Experiment No 9

**Code:**

const CACHE\_NAME = "pwa-cache-v1";

const ASSETS\_TO\_CACHE = [

"/PWA/",

"/PWA/index.html",

"/PWA/styles.css",

"/PWA/app.js",

// Add other assets to cache

];

// Install event - Cache important assets

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

console.log("✅ Service Worker Installed");

event.waitUntil(

caches.open(CACHE\_NAME).then((cache) => {

console.log("✅ Caching assets");

return cache.addAll(ASSETS\_TO\_CACHE);

})

);

});

// Activate event - Cleanup old caches

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

console.log("✅ Service Worker Activated");

event.waitUntil(

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

Promise.all(

cacheNames.map((cache) => {

if (cache !== CACHE\_NAME) {

console.log("🗑️ Deleting old cache:", cache);

return caches.delete(cache);

}

})

)

)

);

});

// Fetch event - Serve cached files when offline

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

console.log("🔄 Fetching:", event.request.url);

event.respondWith(

caches

.match(event.request)

.then((response) => response || fetch(event.request))

.catch(() => caches.match("/PWA/index.html")) // Fallback to index.html if offline

);

});

// Sync event - Background sync for failed requests

self.addEventListener("sync", (event) => {

if (event.tag === "sync-cart") {

event.waitUntil(syncCart());

}

});

async function syncCart() {

// Logic to sync cart data with the server

console.log("🔄 Syncing cart data with the server");

// Example: Fetch unsynced cart data from IndexedDB and send to server

}

// Push event - Handle push notifications

self.addEventListener("push", (event) => {

const data = event.data.json();

console.log("🔔 Push notification received:", data);

const options = {

body: data.body,

icon: "/PWA/icons/icon-192x192.png",

badge: "/PWA/icons/icon-192x192.png",

};

event.waitUntil(

self.registration.showNotification(data.title, options)

);

});









