Name: Atif Ansari Roll no: 04 Class: D15B

MPL Experiment - 8

Aim: To code and register a service worker, and complete the install and activation process for a new service worker for the ~~E-commerce~~ E-commerce PWA.

Theory: A Service Worker is a script that runs in the background of a web browser, independent of the main webpage, and acts as a proxy between the network and the user's browser. It enables features such as caching, push notifications, background sync, and offline access, making Progressive Web Applications (PWAs) more efficient and reliable.

Characteristics of Service Workers:
- Service workers intercept network requests and can modify or respond to them directly from cache, enhancing offline functionality.
- They run seperately from the main JavaScript thread, meaning they don't directly interact with the Document Object Model (DOM).
- Information persistence is managed using IndexedDB instead of global variables.
- Service workers operate only on secure HTTPS connections to prevent security threats.

Key features of Service Workers:
- Caching: Stores files for offline use.
- Network Request Interception: Controls how requests are handled.

• Push Notifications: Sends updates even when the app is closed.
• Background Sync: Synchronize data when connectivity is restored.

Service Worker Lifecycle:

1. Registration: The browser is informed about the service worker.
2. Installation: Required assets are cached.
3. Activation: The service worker takes control, replacing old versions.

It enhances PWA by improving speed, reducing network dependency, and enabling offline functionality.

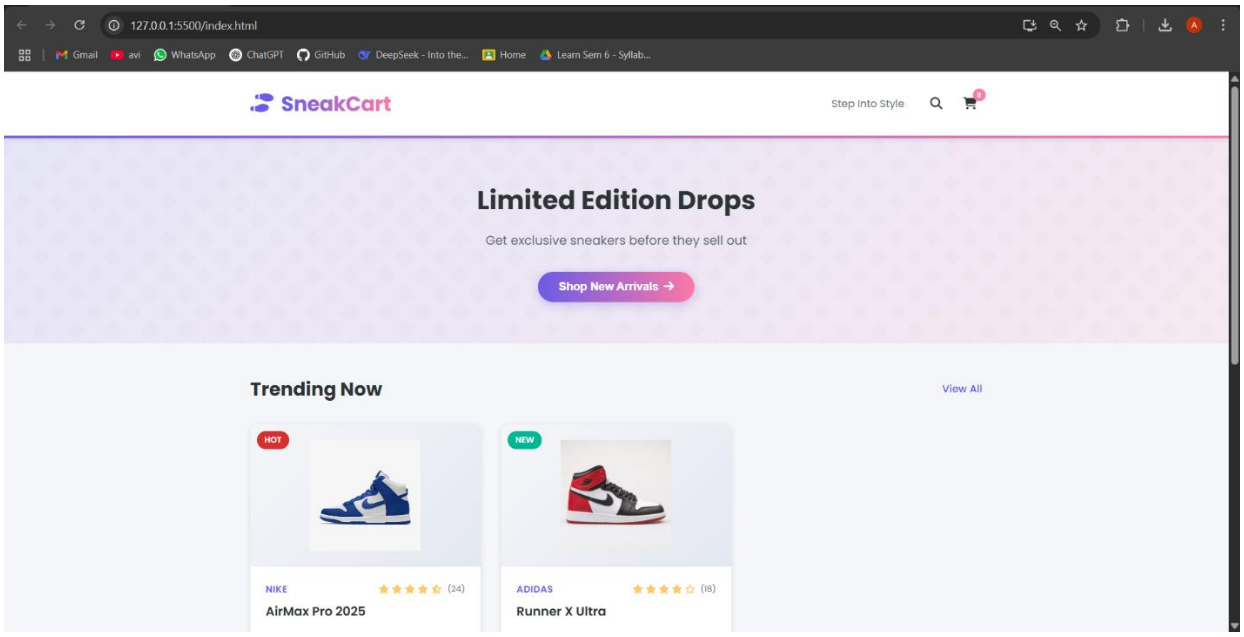
Conclusion:

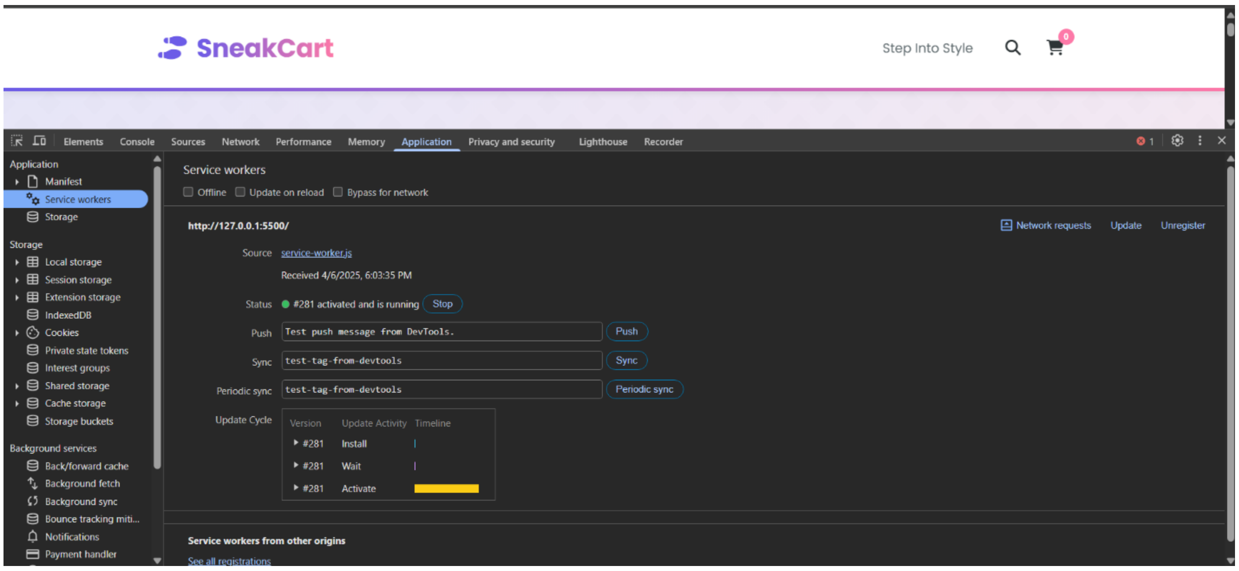
In this experiment, we successfully registered and implemented a service worker, allowing the PWA to function efficiently with caching and offline capabilities. The installation and activation ~~steps~~ steps ensured that essential assets were stored and outdated versions were replaced. This enhances performance, reliability, and user experience by enabling access even in poor network conditions.

Code Implementation of Experiment 8:

Service-worker.js:
```
const CACHE_NAME = "sneakcart-cache-v1";
const urlsToCache = [
  "/",
  "/index.html",
  "/style.css",
  "/script.js",
  "/offline.html",
  "/manifest.json",
  "/products/shoe1.png",
  "/products/shoe2.jpg",
  "/icons/icon-192.png",
  "/icons/icon-512.png"
];

// Install event
self.addEventListener("install", (event) => {
  event.waitUntil(
    caches.open(CACHE_NAME).then((cache) => {
      console.log("[Service Worker] Caching all files");
      return cache.addAll(urlsToCache);
    })
  );
  self.skipWaiting(); // Activate worker immediately
});

// Activate event
self.addEventListener("activate", (event) => {
  event.waitUntil(
    caches.keys().then((cacheNames) => {
      return Promise.all(
        cacheNames.map((name) => {
          if (name !== CACHE_NAME) {
            console.log("[Service Worker] Deleting old cache:", name);
            return caches.delete(name);
          }
        })
      );
    })
  );
  self.clients.claim(); // Claim clients immediately
});

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

```
// Handle page navigations
if (event.request.mode === "navigate") {
  event.respondWith(
    fetch(event.request).catch(() => {
      return caches.match("/offline.html");
    })
  );
} else {
  // Handle other requests (images, CSS, JS, manifest, etc.)
  event.respondWith(
    caches.match(event.request).then((response) => {
      return (
        response ||
        fetch(event.request).catch(() => {
          // Graceful fallback if resource not cached and offline
          return new Response("", {
            status: 503,
            statusText: "Service Unavailable",
          });
        })
      );
    })
  );
}
});
```




SneakCart

Step Into Style

Elements Console Sources Network Performance Memory Application Privacy and security Lighthouse Recorder

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
- Bounce tracking miti...
- Notifications
- Payment handler

Service workers

☐ Offline ☐ Update on reload ☐ Bypass for network

http://127.0.0.1:5500/ Network requests Update Unregister

Source	service-worker.js
	Received 4/6/2025, 6:03:35 PM
Status	● #281 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

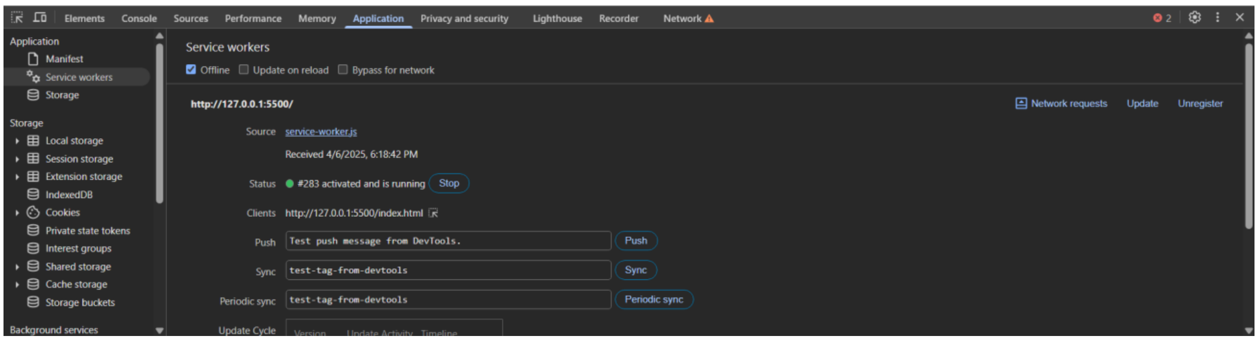
Version	Update Activity	Timeline
#281	Install	
#281	Wait	
#281	Activate	

Service workers from other origins

See all registrations

You're Offline

Please check your internet connection and try again.


Elements Console Sources Performance Memory Application Privacy and security Lighthouse Recorder Network

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

Service workers

☑ Offline ☐ Update on reload ☐ Bypass for network

http://127.0.0.1:5500/ Network requests Update Unregister

Source	service-worker.js
	Received 4/6/2025, 6:18:42 PM
Status	● #283 activated and is running Stop
Clients	http://127.0.0.1:5500/index.html
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