

Day 4 - Dynamic Frontend Components of - E - Commerce Website:

Functional Deliverables:

Product Details: Implement detailed view pages for each product, allowing users to see more information about the product, such as specifications, reviews, and images.

Wishlist: Create a wishlist feature that allows users to save products they are interested in purchasing later.

Add to Cart: Enable users to add products to their shopping cart, with options to update quantities or remove items.

Checkout: Implement the checkout process, including form fields for shipping, billing, and payment information.

```

def __getitem__(self, i):
    """Return the product with the given index"""
    if i < 0:
        i += len(self.products)
    return self.products[i]

def __len__(self):
    """Return the number of products"""
    return len(self.products)

def __str__(self):
    """Return the string representation of the catalog"""
    return "Catalog with %d products" % len(self.products)

def __repr__(self):
    """Return the repr string of the catalog"""
    return "Catalog(%s)" % str(self.products)

def __iter__(self):
    """Return the iterator of the catalog"""
    return iter(self.products)

def __contains__(self, product):
    """Return True if the product is in the catalog, False otherwise"""
    return product in self.products

def __getitem__(self, i):
    """Return the product with the given index"""
    if i < 0:
        i += len(self.products)
    return self.products[i]

def __len__(self):
    """Return the number of products"""
    return len(self.products)

def __str__(self):
    """Return the string representation of the catalog"""
    return "Catalog with %d products" % len(self.products)

def __repr__(self):
    """Return the repr string of the catalog"""
    return "Catalog(%s)" % str(self.products)

def __iter__(self):
    """Return the iterator of the catalog"""
    return iter(self.products)

def __contains__(self, product):
    """Return True if the product is in the catalog, False otherwise"""
    return product in self.products

```

2 Product price calculation :

```

import pandas as pd
import numpy as np

# Create a DataFrame with 10 rows and 5 columns
df = pd.DataFrame({
    'id': range(10),
    'name': ['Alice', 'Bob', 'Charlie', 'David', 'Eve', 'Frank', 'Grace', 'Heidi', 'Ivan', 'Julia'],
    'age': [25, 30, 35, 40, 45, 50, 55, 60, 65, 70],
    'salary': [50000, 60000, 70000, 80000, 90000, 100000, 110000, 120000, 130000, 140000],
    'gender': ['M', 'F', 'M', 'F', 'M', 'F', 'M', 'F', 'M', 'F']
})

# Filter rows where salary is greater than 100,000
filtered_df = df[df['salary'] > 100000]

# Sort the filtered DataFrame by age in descending order
sorted_df = filtered_df.sort_values('age', ascending=False)

# Print the sorted DataFrame
print(sorted_df)

```

3 Card Icon functionality

4 Wishlist Icon functionality

```
import { useState } from "react";
import { useRouter } from "next/router";
import { useCart } from "context/cart";
import { useWishlist } from "context/wishlist";

const WishlistIcon = () => {
  const { id, name, image, price } = useProduct();
  const { addToWishlist } = useWishlist();
  const { removeFromWishlist } = useWishlist();

  const handleClick = () => {
    if (isInWishlist) {
      removeFromWishlist(id);
    } else {
      addToWishlist(id);
    }
  };

  return (
    <button
      type="button"
      className="wishlist-icon"
      onClick={handleClick}
    >
      {isInWishlist ? <img alt="Remove from wishlist icon" /> : <img alt="Add to wishlist icon" />}
    </button>
  );
};

export default WishlistIcon;
```

Technical Documentation :

Project Overview

The project is a dynamic e-commerce marketplace built using Next.js and Sanity CMS. It includes features like product listing, product details, shopping cart, wishlist, and category-