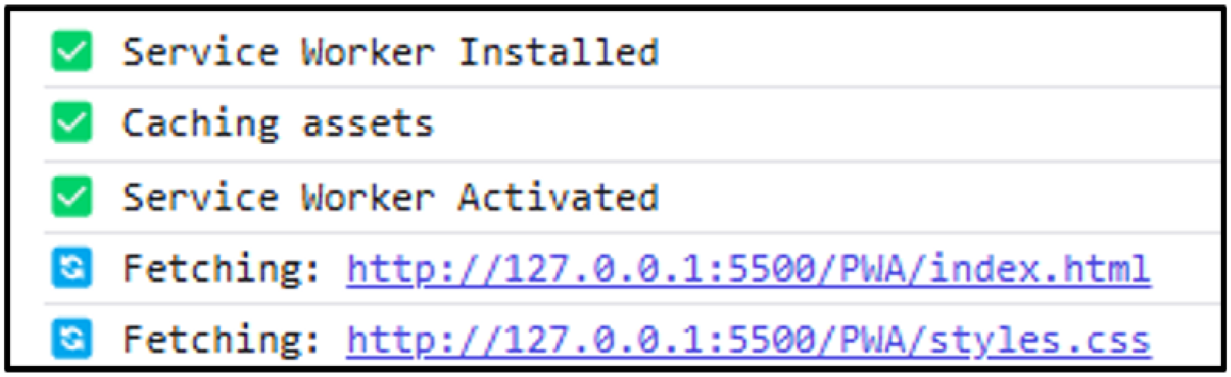
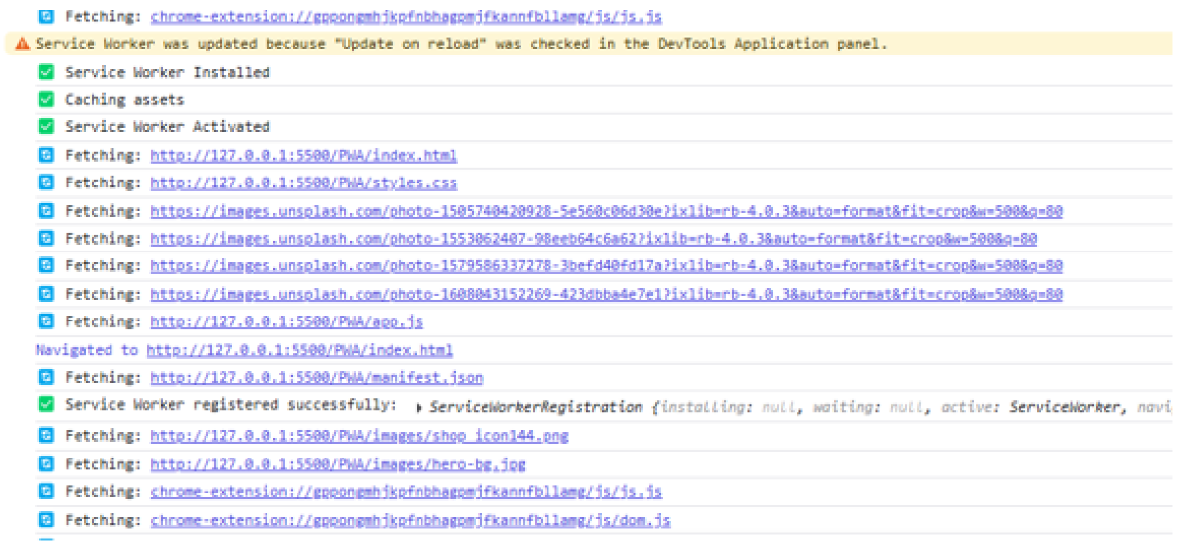
EXPERIMENT NO-9

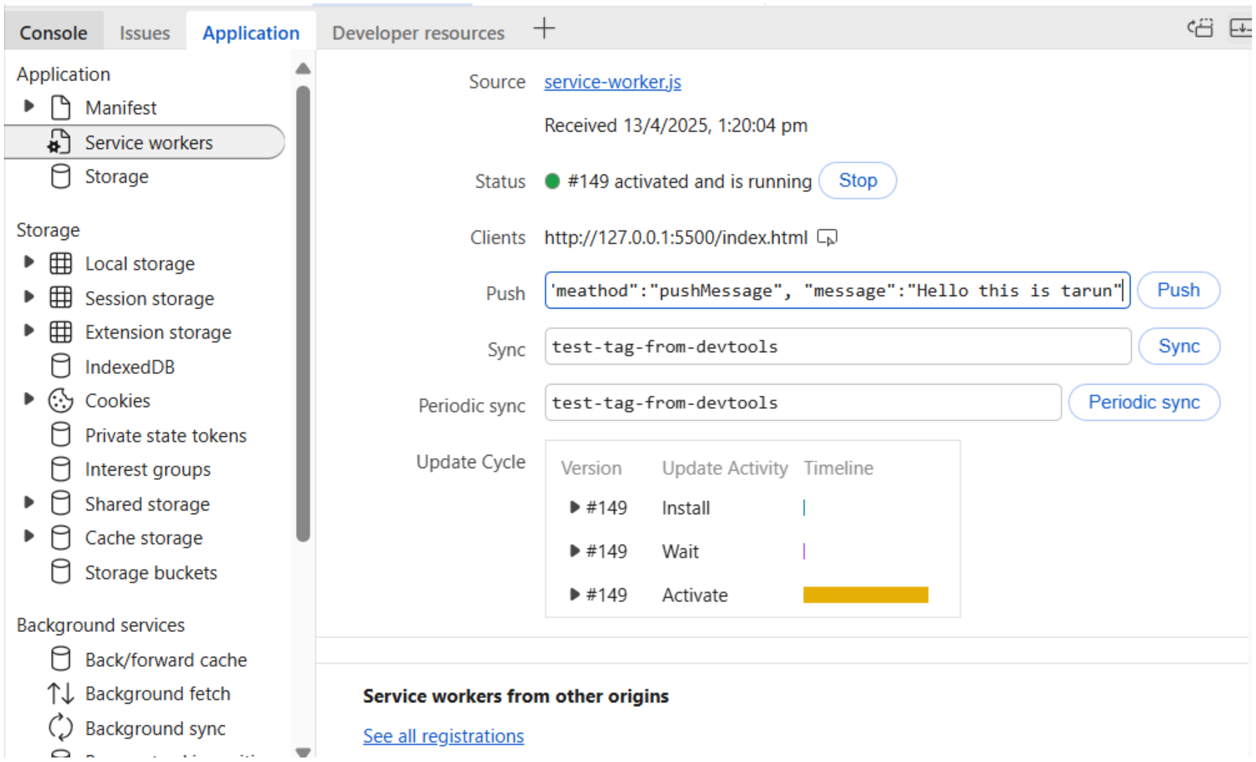
```
// Change this to a new version to trigger an update
const CACHE_NAME = "pwa-cache-v2";

const ASSETS_TO_CACHE = [
  "./",
  "./index.html",
  "./styles.css",
  "./app.js",
  "./manifest.json",
  "./images/shop_icon.png",
];

// Install event - Cache important assets
self.addEventListener("install", (event) => {
  console.log("✅ Service Worker Installed");

  // This line forces the waiting service worker to become active
  self.skipWaiting();

  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");

  // This ensures the service worker takes control immediately
  event.waitUntil(clients.claim());

  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 - Modified to always check network first for HTML files
self.addEventListener("fetch", (event) => {
  const url = new URL(event.request.url);

  // Always get fresh HTML from network
  if (
          event.request.mode === "navigate" ||
          (event.request.method === "GET" &&
          event.request.headers.get("accept").includes("text/html"))
  ) {
          console.log("🔄 Getting fresh HTML for:", url.pathname);

          event.respondWith(
          fetch(event.request).catch(() => caches.match("./index.html"))
          );
  } else {
          // For non-HTML resources, use cache-first strategy
          event.respondWith(
          caches
          .match(event.request)
          .then((response) => {
          return (
          response ||
          fetch(event.request).then((fetchResponse) => {
          // Save new resources in cache
          const responseClone = fetchResponse.clone();
          caches
              .open(CACHE_NAME)
              .then((cache) => cache.put(event.request, responseClone));
          return fetchResponse;
          })
          );
          })
          .catch(() => {
          // Fallback for offline
          if (event.request.url.includes(".html")) {
          return caches.match("./index.html");
          }
          })
          );
  }
});

// Rest of your code remains the same
self.addEventListener("sync", (event) => {
  if (event.tag === "sync-cart") {
          event.waitUntil(syncCart());
  }
});

self.addEventListener("push", (event) => {
  const options = {
          body: "This is a push notification!",
          icon: "/images/shop_icon.png",
          badge: "/images/badge.png",
  };
  event.waitUntil(self.registration.showNotification("New Message!", options));
});

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
          ? event.data.json()
          : { title: "New Notification", body: "Check out our latest products!" };
  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));
});
```



```
✅ Service Worker Installed
✅ Caching assets
✅ Service Worker Activated
🔄 Fetching: http://127.0.0.1:5500/PWA/index.html
🔄 Fetching: http://127.0.0.1:5500/PWA/styles.css
```



```
🔄 Fetching: chrome-extension://gppongmhjkpfnbhagpmjfkannfbllamg/js/js.js
⚠ Service Worker was updated because "Update on reload" was checked in the DevTools Application panel.
✅ Service Worker Installed
✅ Caching assets
✅ Service Worker Activated
🔄 Fetching: http://127.0.0.1:5500/PWA/index.html
🔄 Fetching: http://127.0.0.1:5500/PWA/styles.css
🔄 Fetching: https://images.unsplash.com/photo-1505740420928-5e560c06d30e?ixlib=rb-4.0.3&auto=format&fit=crop&w=500&q=80
🔄 Fetching: https://images.unsplash.com/photo-1553062407-98eeb64c6a62?ixlib=rb-4.0.3&auto=format&fit=crop&w=500&q=80
🔄 Fetching: https://images.unsplash.com/photo-1579586337278-3befd40fd17a?ixlib=rb-4.0.3&auto=format&fit=crop&w=500&q=80
🔄 Fetching: https://images.unsplash.com/photo-1608043152269-423dbba4e7e1?ixlib=rb-4.0.3&auto=format&fit=crop&w=500&q=80
🔄 Fetching: http://127.0.0.1:5500/PWA/app.js
Navigated to http://127.0.0.1:5500/PWA/index.html
🔄 Fetching: http://127.0.0.1:5500/PWA/manifest.json
✅ Service Worker registered successfully:  ▸ ServiceWorkerRegistration {installing: null, waiting: null, active: ServiceWorker, navi
🔄 Fetching: http://127.0.0.1:5500/PWA/images/shop_icon144.png
🔄 Fetching: http://127.0.0.1:5500/PWA/images/hero-bg.jpg
🔄 Fetching: chrome-extension://gppongmhjkpfnbhagpmjfkannfbllamg/js/js.js
🔄 Fetching: chrome-extension://gppongmhjkpfnbhagpmjfkannfbllamg/js/dom.js
```



Console | Issues | Application | Developer resources

Application
- Manifest
- Service workers
- Storage

Storage
- Local storage
- Session storage
- Extension storage
- IndexedDB
- Cookies
- Private state tokens
- Interest groups
- Shared storage
- Cache storage
- Storage buckets

Background services
- Back/forward cache
- Background fetch
- Background sync

Source: service-worker.js
Received 13/4/2025, 1:20:04 pm

Status: #149 activated and is running [Stop]

Clients: http://127.0.0.1:5500/index.html

Push: `'meathod":"pushMessage", "message":"Hello this is tarun"` [Push]

Sync: `test-tag-from-devtools` [Sync]

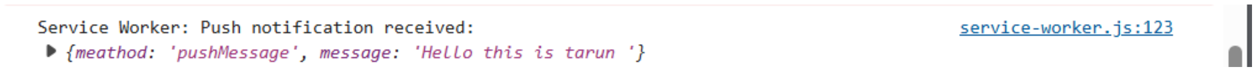
Periodic sync: `test-tag-from-devtools` [Periodic sync]

Update Cycle

Version	Update Activity	Timeline
#149	Install	
#149	Wait	
#149	Activate	

Service workers from other origins

See all registrations



```
Service Worker: Push notification received:                    service-worker.js:123
▶ {meathod: 'pushMessage', message: 'Hello this is tarun '}
```