MIT Art, Design and Technology University
MIT School of Computing, Pune

Department of Information Technology

Lab Manual

Practical - CPAD

Class - T.Y. (SEM-VI), SMAD

Batch - SMAD-TY

Mr.

Vedant Prasad Parulekar

A.Y. 2024 – 2025 (SEM-II)

Experiment No.1

Problem Statement:

To design and develop a dynamic platform for TechKeeda that integrates a tech blog and an e-commerce system, enabling users to access tech-related content and shop for technology products seamlessly within the same ecosystem.

Objective:

- To create an engaging and informative tech blog where users can read about the latest technology trends, reviews, tutorials, and product comparisons.
- To implement a tech e-commerce page that offers a curated selection of tech gadgets and accessories.
- To integrate a shopping cart system that allows users to browse, add items, and securely purchase products from the website.
- To ensure a smooth user experience across both blog and e-commerce functionalities.

Theory:

TechKeeda is a hybrid platform that combines the elements of a technology blog and an e-commerce website. The tech blog component serves as an information hub, offering detailed insights into new technologies, industry news, how-to guides, and product reviews. This builds trust and attracts tech-savvy readers.

The e-commerce component allows users to purchase tech products recommended or reviewed in the blog. The cart system facilitates product selection, quantity adjustments, and secure checkout, thereby enhancing the user experience and encouraging conversions.

The combination of content and commerce is designed to create a holistic tech ecosystem, where users are both informed and empowered to make purchasing decisions based on expert insights.

The TechKeeda website is built as a Single Page Application (SPA) using modern frontend technologies like React + TypeScript, optimized with tools such as Vite and Tailwind CSS for rapid loading and responsive design.

• The technology blog section acts as a digital library of valuable tech knowledge, driving organic traffic through search engine optimization (SEO). It covers a wide range of topics from gadget reviews and programming tutorials to industry news, catering to both beginners and professionals.

• The e-commerce integration complements the blog by offering a personalized shopping experience. Products are often contextually linked with blog articles — for example, an article reviewing the best mechanical keyboards directly links to those keyboards in the store.

• The cart system on the website uses stateful logic (like React Context or Redux) to manage product selection and transactions. Users can seamlessly browse, add to cart, and check out — all within the same interface.

- Being a web platform, it is accessible without installation, works across all browsers, and is ideal for first-time visitors and SEO traffic.

Technical Foundation:

- Frontend Stack: React + TypeScript + Tailwind CSS

- Routing: React Router enables seamless transitions between blog posts, product listings, and cart pages.

- State Management: Cart, products, and user interactions are managed using React Context or similar state libraries.

- CMS Integration (optional): Future scalability may involve integrating a headless CMS like Strapi for dynamic blog content.

Behavioral Insight:

- Web users often arrive via Google search, which makes SEO critical — hence the website emphasizes:

- Blog metadata optimization

- Clean URL structures

- Fast page load speeds (via Vite)

- Users might begin with reading a blog, gain trust in the brand, and transition into buyers via internal product links — a content-to-commerce funnel.

Code:

A. Home page:

code:

```
import React from 'react';
import { blogPosts } from '../data/blogs';
import { products } from '../data/products';
import BlogCard from '../components/blog/BlogCard';
import ProductCard from '../components/products/ProductCard';
import { ArrowRight, Laptop, Cpu, ChevronRight } from 'lucide-react';

interface HomePageProps {
 setActivePage: (page: string) => void;
 setSelectedBlog: (blog: any) => void;
 setSelectedProduct: (product: any) => void;
}
```

```
const HomePage: React.FC<HomePageProps> = ({
 setActivePage,
 setSelectedBlog,
 setSelectedProduct
}) => {
 return (
  <div className="pt-20 bg-gradient-to-b from-gray-50 to-white">
   {/* Hero Section */}
   <section className="py-16 md:py-24">
    <div className="container mx-auto px-4 md:px-6">
     <div className="flex flex-col md:flex-row items-center">
      <div className="md:w-1/2 mb-10 md:mb-0 md:pr-12">
       <h1 className="text-3xl md:text-4xl lg:text-5xl font-bold text-gray-900 mb-4 leading-tight">
        Your Ultimate Tech <br />
        <span className="text-blue-700">Resource Hub</span>
       </h1>
       <p className="text-lg text-gray-600 mb-8 leading-relaxed">
        Stay informed with the latest tech insights and shop premium gadgets. Discover expert articles and cutting-edge products all in one place.
       </p>
       <div className="flex flex-col sm:flex-row space-y-3 sm:space-y-0 sm:space-x-4">
        <button
         onClick={() => setActivePage('blog')}
         className="bg-blue-700 text-white px-6 py-3 rounded-md hover:bg-blue-800 transition-colors flex items-center"
        >
         Explore Blogs
         <ArrowRight className="ml-2 w-5 h-5" />
        </button>
        <button
         onClick={() => setActivePage('shop')}
```

```
              className="border border-blue-700 text-blue-700 px-6 py-3 rounded-md hover:bg-blue-50 transition-colors"
            >
              Shop Products
            </button>
          </div>
        </div>
        <div className="md:w-1/2">
          <img
            src="https://images.pexels.com/photos/1181244/pexels-photo-1181244.jpeg?auto=compress&cs=tinysrgb&w=1260&h=750&dpr=2"
            alt="Tech devices on desk"
            className="rounded-lg shadow-xl"
          />
        </div>
      </div>
    </div>
  </section>

  {/* Featured Blogs Section */}
  <section className="py-16 bg-gray-50">
    <div className="container mx-auto px-4 md:px-6">
      <div className="flex items-center justify-between mb-10">
        <h2 className="text-2xl md:text-3xl font-bold text-gray-900">Featured Articles</h2>
        <button
          onClick={() => setActivePage('blog')}
          className="text-blue-700 hover:text-blue-900 transition-colors flex items-center text-sm font-medium"
        >
          View All
          <ChevronRight className="ml-1 w-4 h-4" />
        </button>
      </div>

      <div className="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-3 gap-8">
        {blogPosts.slice(0, 3).map((blog) => (
          <BlogCard
            key={blog.id}
            blog={blog}
            setSelectedBlog={setSelectedBlog}
          />
        ))}
      </div>
    </div>
  </section>

  {/* Featured Products Section */}
  <section className="py-16">
    <div className="container mx-auto px-4 md:px-6">
      <div className="flex items-center justify-between mb-10">
        <h2 className="text-2xl md:text-3xl font-bold text-gray-900">Top Products</h2>
        <button
          onClick={() => setActivePage('shop')}
          className="text-blue-700 hover:text-blue-900 transition-colors flex items-center text-sm font-medium"
        >
          View All
          <ChevronRight className="ml-1 w-4 h-4" />
        </button>
      </div>

      <div className="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-3 gap-8">
        {products.slice(0, 3).map((product) => (
        <ProductCard

         key={product.id}

         product={product}

         setSelectedProduct={setSelectedProduct}

        />

       ))}

      </div>

     </div>

    </section>


    {/* Categories Section */}

    <section className="py-16 bg-gray-50">

     <div className="container mx-auto px-4 md:px-6">

      <h2 className="text-2xl md:text-3xl font-bold text-gray-900 mb-10 text-center">

       Browse by Category

      </h2>


      <div className="grid grid-cols-1 sm:grid-cols-2 lg:grid-cols-4 gap-6">

       <div className="bg-white p-6 rounded-lg shadow-sm hover:shadow-md transition-shadow text-center">

        <div className="w-16 h-16 bg-blue-100 rounded-full flex items-center justify-center mx-auto mb-4">

         <Laptop className="w-8 h-8 text-blue-700" />

        </div>

        <h3 className="text-lg font-semibold text-gray-900 mb-2">Computing</h3>

        <p className="text-gray-600 text-sm mb-4">Laptops, Desktops, Components</p>

        <button

         onClick={() => setActivePage('shop')}

         className="text-blue-700 hover:text-blue-900 transition-colors flex items-center justify-center mx-auto
  text-sm font-medium"

        >

         View Products

         <ChevronRight className="ml-1 w-4 h-4" />
```

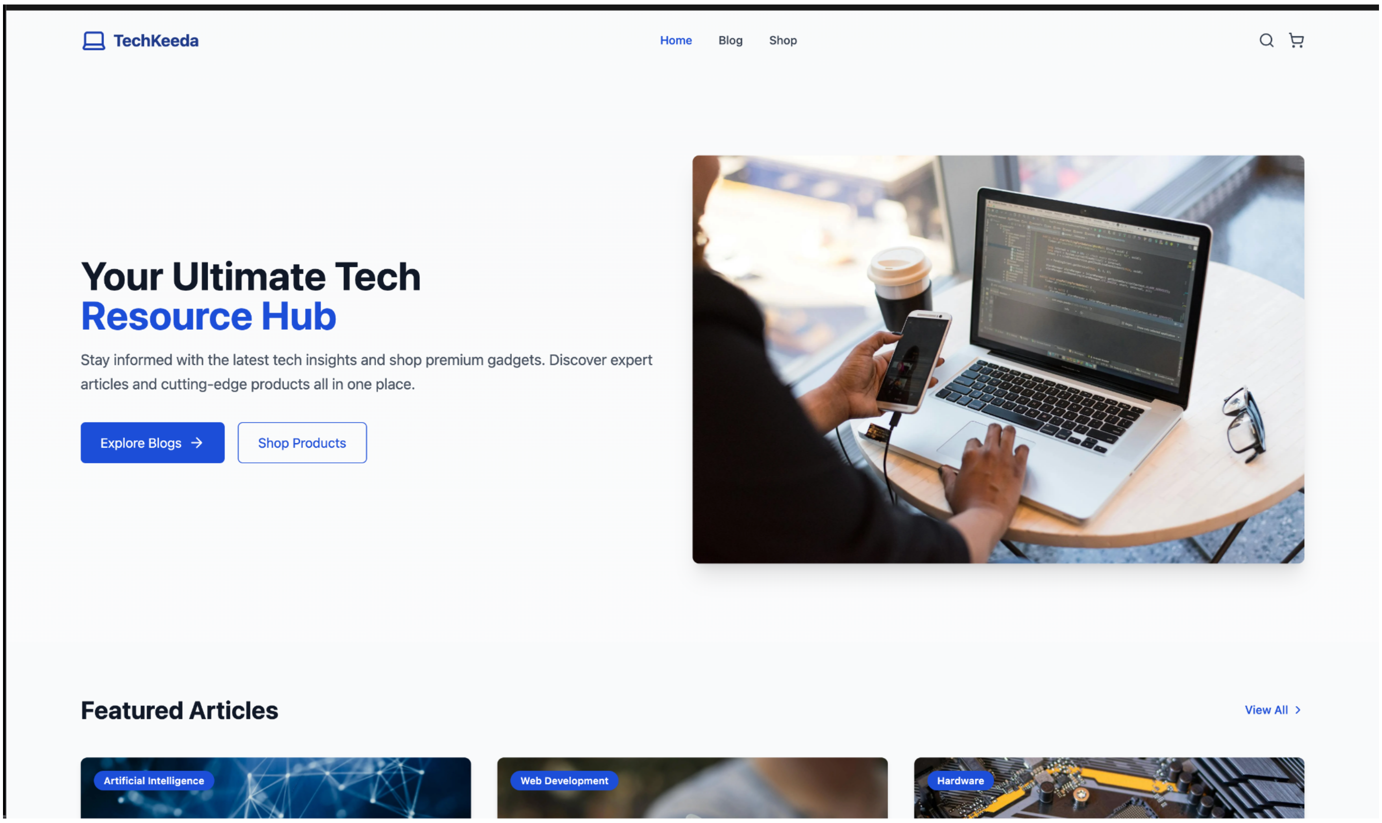
```
            </button>
          </div>

          <div className="bg-white p-6 rounded-lg shadow-sm hover:shadow-md transition-shadow text-center">
            <div className="w-16 h-16 bg-blue-100 rounded-full flex items-center justify-center mx-auto mb-4">
              <svg xmlns="http://www.w3.org/2000/svg" className="w-8 h-8 text-blue-700" viewBox="0 0 24 24" fill="none" stroke="currentColor" strokeWidth="2" strokeLinecap="round" strokeLinejoin="round">
                <rect x="5" y="2" width="14" height="20" rx="2" ry="2" />
                <path d="M12 18h.01" />
              </svg>
            </div>
            <h3 className="text-lg font-semibold text-gray-900 mb-2">Mobile Devices</h3>
            <p className="text-gray-600 text-sm mb-4">Smartphones, Tablets, Accessories</p>
            <button
              onClick={() => setActivePage('shop')}
              className="text-blue-700 hover:text-blue-900 transition-colors flex items-center justify-center mx-auto text-sm font-medium"
            >
              View Products
              <ChevronRight className="ml-1 w-4 h-4" />
            </button>
          </div>

          <div className="bg-white p-6 rounded-lg shadow-sm hover:shadow-md transition-shadow text-center">
            <div className="w-16 h-16 bg-blue-100 rounded-full flex items-center justify-center mx-auto mb-4">
              <svg xmlns="http://www.w3.org/2000/svg" className="w-8 h-8 text-blue-700" viewBox="0 0 24 24" fill="none" stroke="currentColor" strokeWidth="2" strokeLinecap="round" strokeLinejoin="round">
                <circle cx="12" cy="12" r="10" />
                <path d="M12 2a14.5 14.5 0 0 0 0 20 14.5 14.5 0 0 0 0-20" />
                <path d="M2 12h20" />
              </svg>
```

```
                </div>

                <h3 className="text-lg font-semibold text-gray-900 mb-2">Networking</h3>

                <p className="text-gray-600 text-sm mb-4">Routers, Switches, Wi-Fi Systems</p>

                <button

                  onClick={() => setActivePage('shop')}

                  className="text-blue-700 hover:text-blue-900 transition-colors flex items-center justify-center mx-auto text-sm font-medium"

                >

                  View Products

                  <ChevronRight className="ml-1 w-4 h-4" />

                </button>

              </div>



              <div className="bg-white p-6 rounded-lg shadow-sm hover:shadow-md transition-shadow text-center">

                <div className="w-16 h-16 bg-blue-100 rounded-full flex items-center justify-center mx-auto mb-4">

                  <Cpu className="w-8 h-8 text-blue-700" />

                </div>

                <h3 className="text-lg font-semibold text-gray-900 mb-2">Components</h3>

                <p className="text-gray-600 text-sm mb-4">CPUs, GPUs, Memory, Storage</p>

                <button

                  onClick={() => setActivePage('shop')}

                  className="text-blue-700 hover:text-blue-900 transition-colors flex items-center justify-center mx-auto text-sm font-medium"

                >

                  View Products

                  <ChevronRight className="ml-1 w-4 h-4" />

                </button>

              </div>

            </div>

          </div>

        </section>
```

```
    {/* Newsletter Section */}
    <section className="py-16 bg-blue-700">
      <div className="container mx-auto px-4 md:px-6">
        <div className="max-w-3xl mx-auto text-center">
          <h2 className="text-2xl md:text-3xl font-bold text-white mb-4">
            Stay Updated with TechKeeda
          </h2>
          <p className="text-blue-100 mb-8">
            Subscribe to our newsletter for the latest tech news, product releases, and exclusive offers.
          </p>
          <div className="flex flex-col sm:flex-row space-y-3 sm:space-y-0 max-w-lg mx-auto">
            <input
              type="email"
              placeholder="Enter your email"
              className="flex-grow px-4 py-3 rounded-l-md sm:rounded-r-none focus:outline-none focus:ring-2 focus:ring-blue-500"
            />
            <button className="bg-gray-900 text-white px-6 py-3 rounded-md sm:rounded-l-none hover:bg-gray-800 transition-colors">
              Subscribe
            </button>
          </div>
        </div>
      </div>
    </section>
  </div>
 );
};

export default HomePage;
```

Output:

A. Index/Home page output:



TechKeeda

Home Blog Shop

Your Ultimate Tech Resource Hub

Stay informed with the latest tech insights and shop premium gadgets. Discover expert articles and cutting-edge products all in one place.

Explore Blogs → | Shop Products

Featured Articles

View All >

Code:

B. menu/product page:

code:

```
import React, { useState } from 'react';

import { products } from '../data/products';

import ProductCard from '../components/products/ProductCard';

import { Search, Filter, ShoppingBag } from 'lucide-react';


interface ShopPageProps {

 setSelectedProduct: (product: any) => void;

}
```

```
const ShopPage: React.FC<ShopPageProps> = ({ setSelectedProduct }) => {
 const [searchTerm, setSearchTerm] = useState('');
 const [selectedCategory, setSelectedCategory] = useState('All');
 const [showInStock, setShowInStock] = useState(false);
 const [sortBy, setSortBy] = useState('featured');
 const [isFilterOpen, setIsFilterOpen] = useState(false);

 const categories = ['All', ...Array.from(new Set(products.map(product => product.category)))];

 const filteredProducts = products.filter(product => {
   const matchesSearch = product.name.toLowerCase().includes(searchTerm.toLowerCase()) ||
                         product.description.toLowerCase().includes(searchTerm.toLowerCase());

   const matchesCategory = selectedCategory === 'All' || product.category === selectedCategory;

   const matchesStock = showInStock ? product.inStock : true;

   return matchesSearch && matchesCategory && matchesStock;
 }).sort((a, b) => {
   if (sortBy === 'price-low') {
     return a.price - b.price;
   } else if (sortBy === 'price-high') {
     return b.price - a.price;
   } else if (sortBy === 'rating') {
     return b.rating - a.rating;
   }
   return 0; // 'featured' - no sorting
 });

 return (
    <div className="pt-20 pb-16">
      {/* Shop Hero */}
      <section className="bg-blue-700 py-16">
        <div className="container mx-auto px-4 md:px-6 text-center">
          <div className="max-w-3xl mx-auto">
            <div className="inline-block bg-blue-800 p-2 rounded-full mb-4">
              <ShoppingBag className="w-6 h-6 text-white" />
            </div>
            <h1 className="text-3xl md:text-4xl font-bold text-white mb-4">
              Tech Products Shop
            </h1>
            <p className="text-blue-100 text-lg mb-8">
              Premium tech gadgets and accessories for professionals and enthusiasts
            </p>
            <div className="relative max-w-2xl mx-auto">
              <div className="absolute inset-y-0 left-0 pl-3 flex items-center pointer-events-none">
                <Search className="h-5 w-5 text-gray-400" />
              </div>
              <input
                type="text"
                placeholder="Search products..."
                value={searchTerm}
                onChange={(e) => setSearchTerm(e.target.value)}
                className="block w-full pl-10 pr-3 py-3 rounded-md bg-white text-gray-900 placeholder-gray-500 focus:outline-none focus:ring-2 focus:ring-blue-500"
              />
            </div>
          </div>
        </div>
      </section>
```

```
{/* Shop Content */}
<section className="py-12">
 <div className="container mx-auto px-4 md:px-6">
  <div className="flex flex-col md:flex-row">
   {/* Sidebar - Desktop */}
   <div className="hidden md:block w-64 mr-8">
    <div className="bg-white rounded-lg shadow-sm p-6 sticky top-24">
     <h2 className="text-lg font-semibold text-gray-900 mb-4">Filters</h2>

     <div className="mb-6">
      <h3 className="text-sm font-medium text-gray-900 mb-2">Categories</h3>
      <div className="space-y-2">
       {categories.map((category) => (
        <div key={category} className="flex items-center">
         <input
          id={`category-${category}`}
          type="radio"
          checked={selectedCategory === category}
          onChange={() => setSelectedCategory(category)}
          className="h-4 w-4 text-blue-700 focus:ring-blue-500"
         />
         <label htmlFor={`category-${category}`} className="ml-2 text-sm text-gray-700">
          {category}
         </label>
        </div>
       ))}
      </div>
     </div>

     <div className="mb-6">
```

```
          <h3 className="text-sm font-medium text-gray-900 mb-2">Availability</h3>
          <div className="flex items-center">
            <input
              id="in-stock"
              type="checkbox"
              checked={showInStock}
              onChange={() => setShowInStock(!showInStock)}
              className="h-4 w-4 text-blue-700 focus:ring-blue-500"
            />
            <label htmlFor="in-stock" className="ml-2 text-sm text-gray-700">
              In Stock Only
            </label>
          </div>
        </div>

        <div>
          <h3 className="text-sm font-medium text-gray-900 mb-2">Sort By</h3>
          <select
            value={sortBy}
            onChange={(e) => setSortBy(e.target.value)}
            className="block w-full bg-white border border-gray-300 rounded-md py-2 px-3 text-sm focus:outline-none focus:ring-blue-500 focus:border-blue-500"
          >
            <option value="featured">Featured</option>
            <option value="price-low">Price: Low to High</option>
            <option value="price-high">Price: High to Low</option>
            <option value="rating">Highest Rated</option>
          </select>
        </div>
      </div>
    </div>
```

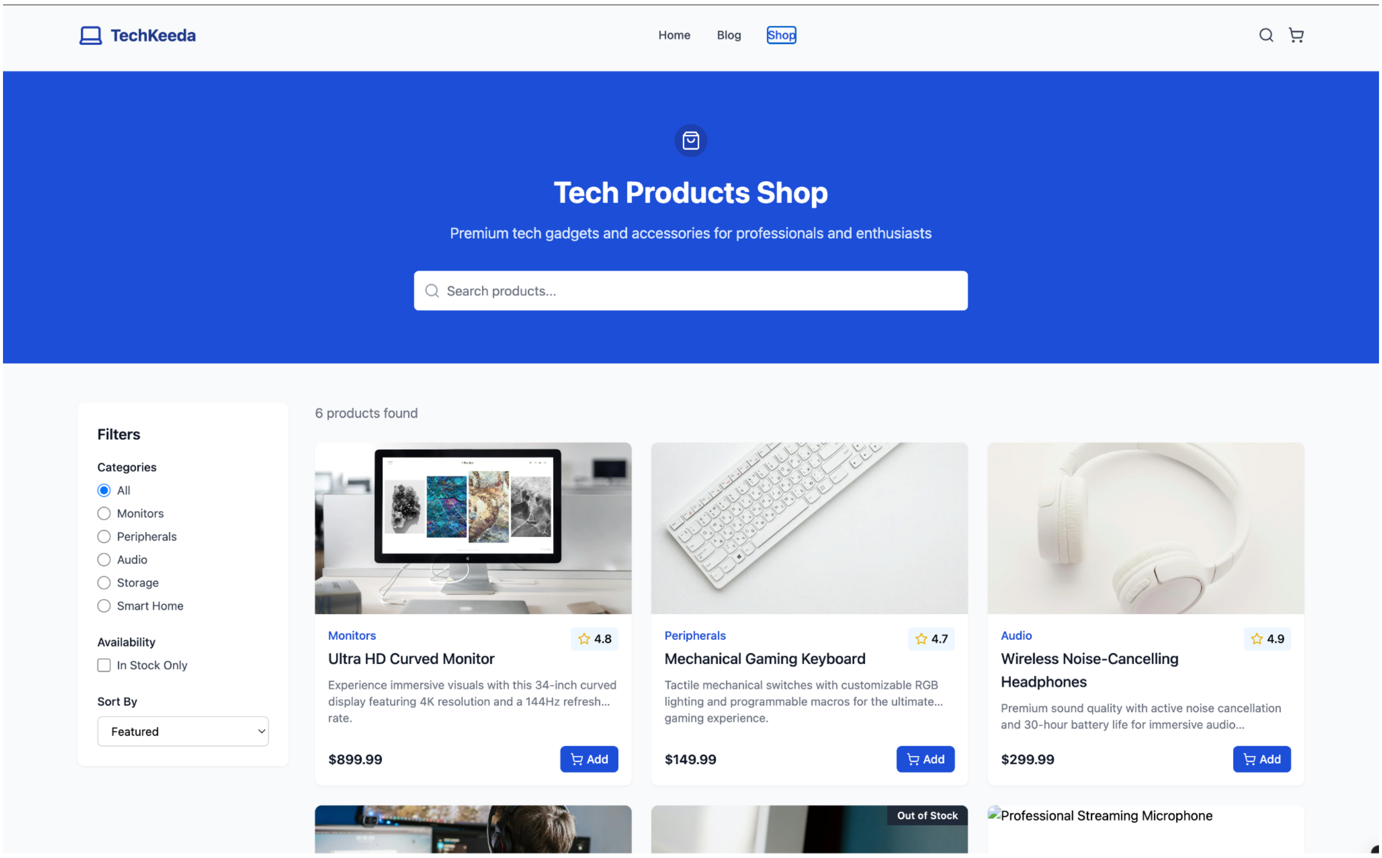
```
{/* Mobile Filter Button */}
<div className="md:hidden mb-4">
  <button
    onClick={() => setIsFilterOpen(!isFilterOpen)}
    className="w-full bg-white border border-gray-300 rounded-md py-2 px-4 flex items-center justify-center text-gray-700 hover:bg-gray-50 transition-colors"
  >
    <Filter className="w-5 h-5 mr-2" />
    Filters & Sorting
  </button>
</div>

{/* Mobile Filters */}
{isFilterOpen && (
  <div className="md:hidden mb-6 bg-white rounded-lg shadow-sm p-4">
    <div className="mb-4">
      <h3 className="text-sm font-medium text-gray-900 mb-2">Categories</h3>
      <div className="flex flex-wrap gap-2">
        {categories.map((category) => (
          <button
            key={category}
            onClick={() => setSelectedCategory(category)}
            className={`px-3 py-1 rounded-full text-xs ${
              selectedCategory === category
                ? 'bg-blue-700 text-white'
                : 'bg-gray-100 text-gray-700'
            }`}
          >
            {category}
          </button>
    ))}
  </div>
</div>

<div className="mb-4">
  <div className="flex items-center">
    <input
      id="mobile-in-stock"
      type="checkbox"
      checked={showInStock}
      onChange={() => setShowInStock(!showInStock)}
      className="h-4 w-4 text-blue-700 focus:ring-blue-500"
    />
    <label htmlFor="mobile-in-stock" className="ml-2 text-sm text-gray-700">
      In Stock Only
    </label>
  </div>
</div>

<div>
  <h3 className="text-sm font-medium text-gray-900 mb-2">Sort By</h3>
  <select
    value={sortBy}
    onChange={(e) => setSortBy(e.target.value)}
    className="block w-full bg-white border border-gray-300 rounded-md py-2 px-3 text-sm focus:outline-none focus:ring-blue-500 focus:border-blue-500"
  >
    <option value="featured">Featured</option>
    <option value="price-low">Price: Low to High</option>
    <option value="price-high">Price: High to Low</option>
    <option value="rating">Highest Rated</option>
```

```
       </select>
      </div>
     </div>
    )}

    {/* Product Grid */}
    <div className="flex-1">
     {filteredProducts.length === 0 ? (
      <div className="text-center py-12">
       <p className="text-gray-500 text-lg">No products found. Try a different search term or filter.</p>
      </div>
     ) : (
      <>
       <p className="text-gray-500 mb-6">{filteredProducts.length} products found</p>
       <div className="grid grid-cols-1 sm:grid-cols-2 lg:grid-cols-3 gap-6">
        {filteredProducts.map((product) => (
         <ProductCard
          key={product.id}
          product={product}
          setSelectedProduct={setSelectedProduct}
         />
        ))}
       </div>
      </>
     )}
    </div>
   </div>
  </div>
 </section>
</div>
```

```
 );

};


export default ShopPage;
```

Output:

B. menu/product page output:



Tech Products Shop

Premium tech gadgets and accessories for professionals and enthusiasts

Search products...

Filters

Categories

● All
○ Monitors
○ Peripherals
○ Audio
○ Storage
○ Smart Home

Availability

□ In Stock Only

Sort By

Featured

6 products found

Monitors ☆ 4.8
Ultra HD Curved Monitor
Experience immersive visuals with this 34-inch curved display featuring 4K resolution and a 144Hz refresh... rate.
$899.99 Add

Peripherals ☆ 4.7
Mechanical Gaming Keyboard
Tactile mechanical switches with customizable RGB lighting and programmable macros for the ultimate... gaming experience.
$149.99 Add

Audio ☆ 4.9
Wireless Noise-Cancelling Headphones
Premium sound quality with active noise cancellation and 30-hour battery life for immersive audio...
$299.99 Add

Out of Stock

Professional Streaming Microphone

Code:

b. Product detail

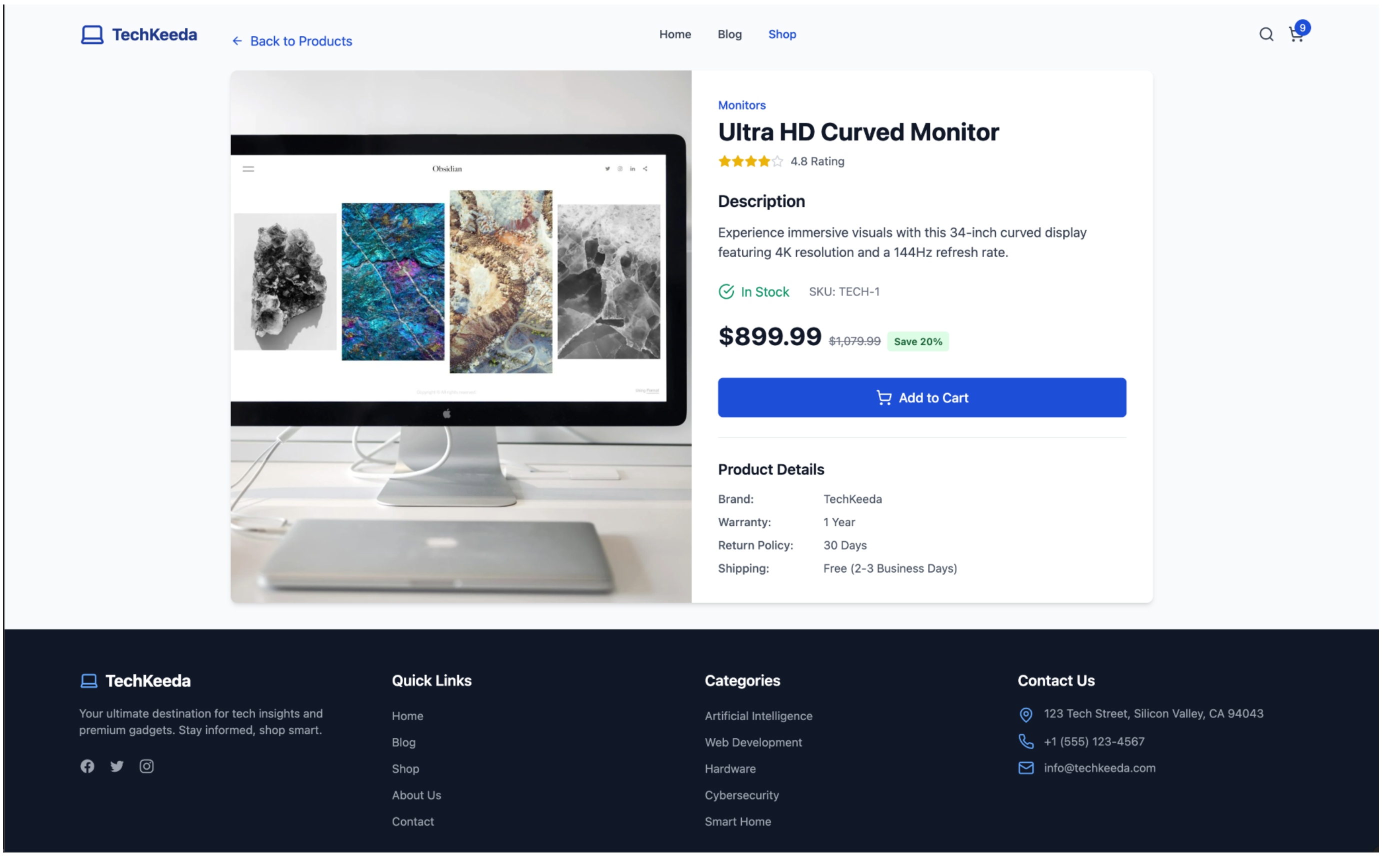
code:

```
import React from 'react';

import { Product } from '../../types';

import { useCart } from '../../contexts/CartContext';

import { Star, ShoppingCart } from 'lucide-react';

import { formatCurrency } from '../../utils/formatters';


interface ProductCardProps {

 product: Product;

 setSelectedProduct: (product: Product) => void;

}


const ProductCard: React.FC<ProductCardProps> = ({ product, setSelectedProduct }) => {

 const { addToCart } = useCart();
const handleAddToCart = (e: React.MouseEvent) => {
  e.stopPropagation();
  addToCart(product);
};

return (
  <div
    className="bg-white rounded-lg overflow-hidden shadow-sm hover:shadow-md transition-shadow duration-300 flex flex-col h-full group"
    onClick={() => setSelectedProduct(product)}
  >
    <div className="relative h-52 overflow-hidden">
      <img
        src={product.imageUrl}
        alt={product.name}
        className="w-full h-full object-cover transition-transform duration-500 group-hover:scale-105"
      />
      {!product.inStock && (
        <div className="absolute top-0 right-0 bg-gray-800 text-white text-xs font-semibold px-3 py-1">
          Out of Stock
        </div>
      )}
    </div>
    <div className="p-4 flex flex-col flex-grow">
      <div className="flex justify-between items-start">
        <div>
          <p className="text-sm text-blue-700 font-medium mb-1">{product.category}</p>
          <h3 className="text-lg font-medium text-gray-900 group-hover:text-blue-700 transition-colors line-clamp-2">
            {product.name}
            </h3>
          </div>
          <div className="flex items-center bg-blue-50 rounded px-2 py-1">
            <Star className="w-4 h-4 text-yellow-500 mr-1" />
            <span className="text-sm font-medium">{product.rating.toFixed(1)}</span>
          </div>
        </div>

        <p className="mt-2 text-gray-500 text-sm line-clamp-2 flex-grow">{product.description}</p>

        <div className="mt-4 flex items-center justify-between">
          <span className="text-gray-900 font-semibold">{formatCurrency(product.price)}</span>

          <button
            onClick={handleAddToCart}
            disabled={!product.inStock}
            className={`flex items-center rounded-md px-3 py-1.5 text-sm font-medium transition-colors ${
              product.inStock
                ? 'bg-blue-700 text-white hover:bg-blue-800'
                : 'bg-gray-200 text-gray-500 cursor-not-allowed'
            }`}
          >
            <ShoppingCart className="w-4 h-4 mr-1" />
            Add
          </button>
        </div>
      </div>
    </div>
  );
};
```

```
export default ProductCard;
```

Output:

b. cart page output:



Monitors

Ultra HD Curved Monitor

★★★★☆ 4.8 Rating

Description

Experience immersive visuals with this 34-inch curved display featuring 4K resolution and a 144Hz refresh rate.

✓ In Stock SKU: TECH-1

$899.99 ~~$1,079.99~~ Save 20%

Product Details

Brand:	TechKeeda
Warranty:	1 Year
Return Policy:	30 Days
Shipping:	Free (2-3 Business Days)

TechKeeda

Your ultimate destination for tech insights and premium gadgets. Stay informed, shop smart.

Quick Links

- Home
- Blog
- Shop
- About Us
- Contact

Categories

- Artificial Intelligence
- Web Development
- Hardware
- Cybersecurity
- Smart Home

Contact Us

123 Tech Street, Silicon Valley, CA 94043
+1 (555) 123-4567
info@techkeeda.com

Code:

C. cart page:

code:

```
import React from 'react';

import { useCart } from '../../contexts/CartContext';

import { X, Plus, Minus, ShoppingBag, ArrowRight } from 'lucide-react';

import { formatCurrency } from '../../utils/formatters';
```

```
interface CartSidebarProps {
  isOpen: boolean;
  onClose: () => void;
}

const CartSidebar: React.FC<CartSidebarProps> = ({ isOpen, onClose }) => {
  const { cart, removeFromCart, updateQuantity, getCartTotal, clearCart } = useCart();

  if (!isOpen) return null;

  return (
    <div className="fixed inset-0 z-50 overflow-hidden">
      <div
        className="absolute inset-0 bg-black bg-opacity-50 transition-opacity"
        onClick={onClose}
      />

      <div className="absolute inset-y-0 right-0 max-w-full flex">
        <div className="relative w-screen max-w-md">
          <div className="h-full flex flex-col bg-white shadow-xl overflow-y-auto">
            {/* Header */}
            <div className="px-4 pt-5 pb-4 sm:px-6 flex items-center justify-between">
              <h2 className="text-lg font-medium text-gray-900 flex items-center">
                <ShoppingBag className="w-5 h-5 mr-2" />
                Your Cart
              </h2>
              <button
                className="text-gray-400 hover:text-gray-500 transition-colors"
                onClick={onClose}
                aria-label="Close"
```

```
 >
  <X className="w-6 h-6" />
 </button>
</div>

{/* Cart Items */}
<div className="flex-1 px-4 sm:px-6">
 {cart.length === 0 ? (
  <div className="py-12 text-center">
   <ShoppingBag className="w-16 h-16 mx-auto text-gray-300" />
   <p className="mt-4 text-gray-500">Your cart is empty</p>
   <button
    className="mt-4 text-blue-600 hover:text-blue-800 transition-colors flex items-center mx-auto"
    onClick={onClose}
   >
    Continue Shopping
    <ArrowRight className="w-4 h-4 ml-1" />
   </button>
  </div>
 ) : (
  <ul className="divide-y divide-gray-200">
   {cart.map((item) => (
    <li key={item.product.id} className="py-4 flex">
     <div className="h-24 w-24 flex-shrink-0 overflow-hidden rounded-md border border-gray-200">
      <img
       src={item.product.imageUrl}
       alt={item.product.name}
       className="h-full w-full object-cover object-center"
      />
     </div>
```

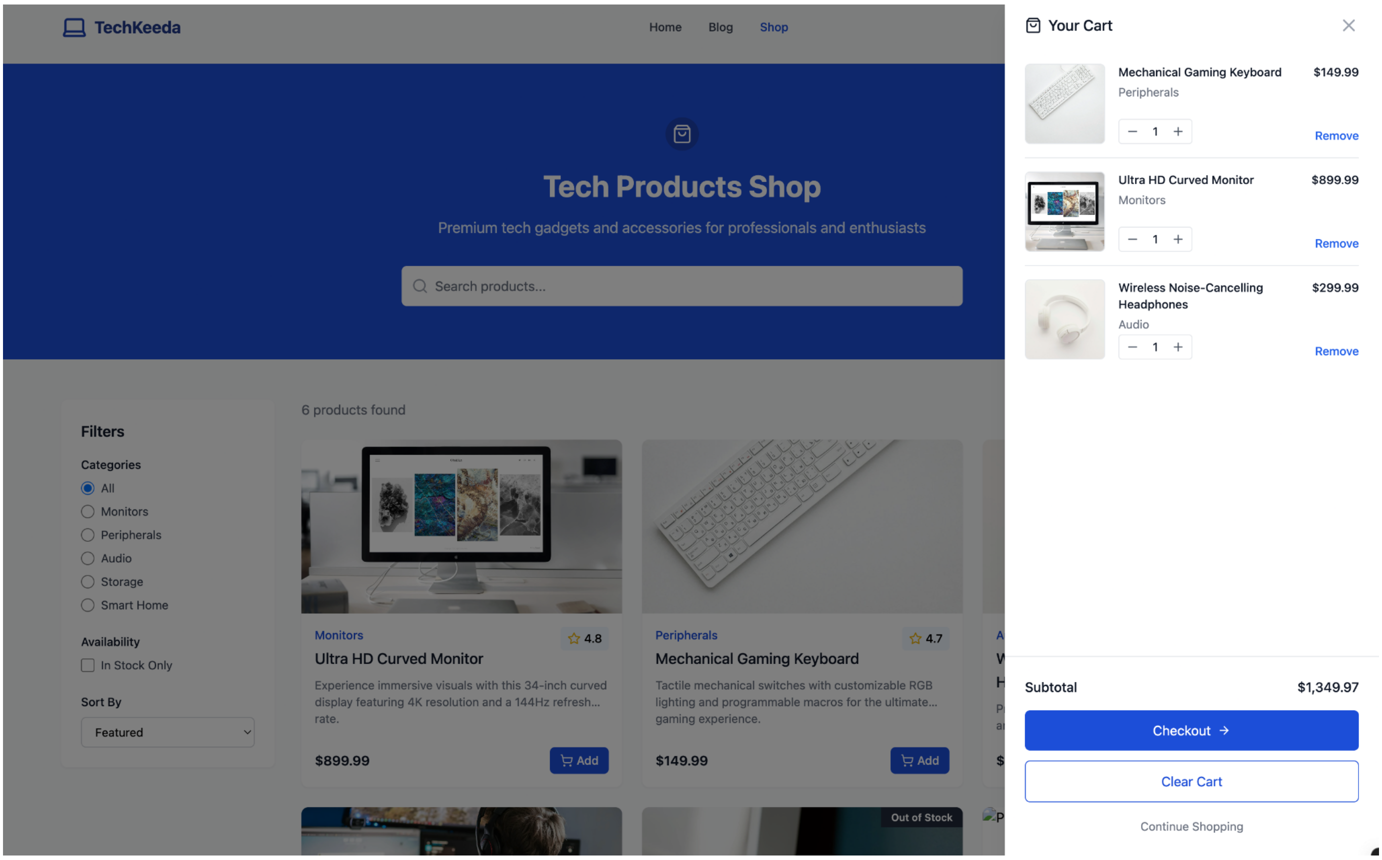
```
<div className="ml-4 flex flex-1 flex-col">
  <div className="flex justify-between">
    <div>
      <h3 className="text-sm font-medium text-gray-900">
        {item.product.name}
      </h3>
      <p className="mt-1 text-sm text-gray-500">
        {item.product.category}
      </p>
    </div>
    <p className="text-sm font-medium text-gray-900">
      {formatCurrency(item.product.price)}
    </p>
  </div>

  <div className="flex flex-1 items-end justify-between text-sm">
    <div className="flex items-center border border-gray-200 rounded">
      <button
        className="px-2 py-1 text-gray-600 hover:text-blue-600 transition-colors"
        onClick={() => updateQuantity(item.product.id, item.quantity - 1)}
        aria-label="Decrease quantity"
      >
        <Minus className="w-4 h-4" />
      </button>
      <span className="px-2 py-1 text-gray-900">{item.quantity}</span>
      <button
        className="px-2 py-1 text-gray-600 hover:text-blue-600 transition-colors"
        onClick={() => updateQuantity(item.product.id, item.quantity + 1)}
        aria-label="Increase quantity"
```

```
                >
                  <Plus className="w-4 h-4" />
                </button>
              </div>

              <button
                type="button"
                className="font-medium text-blue-600 hover:text-blue-800 transition-colors"
                onClick={() => removeFromCart(item.product.id)}
              >
                Remove
              </button>
            </div>
          </div>
        </li>
      ))}
    </ul>
  )}
</div>

{/* Footer */}
{cart.length > 0 && (
  <div className="border-t border-gray-200 px-4 py-6 sm:px-6">
    <div className="flex justify-between text-base font-medium text-gray-900 mb-4">
      <p>Subtotal</p>
      <p>{formatCurrency(getCartTotal())}</p>
    </div>
    <div className="flex flex-col space-y-3">
      <button
        className="w-full bg-blue-700 text-white py-3 px-4 rounded-md hover:bg-blue-800 transition-colors flex items-center justify-center"
```

```
                  >
                    Checkout
                    <ArrowRight className="w-4 h-4 ml-2" />
                  </button>
                  <button
                    className="w-full text-blue-700 border border-blue-700 py-3 px-4 rounded-md hover:bg-blue-50 transition-colors"
                    onClick={clearCart}
                  >
                    Clear Cart
                  </button>
                </div>
                <div className="mt-4 text-center">
                  <button
                    className="text-sm text-gray-500 hover:text-gray-700 transition-colors"
                    onClick={onClose}
                  >
                    Continue Shopping
                  </button>
                </div>
              </div>
            )}
          </div>
        </div>
      </div>
    </div>
  );
};

export default CartSidebar;
```

Output:

C. cart page output:



Your Cart

Mechanical Gaming Keyboard	Peripherals	1	$149.99
Ultra HD Curved Monitor	Monitors	1	$899.99
Wireless Noise-Cancelling Headphones	Audio	1	$299.99

Subtotal $1,349.97

Tech Products Shop

Premium tech gadgets and accessories for professionals and enthusiasts

6 products found

Monitors ☆ 4.8

Ultra HD Curved Monitor

Experience immersive visuals with this 34-inch curved display featuring 4K resolution and a 144Hz refresh... rate.

$899.99

Peripherals ☆ 4.7

Mechanical Gaming Keyboard

Tactile mechanical switches with customizable RGB lighting and programmable macros for the ultimate... gaming experience.

$149.99

Conclusion:

The TechKeeda project is a well-structured platform combining a tech blog with a tech-focused e-commerce system, offering users both informative content and the ability to shop for products. The source code includes key components like the Home Page, Product Menu Page, and a functional Cart System, implemented using modern frameworks and technologies.

While essential components such as the Home, Shop, and Cart pages are fully developed, some pages like About Us, Contact Us, Login, and Registration appear to be either missing or under development. This indicates that the project is a solid work-in-progress with potential for expansion.

Overall, TechKeeda lays a strong foundation for a complete tech ecosystem, seamlessly integrating content and commerce — a strategy that aligns with user engagement and monetization goals.