

*Day-5*

# **Testing and Backend Refinement - Comforty**

## Testing Report:

Test Case ID	Test Case Description	Expected Result	Actual Result	Status	Severity Level	Remarks
TC001	Validate product listing page	Products should load dynamically without errors.	Products displayed correctly.	Passed	Low	Fully functional.
TC002	Validate product detail page	Product detail page should load with accurate details.	Product detail page should load with accurate details.	Passed	Low	Works seamlessly.
TC003	Check category functionality	Relevant products should appear in the selected category.	Relevant products should appear in the selected category.	Passed	Low	Functional with no issues.
TC004	Test search bar	Dropdown with results or "No products found" message.	Dropdown with results or "No products found" message.	Passed	Low	No issues observed.
TC005	Validate cart functionality	Updates cart details correctly and prevents invalid actions.	Updates cart details correctly and prevents invalid actions.	Passed	Medium	Fully functional.
TC006	Validate reviews component	Reviews should be saved and visible for respective products.	Works as expected.	Passed	Medium	Fully functional.
TC007	Test footer and header links	Navigation should be smooth and consistent across pages.	Navigation links function correctly.	Passed	Low	Works seamlessly.

TC008	Validate notifications	Notifications should appear correctly for each action.	Notifications display as expected.	<b>Passed</b>	Medium	No issues.
TC009	Validate contact form	Valid forms should submit, and invalid ones should alert.	Works as expected with accurate validation.	<b>Passed</b>	Low	Validation is robust.
TC010	Validate pagination	Products should load smoothly when switching pages.	Pagination works as intended.	<b>Passed</b>	Low	Fully functional.

# Error Handling

## 1. Add Error Messages

Code Snippet:

```
useEffect(() => {
  async function fetchProductData() {
    setIsLoading(true);
    try {
      const productData = await GetProductData();
      setProductData(productData);
    } catch (error) {
      console.error("Error fetching product data:", error);
    } finally {
      setIsLoading(false);
    }
  }

  fetchProductData();
}, []);
```

Code Snippet:

```
useEffect(() => {
  async function fetchCategoryData() {
    setIsLoading(true);
    try {
      const categoryData: ProductInterface[] = await GetCategoriesData();
      setCategoryData(categoryData);
    } catch (error) {
      console.error("Error fetching category data:", error);
    } finally {
      setIsLoading(false);
    }
  }

  fetchCategoryData();
}, []);
```

## 2. Fallback UI

### Code Snippet:

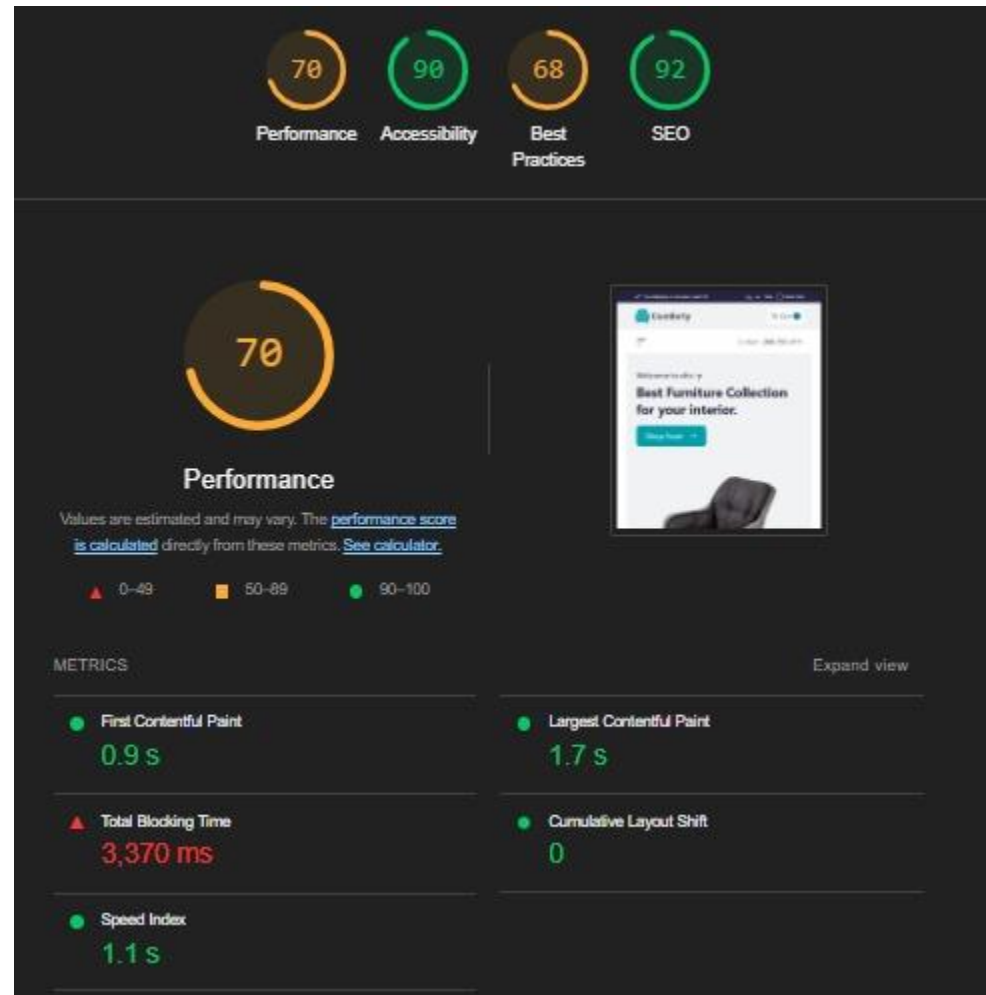
```
<div>{isLoading ? (  
  <p className="md:text-xl py-5">Loading...</p>  
) : (  
  <p>Product not found!!!</p>  
);  
}
```

### Code Snippet

```
> return productData ? (...  
  ) : (  
    <p>Product not found!!!</p>  
  );  
}
```

## Text Website through Lighthouse:

### 1. Mobile:



## 2. Desktop:

