Veydant Sharma D15C 50

EXPERIMENT 9

Aim: To implement Service worker events like fetch, sync and push for E-commerce PWA.

Theory:
Service Workers are a powerful feature of Progressive Web Apps (PWAs) that enable advanced capabilities like background synchronization, push notifications, and offline caching. These capabilities enhance user experience by allowing apps to work seamlessly even in low or no-network conditions. The primary events associated with Service Workers include:

Install Event: This event is triggered when the service worker is first installed on the client. During this event, we typically cache the essential files required for offline functionality, like HTML, CSS, JavaScript files, and assets.

Activate Event: The activate event is fired when the service worker is activated. This is a good place to clean up old caches that are no longer needed. It ensures that only the latest version of the cache is used and old caches are deleted.

Fetch Event: The fetch event allows the Service Worker to intercept network requests made by the app. This event is crucial for enabling offline functionality and caching assets so that they can be served when the network is unavailable.

Sync Event (Background Sync): The sync event is a powerful feature that allows the app to defer actions like sending data to the server when the network becomes available. The sync event is useful for tasks like sending order data or other important information to the server in the background, even after the user has closed the app.

Push Event: Push notifications are an essential feature for PWAs. This event is fired when a push notification is received, and it allows the Service Worker to display notifications to the user, even when the app is not open. Push notifications can be triggered from the server and shown to the user as alerts.

Veydant Sharma D15C 50

Code:

sw.js

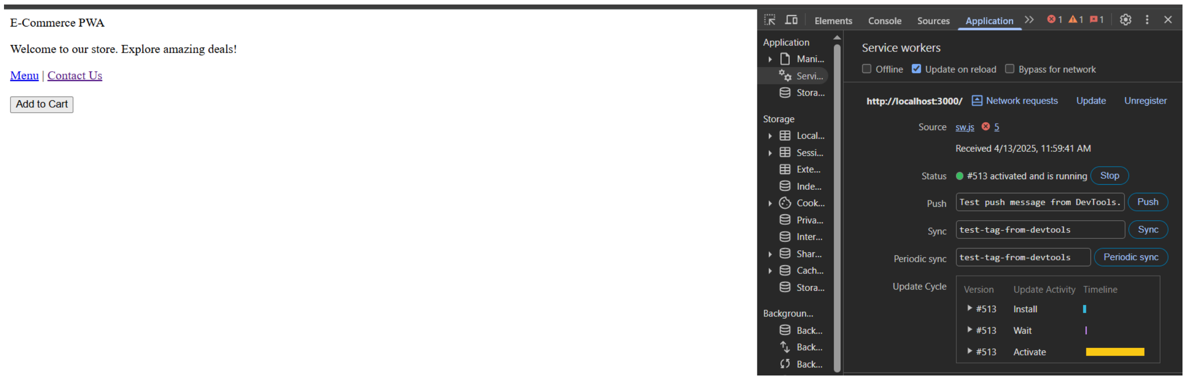
```
const CACHE_NAME = "ecommerce-cache-v1";
const FILES_TO_CACHE = [
  "/",
  "/index.html",
  "/styles.css",
  "/main.js",
  "/offline.html"
];

// INSTALL Event
self.addEventListener("install", event => {
  event.waitUntil(
    caches.open(CACHE_NAME).then(cache => {
      console.log("[Service Worker] Caching app shell");
      return cache.addAll(FILES_TO_CACHE);
    })
  );
  self.skipWaiting();
});

// ACTIVATE Event
self.addEventListener("activate", event => {
  event.waitUntil(
    caches.keys().then(keyList => {
      return Promise.all(
        keyList.map(key => {
          if (key !== CACHE_NAME) return caches.delete(key);
        })
      );
    })
  );
  self.clients.claim();
});

// FETCH Event
self.addEventListener("fetch", event => {
  event.respondWith(
    caches.match(event.request).then(response => {
      if (response) {
        console.log("[Service Worker] Fetched from cache:", event.request.url);
        return response;
      }
      return fetch(event.request).then(networkResponse => {
        if (event.request.url.indexOf(".html") !== -1) {
          caches.open(CACHE_NAME).then(cache => cache.put(event.request,
networkResponse.clone()));
        }
        return networkResponse;
      }).catch(() => caches.match("/offline.html"));
    })
  );
});

// SYNC Event (Background Sync)
self.addEventListener("sync", event => {
  if (event.tag === "sync-order") {
    console.log("[Service Worker] Syncing order data...");
    event.waitUntil(
      self.registration.showNotification("Sync Successful", {
        body: "Order data synced successfully!",
        icon: "https://picsum.photos/128"
      })
    );
  }
});

// PUSH Event
self.addEventListener("push", event => {
  const data = event.data ? event.data.text() : "New Notification";
  console.log("[Service Worker] Push event received:", data);

  const options = {
    body: data,
    icon: "https://picsum.photos/128"
  };

  event.waitUntil(
    self.registration.showNotification("ShopEase", options)
```

```
      .catch(error => console.error("Error showing notification:", error))
  );
});
```

main.js

```
if ('serviceWorker' in navigator && 'SyncManager' in window) {
    window.addEventListener('load', () => {
      navigator.serviceWorker.register('sw.js')
        .then(reg => {
          console.log('Service Worker registered!', reg);

          // Register sync event if sync is supported
          if ('sync' in reg) {
            reg.sync.register('sync-order') // Register background sync with tag 'sync-order'
              .then(() => {
                console.log('Sync event registered');
              })
              .catch(err => {
                console.error('Sync registration failed:', err);
              });
          }
        })
        .catch(err => {
          console.error('Service Worker registration failed:', err);
        });
    });
  } else {
    console.error('Service Worker or SyncManager not supported.');
  }
```

Output:



E-Commerce PWA

Welcome to our store. Explore amazing deals!

Menu | Contact Us

Add to Cart

Service workers

Offline [✓ Update on reload] Bypass for network

http://localhost:3000/   Network requests   Update   Unregister

Source sw.js ✖ 5

Received 4/13/2025, 11:59:41 AM

Status ● #513 activated and is running Stop

Push `Test push message from DevTools.` Push

Sync `test-tag-from-devtools` Sync

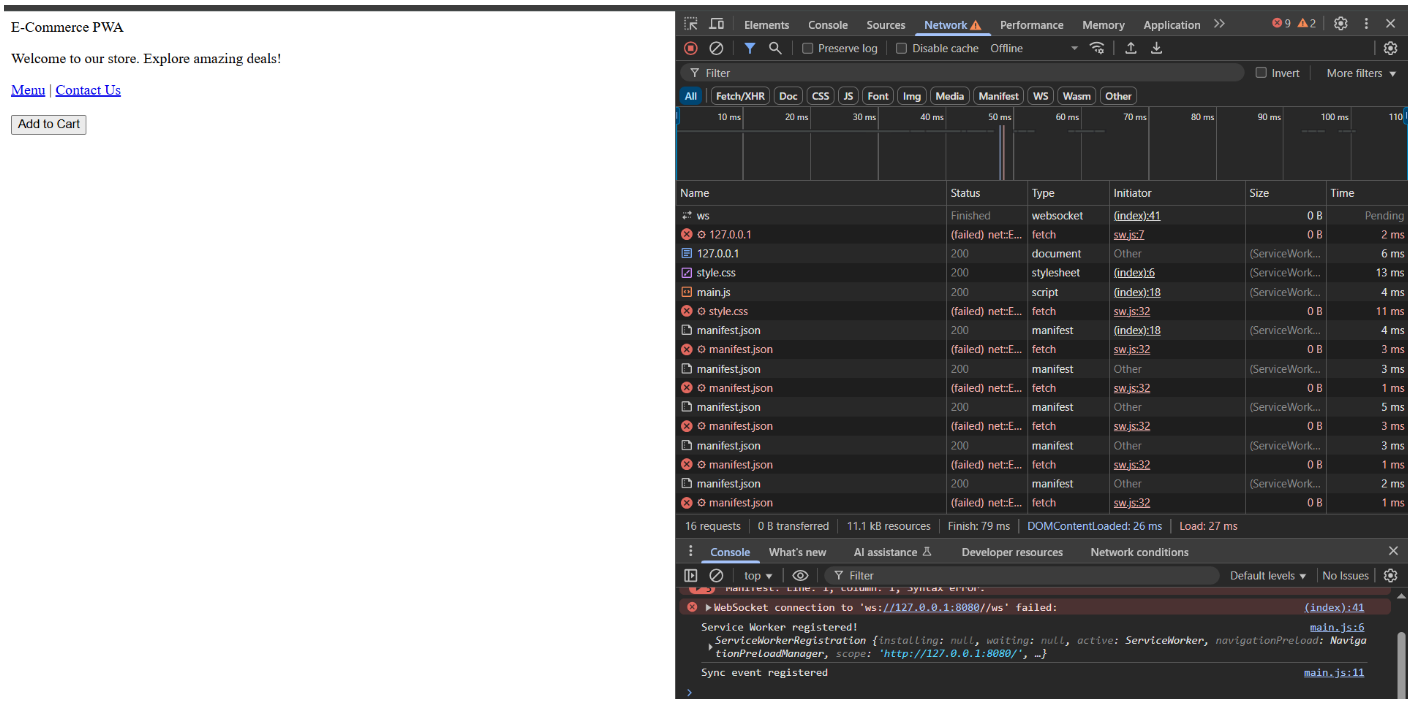
Periodic sync `test-tag-from-devtools` Periodic sync

Update Cycle

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

Veydant Sharma   D15C   50

Sync event


E-Commerce PWA

Welcome to our store. Explore amazing deals!

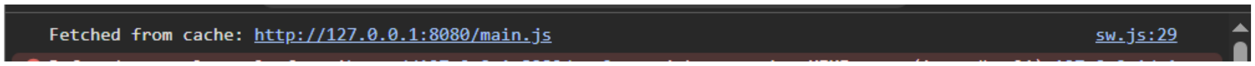
Menu | Contact Us

Add to Cart

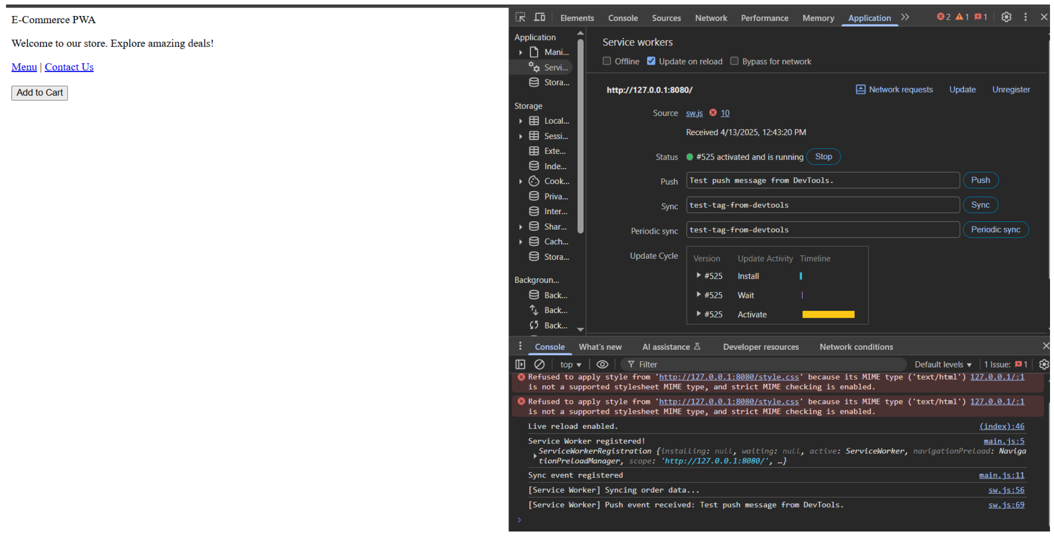
```
Name               Status            Type        Initiator    Size            Time
ws                 Finished          websocket   (index):41   0 B             Pending
127.0.0.1          (failed) net::E...fetch       sw.js:7      0 B             2 ms
127.0.0.1          200               document    Other        (ServiceWork... 6 ms
style.css          200               stylesheet  (index):6    (ServiceWork... 13 ms
main.js            200               script      (index):18   (ServiceWork... 4 ms
style.css          (failed) net::E...fetch       sw.js:32     0 B             11 ms
manifest.json      200               manifest    (index):18   (ServiceWork... 4 ms
manifest.json      (failed) net::E...fetch       sw.js:32     0 B             3 ms
manifest.json      200               manifest    Other        (ServiceWork... 3 ms
manifest.json      (failed) net::E...fetch       sw.js:32     0 B             1 ms
manifest.json      200               manifest    Other        (ServiceWork... 5 ms
manifest.json      (failed) net::E...fetch       sw.js:32     0 B             3 ms
manifest.json      200               manifest    Other        (ServiceWork... 3 ms
manifest.json      (failed) net::E...fetch       sw.js:32     0 B             1 ms
manifest.json      200               manifest    Other        (ServiceWork... 2 ms
manifest.json      (failed) net::E...fetch       sw.js:32     0 B             1 ms

16 requests | 0 B transferred | 11.1 kB resources | Finish: 79 ms | DOMContentLoaded: 26 ms | Load: 27 ms
```

```
WebSocket connection to 'ws://127.0.0.1:8080//ws' failed:                    (index):41
Service Worker registered!                                                    main.js:6
 ServiceWorkerRegistration {installing: null, waiting: null, active: ServiceWorker, navigationPreload: Naviga
 tionPreloadManager, scope: 'http://127.0.0.1:8080/', …}
Sync event registered                                                         main.js:11
```

Fetch event


```
Fetched from cache: http://127.0.0.1:8080/main.js                             sw.js:29
```

Push Event:


E-Commerce PWA

Welcome to our store. Explore amazing deals!

Menu | Contact Us

Add to Cart

Service workers

Offline [✓ Update on reload] Bypass for network

http://127.0.0.1:8080/

Source sw.js 10

Received 4/13/2025, 12:43:20 PM

Status #525 activated and is running

Push: Test push message from DevTools.

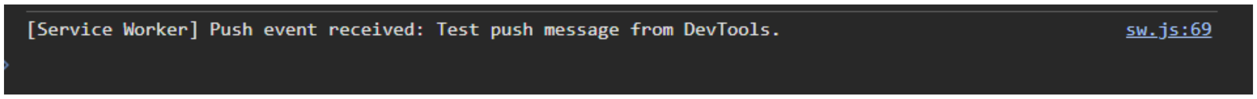
Sync: test-tag-from-devtools

Periodic sync: test-tag-from-devtools

Version	Update Activity
#525	Install
#525	Wait
#525	Activate

```
Refused to apply style from 'http://127.0.0.1:8080/style.css' because its MIME type ('text/html') 127.0.0.1/:1
is not a supported stylesheet MIME type, and strict MIME checking is enabled.
Refused to apply style from 'http://127.0.0.1:8080/style.css' because its MIME type ('text/html') 127.0.0.1/:1
is not a supported stylesheet MIME type, and strict MIME checking is enabled.
Live reload enabled.                                                          (index):46
Service Worker registered!                                                    main.js:5
 ServiceWorkerRegistration {installing: null, waiting: null, active: ServiceWorker, navigationPreload: Naviga
 tionPreloadManager, scope: 'http://127.0.0.1:8080/', …}
Sync event registered                                                         main.js:11
[Service Worker] Syncing order data...                                        sw.js:56
[Service Worker] Push event received: Test push message from DevTools.        sw.js:69
```

Veydant Sharma D15C 50



```
[Service Worker] Push event received: Test push message from DevTools.          sw.js:69
```

Conclusion:
Implementing Service Worker events such as fetch, sync, and push significantly enhances the functionality and user experience of an E-commerce PWA. These events enable offline capabilities, background data synchronization, and real-time push notifications, ensuring that users can interact with the app seamlessly, even in low or no-network conditions. By leveraging these features, PWAs can provide a more reliable, engaging, and responsive experience, making them a powerful tool for modern web development.