20px]">
            {products.description}
            <div className="flex  md:flex-row items-center gap-4">
              <AddToCart currency="PKR" description={products.description} image={products.image_url} name={products.name} price={products.price}
              quantity={products.quantity?.toString() || ''} key={products._id} id={products._id}/>

              {/* <button className="w-36 h-11 mt-16 sm:mt-8 rounded-md hover:bg-blue-200 hover:text-white bg-[#23A6F0] text-white">
                Add to cart
              </button> */}
              <button className="w-36 hover:bg-gray-300 h-11 mt-16 sm:mt-8 rounded-md bg-[#c4c5c5] text-[#1a1a1a]">
                Checkout
              </button>
            </div>
```

Add to Cart:

If you want to view the code for add to cart I would highly recommend you to visit code in my Github repo as it is divided in multiple files, the link to my github repo is: https://github.com/MustafaYamin/Marketplace-Builder-Hackathon-Brands-Bazar

Best Practices followed:

Day 4 of marketplace builder follow the practices of making reusable components and dividing my code in smaller chunks. It makes debugging very easy and does not modifies your whole code if you want to change some functionality or UI.

Challenges faced:

I did face some challenges in making the functionality for cart but outsourcing complex login to use-shopping-cart library and the documentations came very handy. I am still figuring out that how can I implement payment functionality in the project.