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

**Code :-**

**App.jsx**

import './App.css'

import HomePage from './components/HomePage'

function App() {

  return (

    <HomePage />

  )

}

export default App

**App.css**

*#root {*

*max-width: 1280px;*

*margin: 0 auto;*

*padding: 2rem;*

*text-align: center;*

*}*

*.logo {*

*height: 6em;*

*padding: 1.5em;*

*will-change: filter;*

*transition: filter 300ms;*

*}*

*.logo:hover {*

*filter: drop-shadow(0 0 2em #646cffaa);*

*}*

*.logo.react:hover {*

*filter: drop-shadow(0 0 2em #61dafbaa);*

*}*

*@keyframes logo-spin {*

*from {*

*transform: rotate(0deg);*

*}*

*to {*

*transform: rotate(360deg);*

*}*

*}*

*@media (prefers-reduced-motion: no-preference) {*

*a:nth-of-type(2) .logo {*

*animation: logo-spin infinite 20s linear;*

*}*

*}*

*.card {*

*padding: 2em;*

*}*

*.read-the-docs {*

*color: #888;*

*}*

**index.css**

*:root* {

  font-family: Inter, system-ui, Avenir, Helvetica, Arial, sans-serif;

  line-height: 1.5;

  font-weight: 400;

  color-scheme: light dark;

  color: rgba(255, 255, 255, 0.87);

  background-color: #242424;

  font-synthesis: none;

  text-rendering: optimizeLegibility;

  -webkit-font-smoothing: antialiased;

  -moz-osx-font-smoothing: grayscale;

}

a {

  font-weight: 500;

  color: #646cff;

  text-decoration: inherit;

}

a*:hover* {

  color: #535bf2;

}

body {

  margin: 0;

  display: flex;

  place-items: center;

  min-width: 320px;

  min-height: 100vh;

}

h1 {

  font-size: 3.2em;

  line-height: 1.1;

}

button {

  border-radius: 8px;

  border: 1px solid transparent;

  padding: 0.6em 1.2em;

  font-size: 1em;

  font-weight: 500;

  font-family: inherit;

  background-color: #1a1a1a;

  cursor: pointer;

  transition: border-color 0.25s;

}

button*:hover* {

  border-color: #646cff;

}

button*:focus*,

button*:focus-visible* {

  outline: 4px auto -webkit-focus-ring-color;

}

@media (prefers-color-scheme: light) {

*:root* {

    color: #213547;

    background-color: #ffffff;

  }

  a*:hover* {

    color: #747bff;

  }

  button {

    background-color: #f9f9f9;

  }

}

**Components/HomePage.jsx**

import React, { useState } from 'react';

import ImageCarousel from './ImageCarousel';

import FaqAccordion from './FaqAccordion';

import ShoppingList from './ShoppingList';

import "../assets/css/HomePage.css";

const HomePage = () => {

    const [selectedComponent, setSelectedComponent] = useState('ImageCarousel');

    const renderComponent = () => {

        switch (selectedComponent) {

            case 'ImageCarousel':

                return <ImageCarousel />;

            case 'FaqAccordion':

                return <FaqAccordion />;

            case 'ShoppingList':

                return <ShoppingList />;

            default:

                return <ImageCarousel />;

        }

    };

    return (

        <div className="homepage-container">

            <div className="navigation-tabs">

                <button

                    className={`tab-button ${selectedComponent === 'ImageCarousel' ? 'active' : ''}`}

                    onClick={() => setSelectedComponent('ImageCarousel')}

                >

                    Image Carousel

                </button>

                <button

                    className={`tab-button ${selectedComponent === 'FaqAccordion' ? 'active' : ''}`}

                    onClick={() => setSelectedComponent('FaqAccordion')}

                >

                    FAQ / Accordion

                </button>

                <button

                    className={`tab-button ${selectedComponent === 'ShoppingList' ? 'active' : ''}`}

                    onClick={() => setSelectedComponent('ShoppingList')}

                >

                    Product List + Filters

                </button>

            </div>

            <div className="component-display">

                {renderComponent()}

            </div>

        </div>

    );

};

export default HomePage;

**css/HomePage.css**

.homepage-container {

    font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;

    background-color: #f5f5f5;

    min-height: 100vh;

    padding: 40px 20px;

}

.navigation-tabs {

    display: flex;

    justify-content: center;

    gap: 15px;

    margin-bottom: 30px;

    flex-wrap: wrap;

}

.tab-button {

    background-color: #ffffff;

    border: 2px solid #4CAF50;

    padding: 10px 20px;

    border-radius: 25px;

    font-weight: 500;

    color: #4CAF50;

    cursor: pointer;

    transition: all 0.3s ease;

}

.tab-button*:hover* {

    background-color: #4CAF50;

    color: #fff;

}

.tab-button.active {

    background-color: #4CAF50;

    color: #fff;

}

.component-display {

    max-width: 1000px;

    margin: 0 auto;

    background-color: #fff;

    border-radius: 12px;

    padding: 30px;

    box-shadow: 0 4px 20px rgba(0, 0, 0, 0.06);

}

**imageCarousel.jsx**

import React from 'react';

import Slider from 'react-slick';

import "slick-carousel/slick/slick.css";

import "slick-carousel/slick/slick-theme.css";

import img1 from '../assets/images/img1.jpg';

import img2 from '../assets/images/img2.jpg';

import img3 from '../assets/images/img3.jpg';

const ImageCarousel = () => {

    const settings = {

        dots: **true**,

        infinite: **true**,

        speed: 500,

        slidesToScroll: 1,

        autoplay: **true**,

        autoplaySpeed: 3000,

        slidesToShow: 1,

        appendDots: (*dots*) => (

            <div

                style={{

                    display: 'flex',

                    justifyContent: 'center',

                    position: 'absolute',

                    bottom: '10px',

                    left: 0,

                    right: 0,

                }}

            >

                {dots}

            </div>

        ),

    };

    const images = [

        img1,

        img2,

        img3,

    ];

    return (

        <div>

            <h2>Image Carousel</h2>

            <div style={{ width: '500px', margin: 'auto' }}>

                <Slider {...settings}>

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

                        <div key={index} style={{ display: 'flex', justifyContent: 'center' }}>

                            <img src={image} alt={`Slide ${index}`} style={{ width: '500px', height: '300px' }} />

                        </div>

                    ))}

                </Slider>

            </div>

        </div>

    );

};

export default ImageCarousel;

**Components/** **FaqAccordion.jsx**

import React, { useState, useRef, useEffect } from "react";

import { FiPlus } from "react-icons/fi";

import '../assets/css/FaqAccordion.css';

const questions = [

    {

        title: 'How do I sign up for an account?',

        content: 'To sign up for an account, click on the "Sign Up" button at the top right corner of the homepage. Fill out the registration form with your details and click on the "Sign Up" button to create your account.',

    },

    {

        title: 'How do I reset my password?',

        content: 'If you forgot your password, click on the "Forgot Password" link on the login page. Enter your email address and click on the "Reset Password" button. You will receive an email with instructions on how to reset your password.',

    },

    {

        title: 'How do I make a purchase?',

        content: 'To make a purchase, click on the "Add to Cart" button next to the product you want to buy. Once you have added all the products to your cart, click on the "Cart" icon at the top right corner of the page. Review your cart and click on the "Checkout" button to proceed to the payment page. Enter your payment details and click on the "Place Order" button to complete the purchase.',

    },

    {

        title: 'How do I track my order?',

        content: 'To track your order, click on the "Track Order" link at the top right corner of the homepage. Enter your order number and email address and click on the "Track" button to view the status of your order.',

    },

    {

        title: 'How do I return a product?',

        content: 'To return a product, click on the "Return" link at the bottom of the homepage. Fill out the return form with your order number, email address, and reason for return. Click on the "Submit" button to initiate the return process. You will receive an email with instructions on how to return the product.',

    },

];

const FaqAccordion = () => {

    const [active, setActive] = useState([]);

    const toggleAccordion = (*index*) => {

        if (active.includes(index)) {

            setActive(active.filter((*i*) => i !== index));

        } else {

            setActive([...active, index]);

        }

    };

    const contentRefs = useRef([]);

    useEffect(() => {

        contentRefs.current.forEach((*ref*, *index*) => {

            ref.style.maxHeight = active.includes(index)

                ? `${ref.scrollHeight}px`

                : "0px";

        });

    }, [active]);

    return (

        <div>

            <h2>FAQ/Accordion</h2>

            <div className="faq-section">

                {questions.map((*question*, *index*) => (

                    <button

                        className={`question-section ${active.includes(index) ? 'active' : ''}`}

                        key={index}

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

                    >

                        <div className="question-align">

                            <h4 className="question-style">{question.title}</h4>

                            <FiPlus

                                className={active.includes(index) ? `question-icon rotate` : `question-icon`}

                            />

                        </div>

                        <div

                            ref={(*ref*) => (contentRefs.current[index] = ref)}

                            className={active.includes(index) ? `answer answer-divider` : `answer`}

                        >

                            <p>{question.content}</p>

                        </div>

                    </button>

                ))}

            </div>

        </div>

    );

};

export default FaqAccordion;

**css/** **FaqAccordion.css**

.faq-section {

    max-width: 800px;

    margin: 0 auto;

    padding: 40px 20px;

    background-color: #fff;

    border-radius: 12px;

    box-shadow: 0 4px 20px rgba(0, 0, 0, 0.08);

    font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;

}

h2 {

    text-align: center;

    font-size: 2rem;

    font-weight: 600;

    margin-bottom: 30px;

    color: #333;

}

.question-section {

    width: 100%;

    border: none;

    background-color: #f9f9f9;

    border-radius: 8px;

    margin-bottom: 15px;

    text-align: left;

    padding: 20px;

    transition: all 0.3s ease;

    cursor: pointer;

    overflow: hidden;

}

.question-section*:hover* {

    background-color: #f1f1f1;

}

.question-align {

    display: flex;

    justify-content: space-between;

    align-items: center;

}

.question-style {

    font-size: 1.1rem;

    font-weight: 500;

    color: #222;

    margin: 0;

}

.question-icon {

    font-size: 1.5rem;

    color: #4CAF50;

    transition: transform 0.3s ease;

}

.rotate {

    transform: rotate(45deg); */\* Turning + into x \*/*

}

.answer {

    max-height: 0;

    overflow: hidden;

    transition: max-height 0.3s ease, padding 0.3s ease;

    padding-left: 2px;

    padding-right: 2px;

}

.answer-divider {

    margin-top: 10px;

    padding-top: 10px;

    border-top: 1px solid #ddd;

}

.answer p {

    font-size: 0.95rem;

    color: #555;

    line-height: 1.5;

    margin: 0;

}

**Components/** **ShoppingList.jsx**

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

import '../assets/css/ShoppingList.css';

import productimg from "../assets/product.jpg"

const ShoppingList = () => {

    const [products, setProducts] = useState([

        { id: 1, name: 'iPhone 12', price: 74999, category: 'Smartphones' },

        { id: 2, name: 'Samsung Galaxy S21', price: 59999, category: 'Smartphones' },

        { id: 3, name: 'MacBook Pro 16-inch', price: 179999, category: 'Laptops' },

        { id: 4, name: 'Dell XPS 15', price: 134999, category: 'Laptops' },

        { id: 5, name: 'Canon EOS 5D Mark IV', price: 187499, category: 'Cameras' },

        { id: 6, name: 'Sony PlayStation 5', price: 39999, category: 'Gaming Consoles' },

        { id: 7, name: 'Nintendo Switch', price: 23999, category: 'Gaming Consoles' },

        { id: 8, name: 'Samsung QLED 4K TV', price: 112499, category: 'Televisions' },

        { id: 9, name: 'Bose Noise Cancelling Headphones 700', price: 28499, category: 'Headphones' },

        { id: 10, name: 'Fitbit Charge 4', price: 11249, category: 'Wearable Technology' }

    ]);

    const categories = [...(new Set(products.map(*product* => product.category))).values()];

    const priceRanges = [

        { min: 0, max: 1000 },

        { min: 1001, max: 20000 },

        { min: 20001, max: 30000 },

        { min: 30001, max: 40000 },

        { min: 40001, max: 50000 },

    ];

    useEffect(() => {

        const updatedProducts = products.map((*product*) => {

            const priceIndex = priceRanges.findIndex(*priceRange* => product.price >= priceRange.min && product.price <= priceRange.max);

            return { ...product, 'index': priceIndex };

        }, []);

        setProducts(updatedProducts);

    }, [])

    const [selectedCategories, setSelectedCategories] = useState([]);

    const [selectedPriceRange, setSelectedPriceRange] = useState([]);

    const filteredProducts = products.filter((*product*) =>

        (selectedCategories.length === 0 && selectedPriceRange.length === 0) ||

        (selectedCategories.length !== 0 && selectedPriceRange.length === 0 && selectedCategories.includes(product.category)) ||

        (selectedCategories.length === 0 && selectedPriceRange.length !== 0 && selectedPriceRange.includes(product.index)) ||

        (selectedCategories.length !== 0 && selectedPriceRange.length !== 0 && selectedCategories.includes(product.category) && selectedPriceRange.includes(product.index))

    );

    const handleCategoryChange = (*event*) => {

        const category = event.target.value;

        if (selectedCategories.includes(category)) {

            setSelectedCategories(selectedCategories.filter((*c*) => c !== category));

        } else {

            setSelectedCategories([...selectedCategories, category]);

        }

    };

    const handlePriceRangeChange = (*event*) => {

        const index = parseInt(event.target.value);

        if (selectedPriceRange.includes(index)) {

            setSelectedPriceRange(selectedPriceRange.filter((*c*) => c !== index));

        } else {

            setSelectedPriceRange([...selectedPriceRange, index]);

        }

    };

    return (

        <div>

            <h2>Shopping item list with filters</h2>

            <div className="shopping-list">

                <div className="filters">

                    <div className="categories">

                        <h3>Categories:</h3>

                        {categories.map((*category*) => (

                            <div key={category}>

                                <label htmlFor={`category-${category.replace(/\s+/g, '-')}`}>

                                    <input

                                        type="checkbox"

                                        value={category}

                                        id={`category-${category.replace(/\s+/g, '-')}`}

                                        checked={selectedCategories.includes(category)}

                                        onChange={handleCategoryChange}

                                    />

                                    {category}

                                </label>

                            </div>

                        ))}

                    </div>

                    <div className="price-range">

                        <h3>Price Range:</h3>

                        {priceRanges.map((*priceRange*, *index*) => (

                            <div key={`$${priceRange.min}-$${priceRange.max}`}>

                                <label htmlFor={`price-range-${priceRange.min}-$${priceRange.max}`}>

                                    <input

                                        type="checkbox"

                                        value={index.toString()}

                                        id={`price-range-$${priceRange.min}-$${priceRange.max}`}

                                        checked={selectedPriceRange.includes(index)}

                                        onChange={handlePriceRangeChange}

                                    />

                                    {`Rs.${priceRange.min}-Rs.${priceRange.max}`}

                                </label>

                            </div>

                        ))}

                    </div>

                </div>

                <div className="product-cards">

                    {filteredProducts.map((*product*) => (

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

                            <img src={productimg} alt={"Product Image"} />

                            <div className="product-details">

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

                                <div className="row">

                                    <p>{product.category}</p>

                                </div>

                                <p>{`Rs.${product.price.toLocaleString()}`}</p>

                            </div>

                        </div>

                    ))}

                </div>

            </div>

        </div>

    );

};

export default ShoppingList;

**css/** **ShoppingList.css**

*/\* ShoppingList.css \*/*

.shopping-list {

    display: flex;

    flex-direction: column;

    gap: 30px;

    max-width: 1200px;

    margin: 0 auto;

    padding: 30px 20px;

    background-color: #fff;

    border-radius: 12px;

    box-shadow: 0 4px 20px rgba(0, 0, 0, 0.1);

}

.filters {

    display: flex;

    gap: 30px;

    flex-wrap: wrap;

    justify-content: space-between;

}

.categories, .price-range {

    width: 45%;

    min-width: 250px;

}

h3 {

    font-size: 1.2em;

    font-weight: 600;

    color: #333;

    margin-bottom: 10px;

}

label {

    font-size: 1em;

    color: #666;

    display: flex;

    align-items: center;

    margin: 10px 0;

}

input[type="checkbox"] {

    margin-right: 10px;

    width: 18px;

    height: 18px;

    border-radius: 4px;

    transition: all 0.3s ease;

}

input[type="checkbox"]*:checked* {

    background-color: #4CAF50;

    border: none;

}

.product-cards {

    display: grid;

    grid-template-columns: repeat(auto-fill, minmax(240px, 1fr));

    gap: 20px;

}

.product-card {

    background-color: #fff;

    border-radius: 12px;

    box-shadow: 0 4px 15px rgba(0, 0, 0, 0.1);

    overflow: hidden;

    transition: transform 0.3s ease;

}

.product-card*:hover* {

    transform: translateY(-5px);

}

.product-card img {

    width: 100%;

    height: 200px;

    object-fit: cover;

}

.product-details {

    padding: 15px;

}

.product-details h3 {

    font-size: 1.1em;

    font-weight: 500;

    color: #333;

    margin: 0;

}

.product-details .row {

    font-size: 0.9em;

    color: #999;

}

.product-details p {

    font-size: 1.2em;

    font-weight: 600;

    color: #4CAF50;

    margin-top: 10px;

}

@media (max-width: 768px) {

    .filters {

        flex-direction: column;

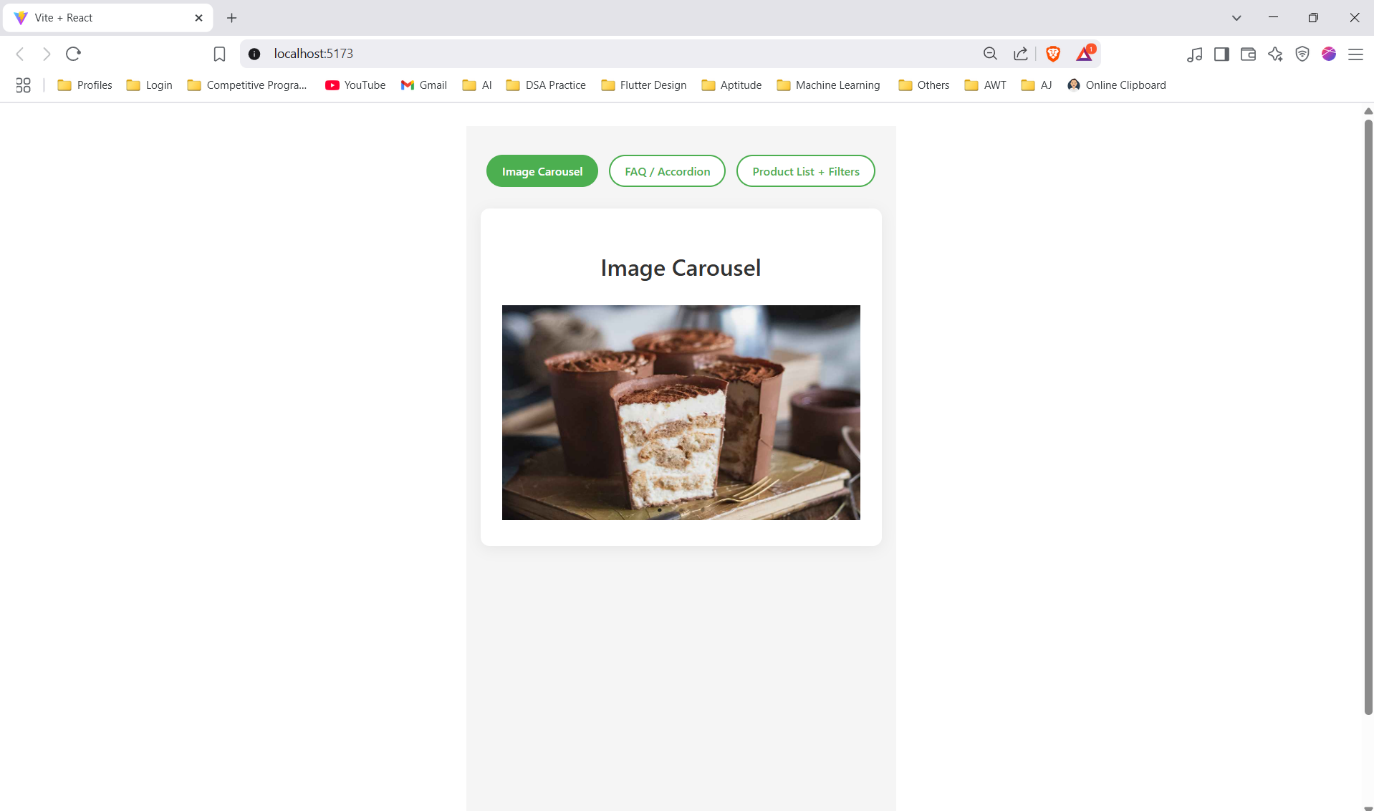
    }

    .categories, .price-range {

        width: 100%;

    }

}

**Output :-**

