Veydant Sharma D15C 50

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 PWA.

Theory:
A Service Worker is a background script that runs separately from the main browser thread. It allows web applications to intercept and handle network requests, cache resources, and provide offline experiences. Service workers are a core technology of Progressive Web Apps (PWAs), enabling functionalities like offline access, background sync, and push notifications.

Service workers follow a strict life cycle consisting of registration, installation, and activation. They only work on secure origins (HTTPS) or on localhost during development. Once activated, a service worker can control pages and handle network events using the Fetch API.

Key Concepts:

- Registration: This is the process of registering the service worker script (sw.js) with the browser using JavaScript. It tells the browser where to find the service worker and initiates the install phase.

- Installation: During this phase, the service worker installs and typically caches the core resources of the web app using the Cache API. This ensures that the app shell can be loaded quickly and even offline.

- Activation: Once the installation is successful and older service workers are no longer in use, the new service worker gets activated. At this point, it can clean up old caches and take control of the pages within its scope.

- Fetch Event Handling: The service worker can intercept network requests and decide how to respond — either with cached data or by making a network call.

Code:

sw.js (Contains code for service worker)

```
const filesToCache = [
  '/',
  '/menu.html',
  '/contactUs.html',
```

```
  '/offline.html',
  '/style.css',
  '/index.html',
  '/icon.png' // Your icon
];

// Install event - Precache files
self.addEventListener('install', event => {
  event.waitUntil(
    caches.open('ecommerce-cache').then(cache => {
      console.log('Caching files during installation...');
      return cache.addAll(filesToCache);
    })
  );
});

// Activate event - Clean up old caches
self.addEventListener('activate', event => {
  event.waitUntil(
    caches.keys().then(cacheNames => {
      return Promise.all(
        cacheNames.map(cacheName => {
          if (cacheName !== 'ecommerce-cache') {
            console.log('Deleting old cache:', cacheName);
            return caches.delete(cacheName);
          }
        })
      );
    })
  );
});

// Fetch event - Serve files from cache if offline
self.addEventListener('fetch', event => {
  event.respondWith(
    caches.match(event.request).then(cachedResponse => {
      if (cachedResponse) {
        return cachedResponse; // Return cached response if found
      }
      return fetch(event.request); // Otherwise, fetch from the network
```

```
    })
  );
});
```

Index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
  <title>E-commerce PWA</title>
  <script src="main.js"></script>
  <link rel="stylesheet" href="style.css" />
  <link rel="manifest" href="manifest.json" />
</head>
<body>
  <header>E-Commerce PWA</header>
  <main>
    <p>Welcome to our store. Explore amazing deals!</p>
    <a href="menu.html">Menu</a> |
    <a href="contactUs.html">Contact Us</a>
    <br><br>
    <button onclick="alert('Added to Cart!')">Add to Cart</button>
  </main>
  <script>
    if ('serviceWorker' in navigator) {
      navigator.serviceWorker.register('sw.js')
        .then(reg => console.log("SW registered!", reg))
        .catch(err => console.error("SW registration failed", err));
    }
  </script>
</body>
</html>
```

manifest.json

```
{
    "name": "E-commerce App",
    "short_name": "E-commerce",
    "start_url": "index.html",
    "scope": "/",
```

Veydant Sharma D15C 50

```
  "icons": [
    {
      "src": "https://picsum.photos/200",
      "sizes": "192x192",
      "type": "image/png"
    },
    {
      "src": "https://picsum.photos/200",
      "sizes": "512x512",
      "type": "image/png"
    }
  ],
  "theme_color": "#4CAF50",
  "background_color": "#333",
  "display": "standalone"
}
```

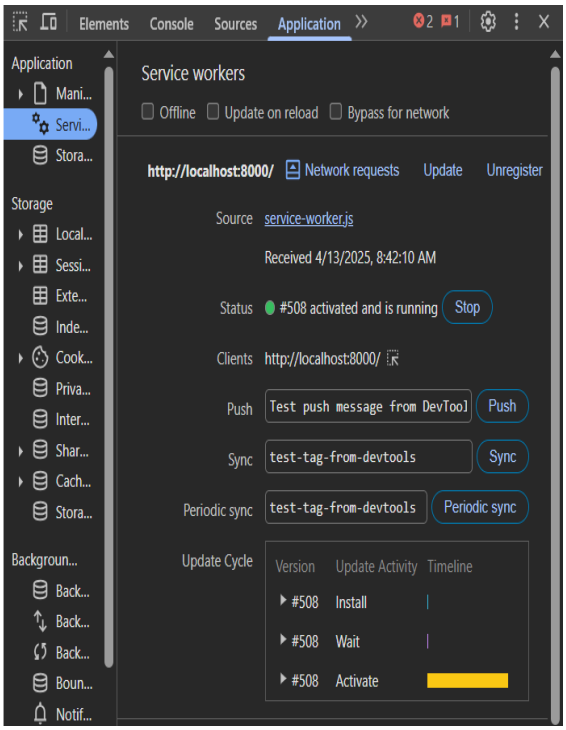
Output:

E-Commerce PWA

Welcome to our store. Explore amazing deals!

Menu | Contact Us

Add to Cart



Conclusion:

In this experiment, we successfully implemented and registered a service worker for the E-commerce PWA. We completed the installation and activation process, cached important assets, and enabled offline support. This demonstrated how service workers enhance web applications by providing better performance, offline access, and improved reliability, which are essential features of Progressive Web Apps.