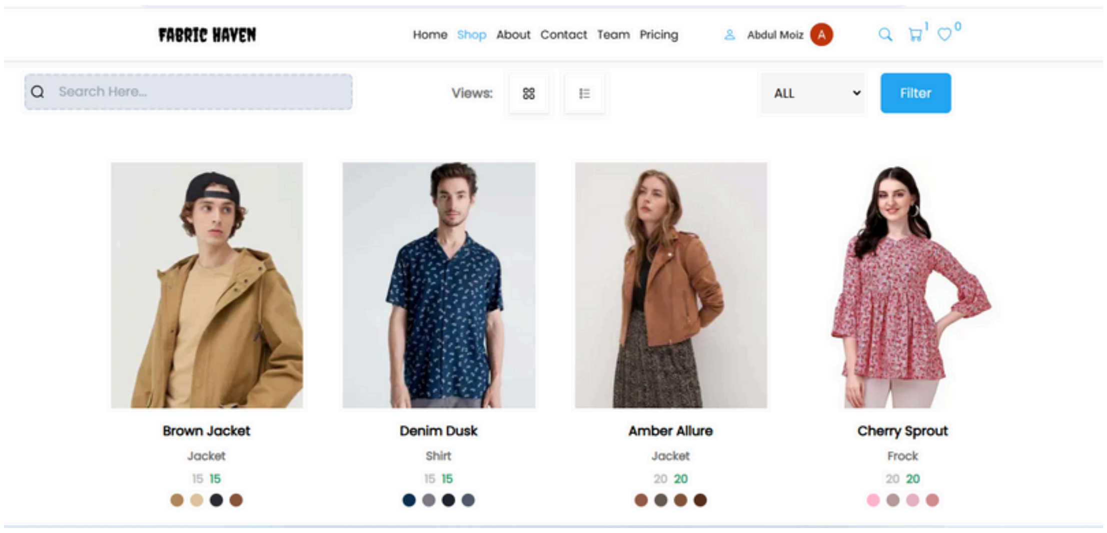
HACKATHON
DAY 05

The Goal of Day 05 is to setup our marketplace through testing component and many other thing, handling errors, check performance for optimization and then ready to handle for real-time traffic enviroment and aim to deliever best User Experience.

Key Section to Point Out

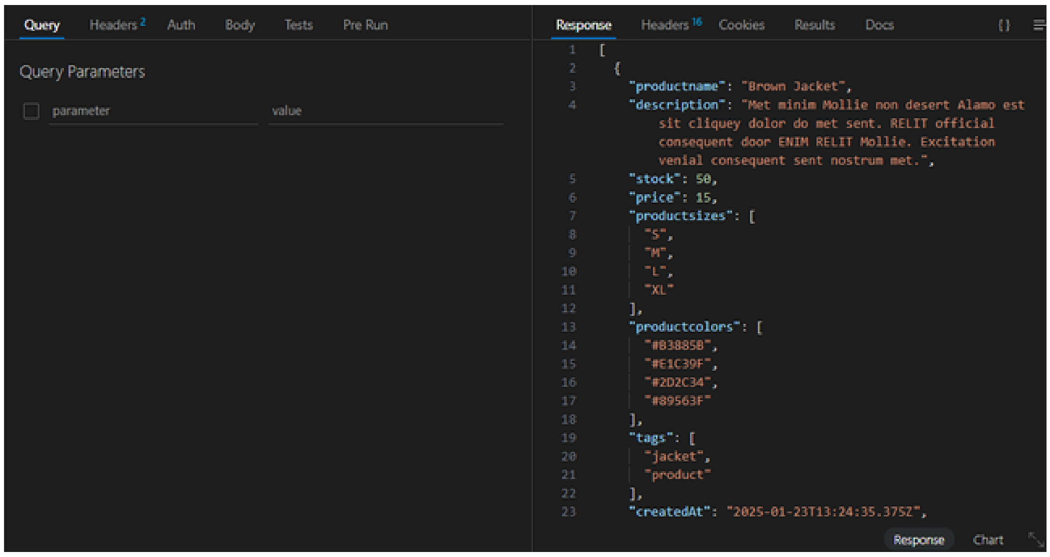
Functional Testing :

Here the first we must be do in E-commerce is to set our product page because that page is the backbone of every e-commerce web-application. The Product of that company must be display perfectly and here I have also included a search bar and also Filter box to Filter-out our products through category. We have to also perform Cart Operation and also show a detail page of our product. Below I have mention a screenshot of what I done.



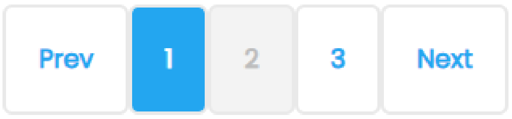
Next Task is to do is testing our API

We need to test our API because the data of all products is come from API means (Application Programming Interface). We can test that API through Thunder Client and many other testing platforms.



```
[
  {
    "productname": "Brown Jacket",
    "description": "Met minim Mollie non desert Alamo est
        sit cliquey dolor do met sent. RELIT official
        consequent door ENIM RELIT Mollie. Excitation
        venial consequent sent nostrum met.",
    "stock": 50,
    "price": 15,
    "productsizes": [
      "S",
      "M",
      "L",
      "XL"
    ],
    "productcolors": [
      "#B3885B",
      "#E1C39F",
      "#2D2C34",
      "#89563F"
    ],
    "tags": [
      "jacket",
      "product"
    ],
    "createdAt": "2025-01-23T13:24:35.375Z",
```

I Forget Something !

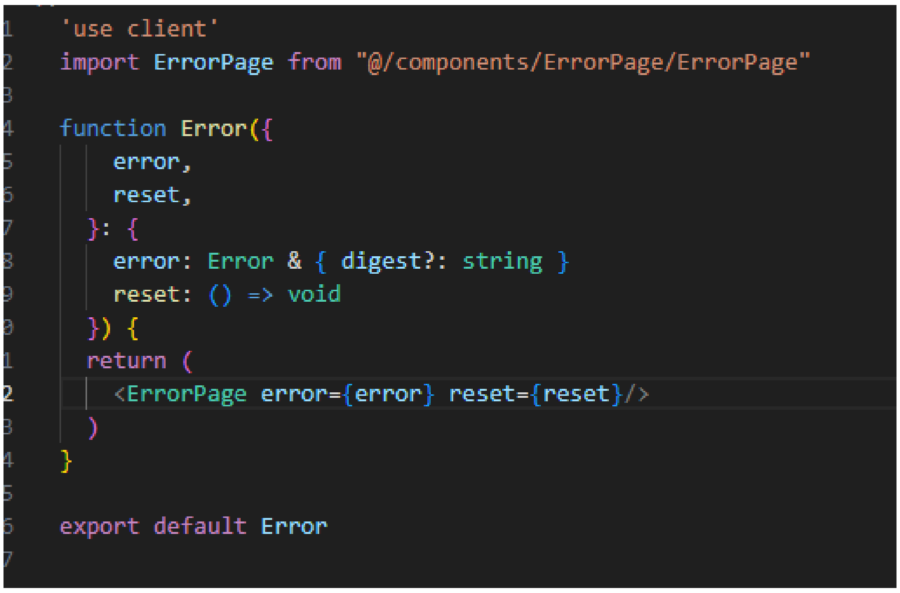


And that is Pagination , I have also include Pagination.

Error Handling :

When we develop something so the error will come on the board and then we have to solve it. Here is no path where error is not include. When we perform anything in the code and make some mistakes so the error will come and then we solve it the error will gone, But the Error Handling is not like that . It is like that , I am handling any API request and then error will come so then what fallback message that user can easily read and understand that what is this. In Nextjs, we many thing to handle the error like we have error.tsx file for if the error is come so then the error page will show with UI with user friendly content that user can understand.

Error.tsx

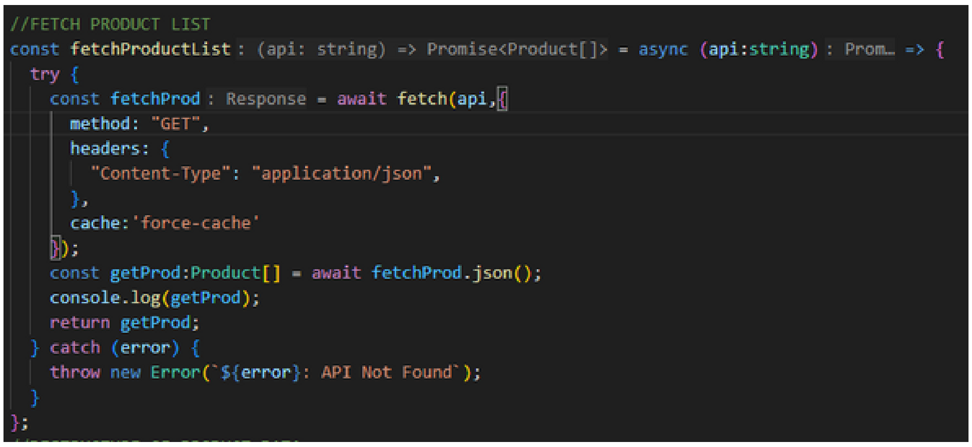


```tsx
'use client'
import ErrorPage from "@/components/ErrorPage/ErrorPage"

function Error({
    error,
    reset,
  }: {
    error: Error & { digest?: string }
    reset: () => void
  }) {
  return (
    <ErrorPage error={error} reset={reset}/>
  )
}

export default Error
```

API Request

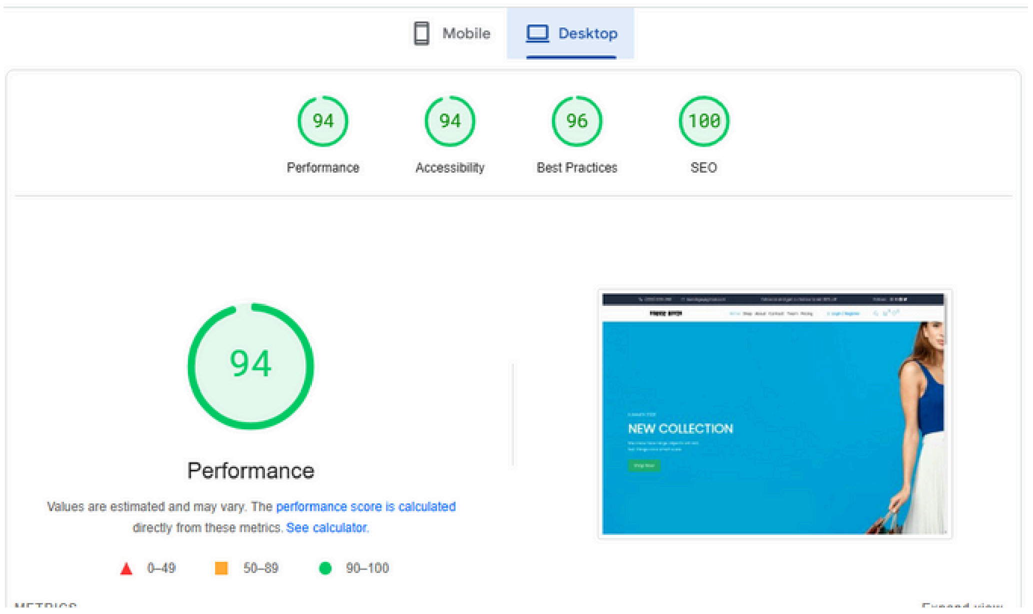


```ts
//FETCH PRODUCT LIST
const fetchProductList : (api: string) => Promise<Product[]> = async (api:string) : Prom… => {
  try {
    const fetchProd : Response = await fetch(api,{
      method: "GET",
      headers: {
        "Content-Type": "application/json",
      },
      cache:'force-cache'
    });
    const getProd:Product[] = await fetchProd.json();
    console.log(getProd);
    return getProd;
  } catch (error) {
    throw new Error(`${error}: API Not Found`);
  }
};
```

Performance Optimization :

Just Think like a CEO , An Employee of our company is doing his job , so as a CEO we have to know what the performance of that employee , so basically the performance of website is contain a matter. Let me show you desktop and mobile performance

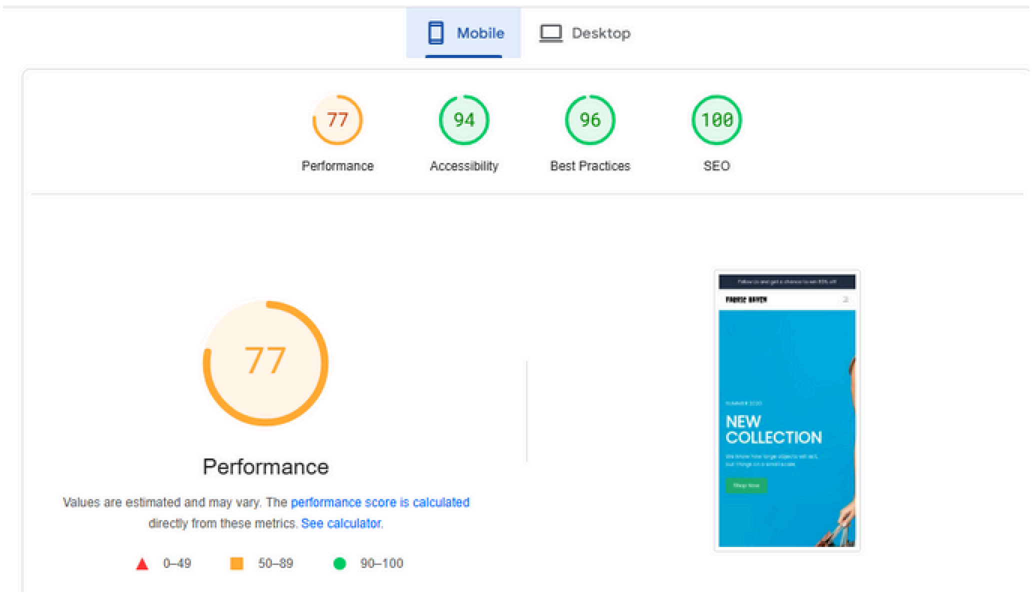
DESKTOP



Values are estimated and may vary. The performance score is calculated directly from these metrics. See calculator.

▲ 0–49 ■ 50–89 ● 90–100

MOBILE



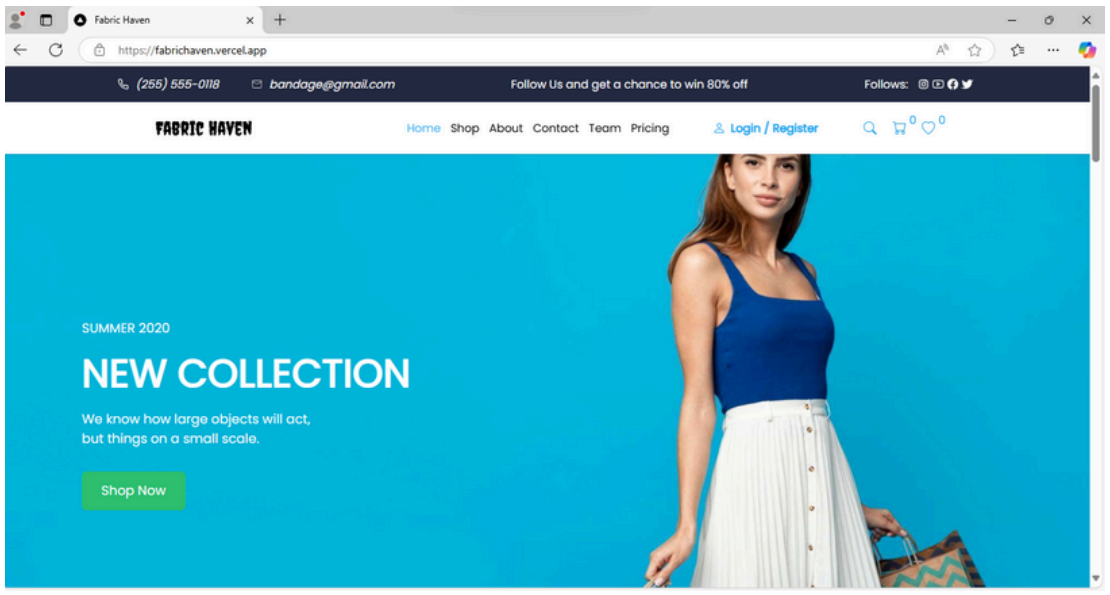
Values are estimated and may vary. The performance score is calculated directly from these metrics. See calculator.

▲ 0–49 ■ 50–89 ● 90–100

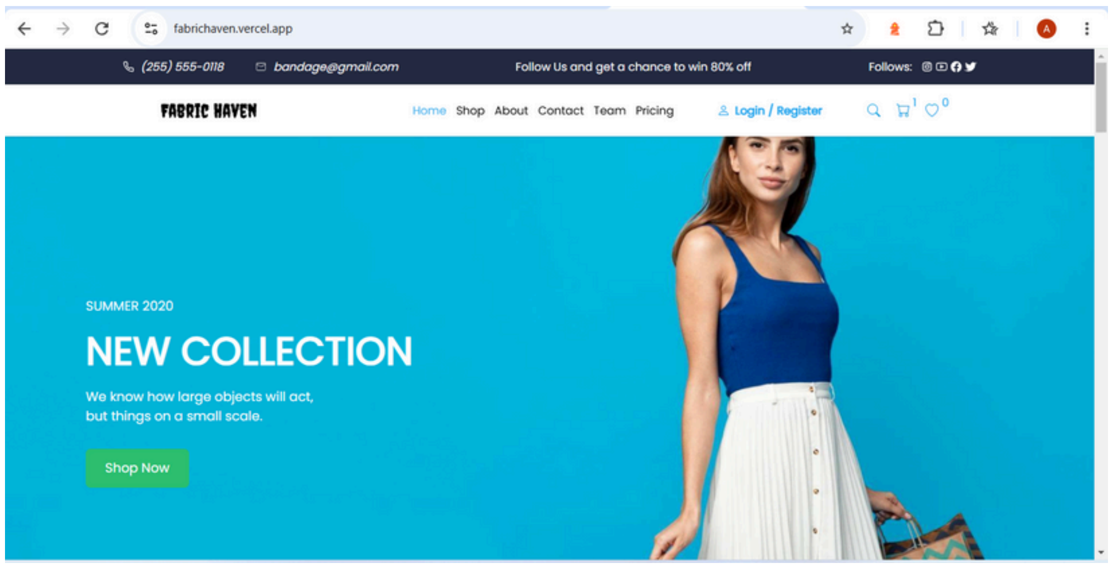
Cross-Browser and Device Testing :

When we developed a website. We need configure that our web-application must be working on cross browser like Chrome, Firefox, Safari, Edge etc. we have to test on Edge, Chrome.

MICROSOFT EDGE



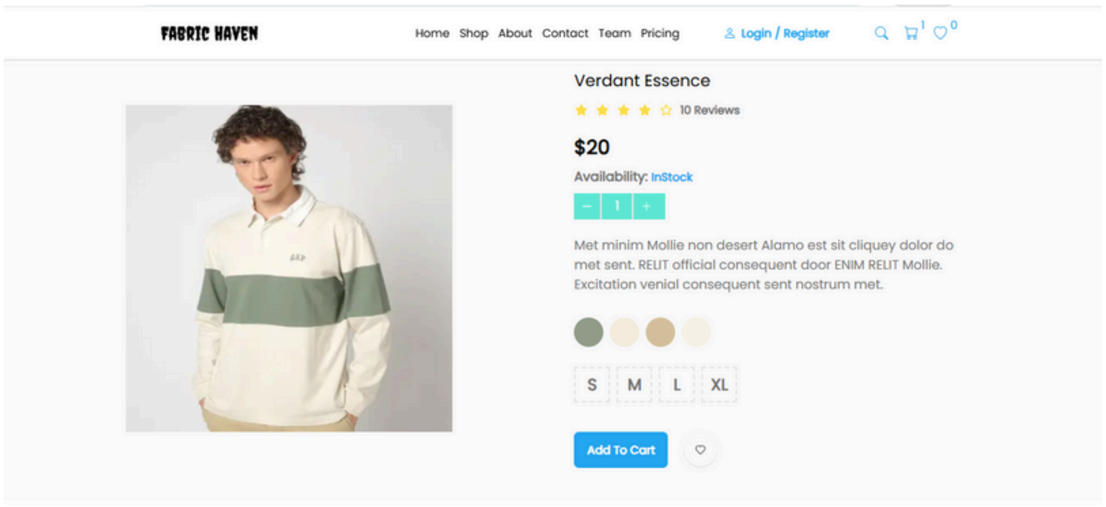
GOOGLE CHROME



That's how it's showing on Microsoft Edge and Chrome , Perfectly.

User Acceptance Testing (UAT) :

The main thing of every product is to check is yourself then go to another to check it. Once we developed a web-application so then we have to test it many time that is working perfectly or not, because when a product is build it’s test many time. The web-application is also a product and it’s our jod to check it many time.



Met minim Mollie non desert Alamo est sit cliquey dolor do met sent. RELIT official consequent door ENIM RELIT Mollie. Excitation venial consequent sent nostrum met.

Challenges Facing :

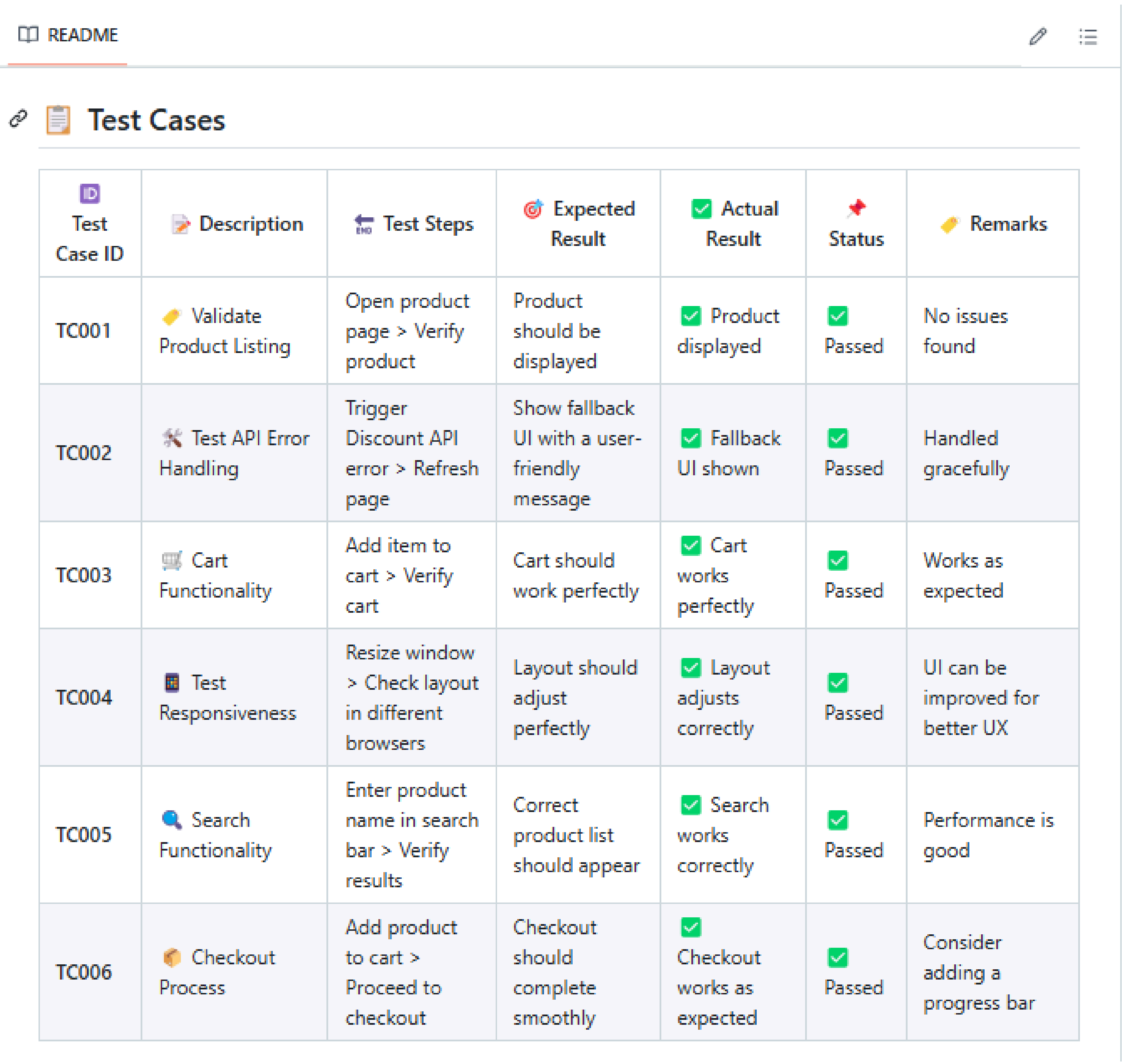
Facing Challenges in Program is a daily routine of a developer and here, I have also faced many challenges to setup search, functionality, filter and also on pagination and also on Error Handling and many other thing like Keys, Responsiveness, migration.

For Future :

As you know marketplace have many features what else features can I add for the Future. Here are many like connect with database, Multi Language Support and many other things.

Testing Report (CSV) :

After That, we need to create a Professional documentation where is the many detail of web-application



README

📋 Test Cases

🆔 Test Case ID	📝 Description	🔚 Test Steps	🎯 Expected Result	✅ Actual Result	📌 Status	🏷️ Remarks
TC001	🏷️ Validate Product Listing	Open product page > Verify product	Product should be displayed	✅ Product displayed	✅ Passed	No issues found
TC002	🛠️ Test API Error Handling	Trigger Discount API error > Refresh page	Show fallback UI with a user-friendly message	✅ Fallback UI shown	✅ Passed	Handled gracefully
TC003	🛒 Cart Functionality	Add item to cart > Verify cart	Cart should work perfectly	✅ Cart works perfectly	✅ Passed	Works as expected
TC004	📱 Test Responsiveness	Resize window > Check layout in different browsers	Layout should adjust perfectly	✅ Layout adjusts correctly	✅ Passed	UI can be improved for better UX
TC005	🔍 Search Functionality	Enter product name in search bar > Verify results	Correct product list should appear	✅ Search works correctly	✅ Passed	Performance is good
TC006	📦 Checkout Process	Add product to cart > Proceed to checkout	Checkout should complete smoothly	✅ Checkout works as expected	✅ Passed	Consider adding a progress bar