Experiment N0 8

Code:

// PWA functionality

let deferredPrompt;

const installBanner = document.getElementById('install-banner');

const installButton = document.getElementById('install-button');

const dismissButton = document.getElementById('dismiss-button');

// Check if service worker is supported

if ('serviceWorker' in navigator) {

    window.addEventListener('load', () => {

        navigator.serviceWorker.register('service-worker.js')

            .then(registration => {

                console.log('Service Worker registered with scope:', registration.scope);

            })

            .catch(error => {

                console.error('Service Worker registration failed:', error);

            });

    });

}

// Listen for the beforeinstallprompt event

window.addEventListener('beforeinstallprompt', (e) => {

    // Prevent Chrome 67 and earlier from automatically showing the prompt

    e.preventDefault();

    // Stash the event so it can be triggered later

    deferredPrompt = e;

    // Show the install banner

    installBanner.classList.remove('hidden');

});

// Install button click handler

installButton.addEventListener('click', () => {

    // Hide the install banner

    installBanner.classList.add('hidden');

    // Show the install prompt

    if (deferredPrompt) {

        deferredPrompt.prompt();

        // Wait for the user to respond to the prompt

        deferredPrompt.userChoice.then((choiceResult) => {

            if (choiceResult.outcome === 'accepted') {

                console.log('User accepted the install prompt');

            } else {

                console.log('User dismissed the install prompt');

            }

            // Clear the deferredPrompt variable

            deferredPrompt = null;

        });

    }

});

// Dismiss button click handler

dismissButton.addEventListener('click', () => {

    installBanner.classList.add('hidden');

});

// E-commerce functionality

const cartCount = document.querySelector('.cart-count');

const addToCartButtons = document.querySelectorAll('.add-to-cart');

let cartItems = 0;

// Add to cart functionality

addToCartButtons.forEach(button => {

    button.addEventListener('click', () => {

        cartItems++;

        cartCount.textContent = cartItems;

        // Animate button

        button.textContent = 'Added!';

        button.style.backgroundColor = '#4caf50';

        setTimeout(() => {

            button.textContent = 'Add to Cart';

            button.style.backgroundColor = '#4285f4';

        }, 1000);

    });

});

// Service Worker installation and activation

self.addEventListener('install', (event) => {

    console.log('Service Worker installing.');

    // Perform install steps

    event.waitUntil(

        caches.open('ecommerce-cache-v1')

            .then((cache) => {

                console.log('Opened cache');

                return cache.addAll([

                    '/',

                    '/index.html',

                    '/styles.css',

                    '/script.js',

                    '/images/logo.png',

                    '/images/shop\_icon.png',

                    // Add other assets to cache

                ]);

            })

    );

});

self.addEventListener('activate', (event) => {

    console.log('Service Worker activating.');

    const cacheWhitelist = ['ecommerce-cache-v1'];

    event.waitUntil(

        caches.keys().then((cacheNames) => {

            return Promise.all(

                cacheNames.map((cacheName) => {

                    if (cacheWhitelist.indexOf(cacheName) === -1) {

                        return caches.delete(cacheName);

                    }

                })

            );

        })

    );

});

self.addEventListener('fetch', (event) => {

    event.respondWith(

        caches.match(event.request)

            .then((response) => {

                // Cache hit - return response

                if (response) {

                    return response;

                }

                return fetch(event.request);

            })

    );

});









