|  |
| --- |
| Objective : Frontend: Using ReactJS, make Image Carousel, FAQ/Accordion and Shopping item list with filters. |

**App.jsx**

import React from 'react';

import Carousel from './components/Corousel/Carousel';

import Accordion from './components/Accordion/Accordion';

import ProductList from './components/ProductList/ProductList';

const App = () => {

// Sample data for Carousel

const carouselImages = [

'https://img.freepik.com/free-psd/black-friday-super-sale-web-banner-background-template\_120329-5119.jpg',

'https://img.freepik.com/free-vector/dark-style-realistic-smart-watch-horizontal-banner-with-advertising-site-vector-illustration\_1284-30193.jpg',

'https://i.pinimg.com/736x/b2/2d/29/b22d296ca37c3400fe8838d135ae2c3d.jpg',

'https://www.shutterstock.com/image-vector/sport-shoes-banner-website-button-260nw-2167767027.jpg'

];

// Sample data for Accordion

const faqItems = [

{

question: 'How do I place an order?',

answer: 'You can place an order by selecting products and clicking the checkout button.'

},

{

question: 'What payment methods do you accept?',

answer: 'We accept credit cards, PayPal, and bank transfers.'

},

{

question: 'What is your return policy?',

answer: 'You can return items within 30 days of purchase with original receipt.'

}

];

// Sample data for ProductList

const products = [

{ id: 1, name: 'Wireless Headphones', category: 'electronics', price: 99.99, image: 'https://via.placeholder.com/300' },

{ id: 2, name: 'Smart Watch', category: 'electronics', price: 199.99, image: 'https://via.placeholder.com/300' },

{ id: 3, name: 'Cotton T-Shirt', category: 'clothing', price: 24.99, image: 'https://via.placeholder.com/300' },

{ id: 4, name: 'Running Shoes', category: 'footwear', price: 89.99, image: 'https://via.placeholder.com/300' },

{ id: 5, name: 'Coffee Mug', category: 'home', price: 12.99, image: 'https://via.placeholder.com/300' },

{ id: 6, name: 'Bluetooth Speaker', category: 'electronics', price: 59.99, image: 'https://via.placeholder.com/300' },

];

return (

<div className="app">

<h1>Image Carousel</h1>

<Carousel images={carouselImages} />

<h1>Product List with Filters</h1>

<ProductList products={products} />

<h1>FAQ Accordion</h1>

<Accordion items={faqItems} />

</div>

);

};

export default App;

**Carousel.jsx**

import { useState, useEffect } from 'react';

import './Corousel.css';

const Carousel = ({ images }) => {

const [currentIndex, setCurrentIndex] = useState(0);

const nextSlide = () => {

setCurrentIndex((prevIndex) =>

prevIndex === images.length - 1 ? 0 : prevIndex + 1

);

};

const prevSlide = () => {

setCurrentIndex((prevIndex) =>

prevIndex === 0 ? images.length - 1 : prevIndex - 1

);

};

// Auto-advance slides

useEffect(() => {

const interval = setInterval(() => {

nextSlide();

}, 5000);

return () => clearInterval(interval);

}, [currentIndex]);

return (

<div className="carousel">

<button className="carousel-button prev" onClick={prevSlide}>&lt;</button>

<div className="carousel-images">

{images.map((image, index) => (

<div

key={index}

className={`carousel-slide ${index === currentIndex ? 'active' : ''}`}

>

<img src={image} alt={`Slide ${index}`} />

</div>

))}

</div>

<button className="carousel-button next" onClick={nextSlide}>&gt;</button>

<div className="carousel-indicators">

{images.map((\_, index) => (

<button

key={index}

className={`indicator ${index === currentIndex ? 'active' : ''}`}

onClick={() => setCurrentIndex(index)}

/>

))}

</div>

</div>

);

};

export default Carousel;

**ProductList.jsx**

import { useState } from 'react';

import './ProductList.css';

const ProductList = ({ products }) => {

const [filters, setFilters] = useState({

category: 'all',

priceRange: 'all',

sortBy: 'default'

});

const categories = ['all', ...new Set(products.map(p => p.category))];

const priceRanges = [

{ value: 'all', label: 'All Prices' },

{ value: '0-50', label: 'Under $50' },

{ value: '50-100', label: '$50 - $100' },

{ value: '100-200', label: '$100 - $200' },

{ value: '200+', label: 'Over $200' }

];

const filteredProducts = products.filter(product => {

// Category filter

if (filters.category !== 'all' && product.category !== filters.category) {

return false;

}

// Price range filter

if (filters.priceRange !== 'all') {

const [min, max] = filters.priceRange.split('-').map(Number);

if (filters.priceRange.endsWith('+')) {

if (product.price < parseInt(filters.priceRange)) return false;

} else if (product.price < min || product.price > max) {

return false;

}

}

return true;

}).sort((a, b) => {

// Sorting

switch (filters.sortBy) {

case 'price-asc':

return a.price - b.price;

case 'price-desc':

return b.price - a.price;

case 'name-asc':

return a.name.localeCompare(b.name);

case 'name-desc':

return b.name.localeCompare(a.name);

default:

return 0;

}

});

return (

<div className="product-list-container">

<div className="filters">

<div className="filter-group">

<label>Category:</label>

<select

value={filters.category}

onChange={(e) => setFilters({...filters, category: e.target.value})}

>

{categories.map(category => (

<option key={category} value={category}>

{category.charAt(0).toUpperCase() + category.slice(1)}

</option>

))}

</select>

</div>

<div className="filter-group">

<label>Price Range:</label>

<select

value={filters.priceRange}

onChange={(e) => setFilters({...filters, priceRange: e.target.value})}

>

{priceRanges.map(range => (

<option key={range.value} value={range.value}>

{range.label}

</option>

))}

</select>

</div>

<div className="filter-group">

<label>Sort By:</label>

<select

value={filters.sortBy}

onChange={(e) => setFilters({...filters, sortBy: e.target.value})}

>

<option value="default">Default</option>

<option value="price-asc">Price: Low to High</option>

<option value="price-desc">Price: High to Low</option>

<option value="name-asc">Name: A-Z</option>

<option value="name-desc">Name: Z-A</option>

</select>

</div>

</div>

<div className="product-list">

{filteredProducts.length > 0 ? (

filteredProducts.map(product => (

<div key={product.id} className="product-card">

<img src={product.image} alt={product.name} />

<h3>{product.name}</h3>

<p className="category">{product.category}</p>

<p className="price">${product.price.toFixed(2)}</p>

<button className="add-to-cart">Add to Cart</button>

</div>

))

) : (

<p className="no-results">No products match your filters.</p>

)}

</div>

</div>

);

};

export default ProductList;

**OUTPUT**







 



