HACKATHON

Day 04

The goal of Day 04 is to design and develop our marketplace. Here, we have to fetch data from Sanity and display it in our marketplace. The best practice to develop a marketplace is to set it up in a modular form and create reusable components.

Key Section to Point Out:

Product Listing Component:

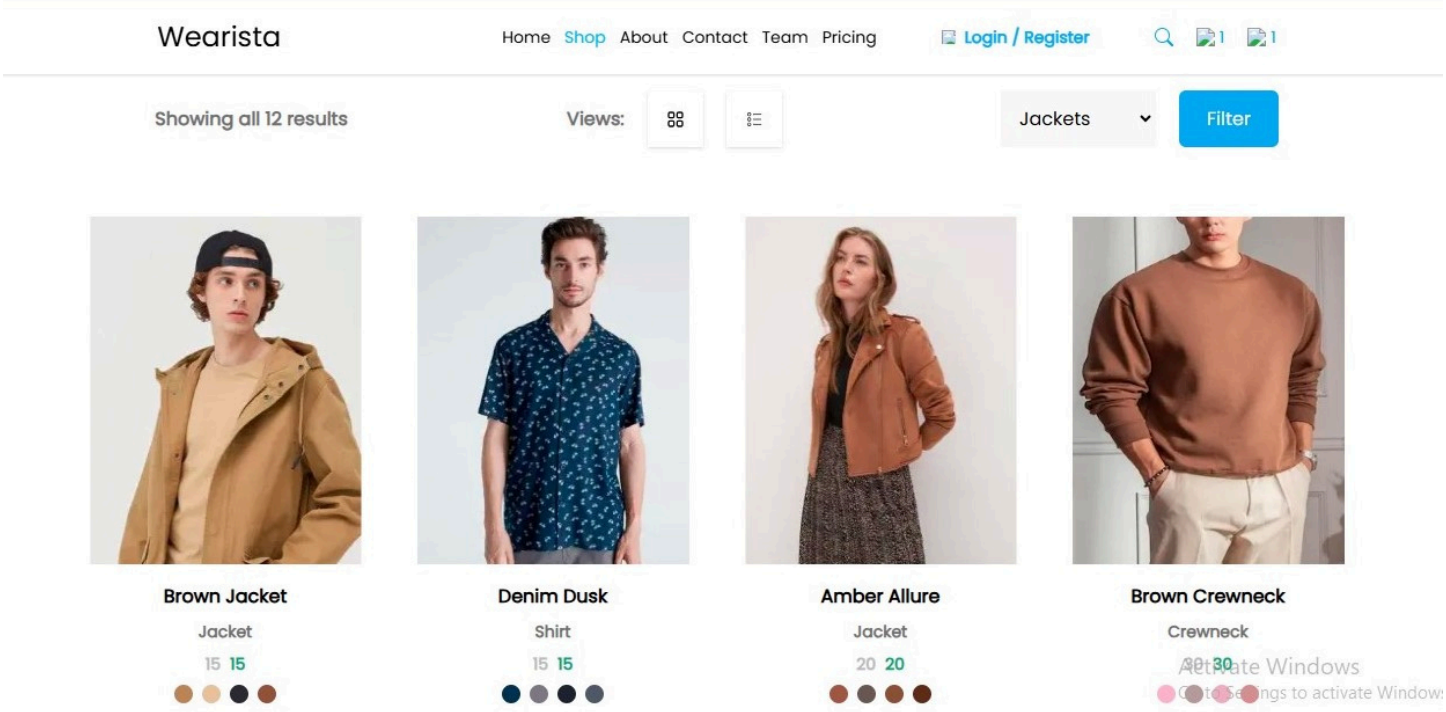
The backbone of every e-commerce marketplace is the product page. Here, I've listed the products using Flexbox. Each product has properties like quantity, price, discount, image, name, colors, sizes, and stock. I've displayed the products as cards showing the product image, name, and available colors.

I've also included:

- A search bar
- Filter operations via category
- Pagination for navigating through products

Challenges Faced:

The main challenges I faced were in developing the search bar and pagination. For pagination, I fetched data using queries like page and limit. For the search bar and filters, I fetched data without queries to perform the required operations.

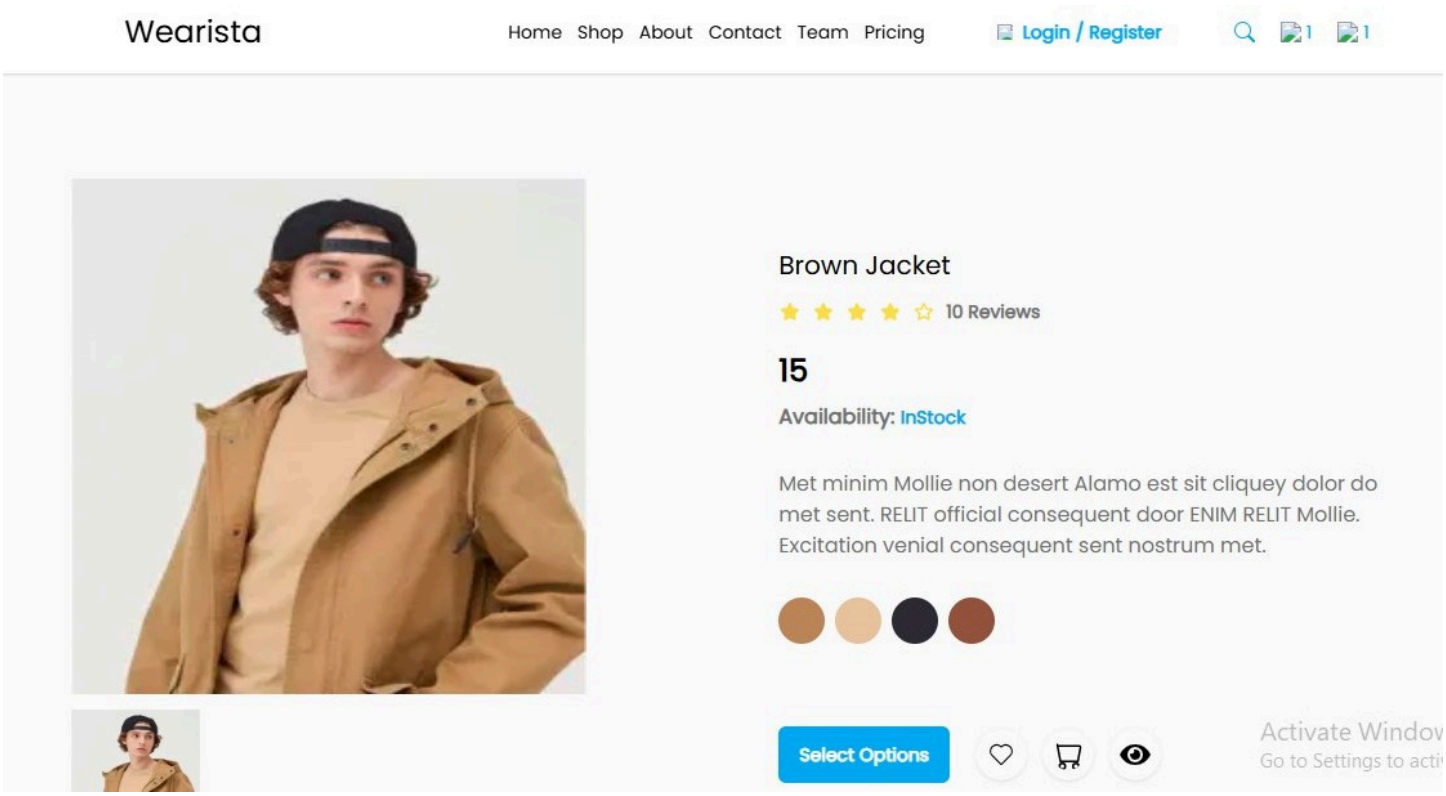


Product Detail Page:

The next task after developing the product page is to create the Product Detail Page, where the complete details of a product are displayed.

I've also added features for users like:

- Product suggestions
- An option to buy the product



Cart Page:

once the user completes the process to buy a product on the Product Detail Page, it's our responsibility to redirect the user to the Cart Page. Here, the user can see the list of products they want to buy

Wishlist Page:

You might be thinking, "What is a Wishlist page, and why do we need it?" Let me explain with a simple example:

Suppose you go to a supermarket and find a product you like, but for some reason, you don buy it. To remember it for next time, you either memorize the name or take a picture of it.

The Wishlist Page works the same way. If you like a product on our website but don't want to buy it right now, you can save it to your Wishlist.

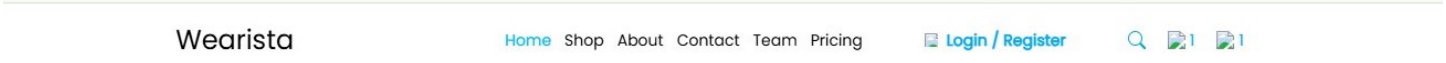
Next time you visit the website, you can simply go to the Wishlist Page, and the product you saved will be right there, ready for you to buy.

Header Section:

The Header is one of the most important parts of any web application.

With the help of the header, users can easily navigate between different pages like

- Home Page
- Product Page
- Cart Page
- Wishlist Page
- Login/Signup Page

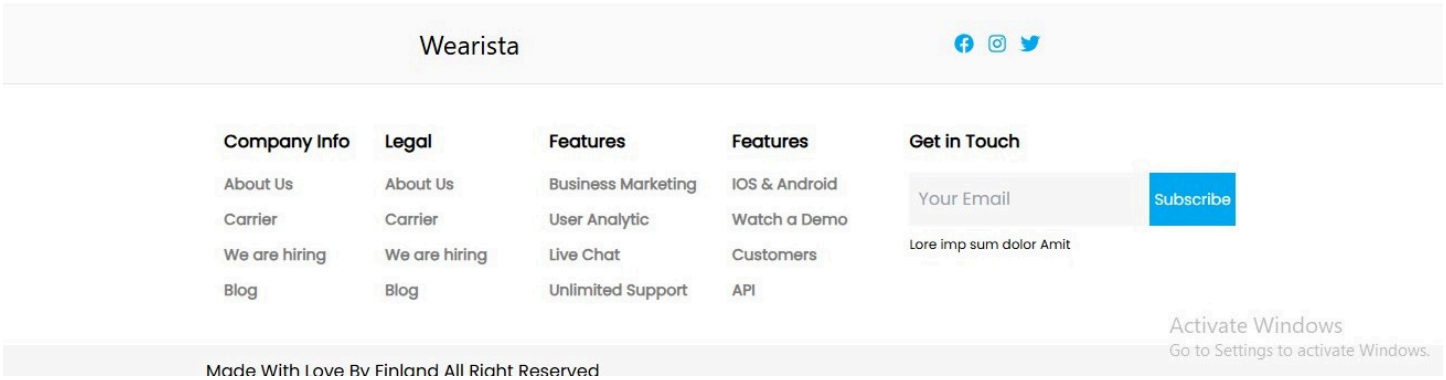


Footer Section:

The Footer is also an important part of a web application, just like the header.

In the footer, we display additional details such as:

- Subscription options
- Timings
- Address
- And many other useful details



Checkout Page:

The next important task after adding products to the cart is to set up the Checkout Page.

Here, users can pay for the products they want to buy. The checkout process includes all the necessary details like:

- Email
- Account Number
- Username

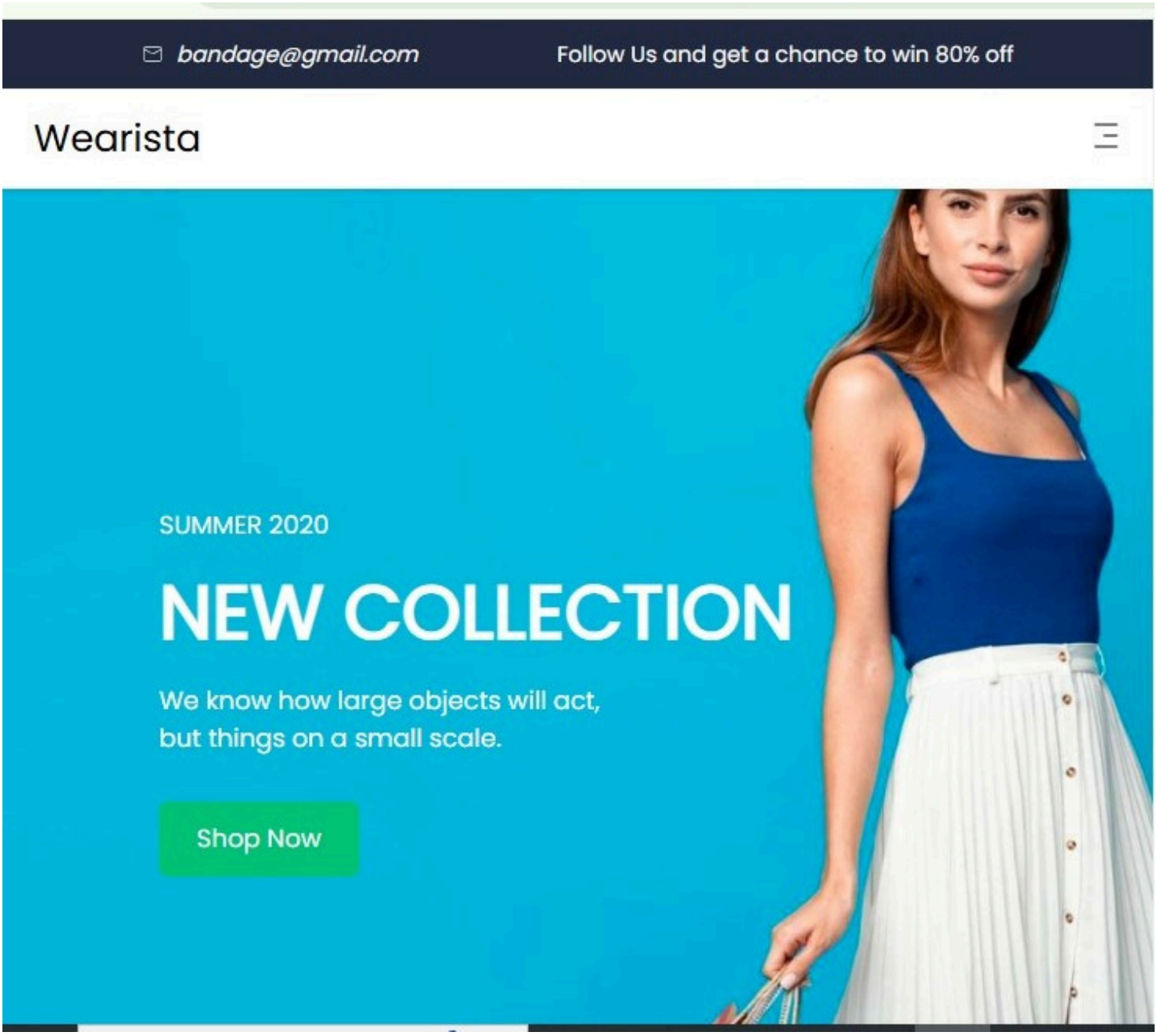
Successfully Purchase Page:

This is also an important task. After the user completes the checkout process, we show a confirmation page with a user-friendly message to confirm the successful purchase

Responsiveness:

The most important task is to make sure our website works perfectly on multiple screen sizes like:

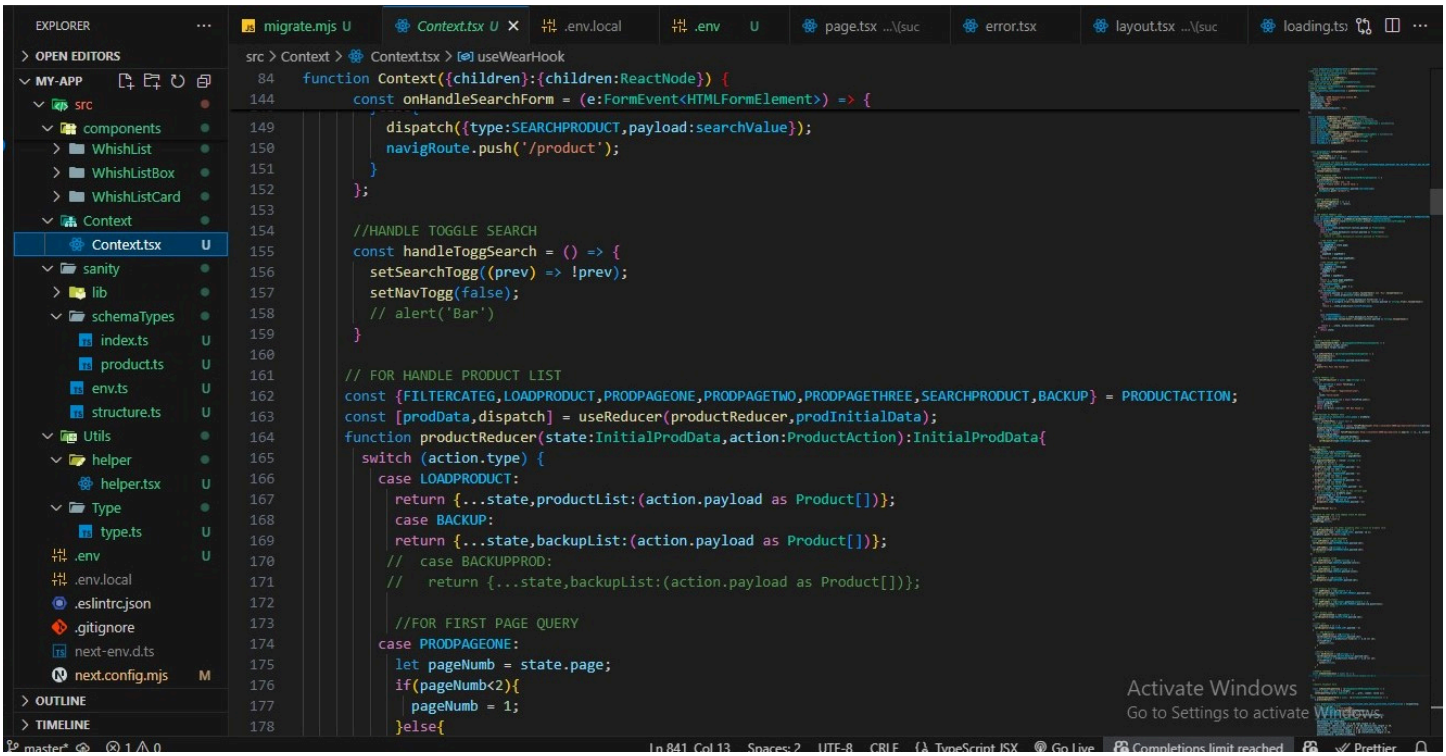
- Desktop
- Mobile
- Tablet
- TV



Challenges Faced:

While developing my marketplace, I faced several challenges like

- Creating API routes
- Fetching data
- Setting up functionalities
- Handling responsiveness


EXPLORER
migrate.mjs U
Context.tsx U
.env.local
.env U
page.tsx ...\(suc
error.tsx
layout.tsx ...\(suc
loading.ts
src > Context > Context.tsx > useWearHook
OPEN EDITORS
MY-APP
src
components
WhishList
WhishListBox
WhishListCard
Context
Context.tsx U
sanity
lib
schemaTypes
index.ts U
product.ts U
env.ts U
structure.ts U
Utils
helper
helper.tsx U
Type
type.ts U
.env U
.env.local
.eslintrc.json
.gitignore
next-env.d.ts
next.config.mjs M
OUTLINE
TIMELINE
84 function Context({children}:{children:ReactNode}) {
144 const onHandleSearchForm = (e:FormEvent<HTMLFormElement>) => {
149 dispatch({type:SEARCHPRODUCT,payload:searchValue});
150 navigRoute.push('/product');
151 }
152 };
153
154 //HANDLE TOGGLE SEARCH
155 const handleToggSearch = () => {
156 setSearchTogg((prev) => !prev);
157 setNavTogg(false);
158 // alert('Bar')
159 }
160
161 // FOR HANDLE PRODUCT LIST
162 const {FILTERCATEG,LOADPRODUCT,PRODPAGEONE,PRODPAGETWO,PRODPAGETHREE,SEARCHPRODUCT,BACKUP} = PRODUCTACTION;
163 const [prodData,dispatch] = useReducer(productReducer,prodInitialData);
164 function productReducer(state:InitialProdData,action:ProductAction):InitialProdData{
165 switch (action.type) {
166 case LOADPRODUCT:
167 return {...state,productList:(action.payload as Product[])};
168 case BACKUP:
169 return {...state,backupList:(action.payload as Product[])};
170 // case BACKUPPROD:
171 // return {...state,backupList:(action.payload as Product[])};
172
173 //FOR FIRST PAGE QUERY
174 case PRODPAGEONE:
175 let pageNumb = state.page;
176 if(pageNumb<2){
177 pageNumb = 1;
178 }else{
Activate Windows
Go to Settings to activate Windows.
master*
Ln 841, Col 13
Spaces: 2
UTF-8
CRLF
TypeScript JSX
Go Live
Completions limit reached
Prettier