Name: Kshitij Hundre DOP:
Div: D15C DOS:
Roll no: 18

Experiment-8

1. Aim

To code and register a service worker, and complete the install and activation process for a new service worker in an E-commerce Progressive Web App (PWA).

2. Basic Description

A **Progressive Web App (PWA)** uses modern web capabilities to deliver an app-like experience to users. PWAs are reliable, fast, and engaging. One of the core technologies behind a PWA is the **Service Worker**.

A **Service Worker** is a JavaScript file that runs in the background, separate from the main browser thread. It acts like a network proxy that enables features like:

- **Offline support**
- **Caching assets**
- **Push notifications**
- **Background sync**

Lifecycle of a Service Worker:

1. **Installation** – Triggered when the service worker is first registered.
2. **Activation** – Occurs after installation; prepares the worker to control pages.
3. **Fetch** – Intercepts network requests and serves them from the cache or network.

How it works in our E-Commerce PWA:

- When a user visits the website, the service worker is registered.
- It caches essential files like `index.html`, `style.css`, `manifest.json`, and product images.
- On subsequent visits, it loads content from the cache, ensuring faster performance and offline access.

Github: https://github.com/pixelbypixels/EXP7-PWA.git

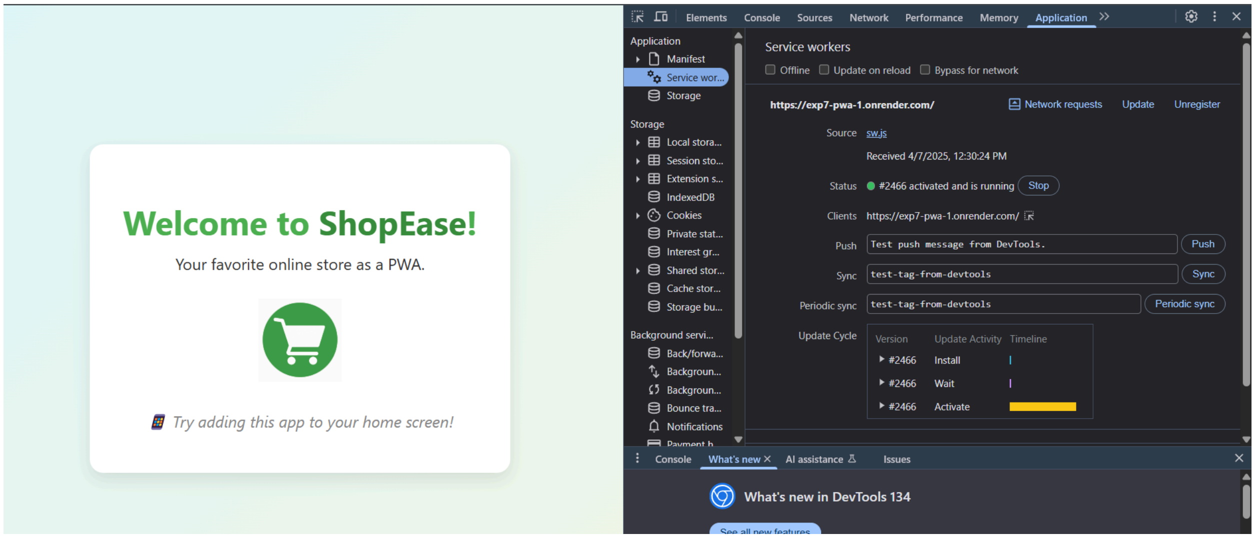
Index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>ShopEase - E-Commerce</title>

  <!-- Link to manifest -->
  <link rel="manifest" href="manifest.json">

  <!-- Link to external CSS -->
  <link rel="stylesheet" href="style.css">

  <!-- Meta tags for theme -->
  <meta name="theme-color" content="#4CAF50" />
</head>
<body>
  <div class="container">
    <h1>Welcome to <span class="highlight">ShopEase</span>!</h1>
    <p>Your favorite online store as a PWA.</p>
    <img src="icons/icon-192.png" alt="App Icon" class="app-icon" />
    <p class="tip"> Try adding this app to your home screen!</p>
  </div>
  <script src="script.js"></script>
</body>
</html>
```

sw.js

```
// INSTALL: Cache app shell
self.addEventListener('install', (event) => {
    console.log('[Service Worker] Installed');
    event.waitUntil(
      caches.open('v2').then((cache) => {
        return cache.addAll([
          '/',
          '/index.html',
          '/style.css',
          '/script.js',
          '/manifest.json',
          '/icons/icon-192.png'
        ]);
      })
    );
  });

  // ACTIVATE
  self.addEventListener('activate', (event) => {
    console.log('[Service Worker] Activated');
    // Optional: Clean old caches
    event.waitUntil(
      caches.keys().then((keys) =>
        Promise.all(
          keys.map((key) => {
            if (key !== 'v2') {
              console.log('[Service Worker] Removing old cache:', key);
              return caches.delete(key);
            }
          })
        )
      )
    );
  });

  // FETCH: Serve from cache or network
  self.addEventListener('fetch', (event) => {
    console.log('[Service Worker] Fetching:', event.request.url);
    event.respondWith(
      caches.match(event.request).then((res) => {
        return res || fetch(event.request);
      })
    );
  });

  // SYNC: Background sync example
  self.addEventListener('sync', (event) => {
    if (event.tag === 'sync-products') {
      console.log('[Service Worker] Background Sync - Products');
      event.waitUntil(syncProductData());
    }
  });

  function syncProductData() {
    return new Promise((resolve) => {
      // Simulated background sync task
      setTimeout(() => {
        console.log('✅ Product data synced in background!');
        resolve();
      }, 2000);
    });
  }

  // PUSH: Show notification when push received
  self.addEventListener('push', (event) => {
    console.log('[Service Worker] Push Received');
    let data = {};

    try {
      data = event.data.json();
    } catch (e) {
      console.warn('Push data is not JSON:', event.data.text());
      data = { title: 'ShopEase Notification', body: event.data.text() };
    }

    const title = data.title || 'ShopEase';
    const options = {
      body: data.body || 'You have a new update!',
      icon: '/icons/icon-192.png',
    };

    event.waitUntil(
      self.registration.showNotification(title, options)
    );
  });
```

3. Output:



Welcome to ShopEase!

Your favorite online store as a PWA.

📱 *Try adding this app to your home screen!*

Service workers

□ Offline □ Update on reload □ Bypass for network

https://exp7-pwa-1.onrender.com/ Network requests Update Unregister

Source sw.js

Received 4/7/2025, 12:30:24 PM

Status ● #2466 activated and is running Stop

Clients https://exp7-pwa-1.onrender.com/

Push `Test push message from DevTools.` Push

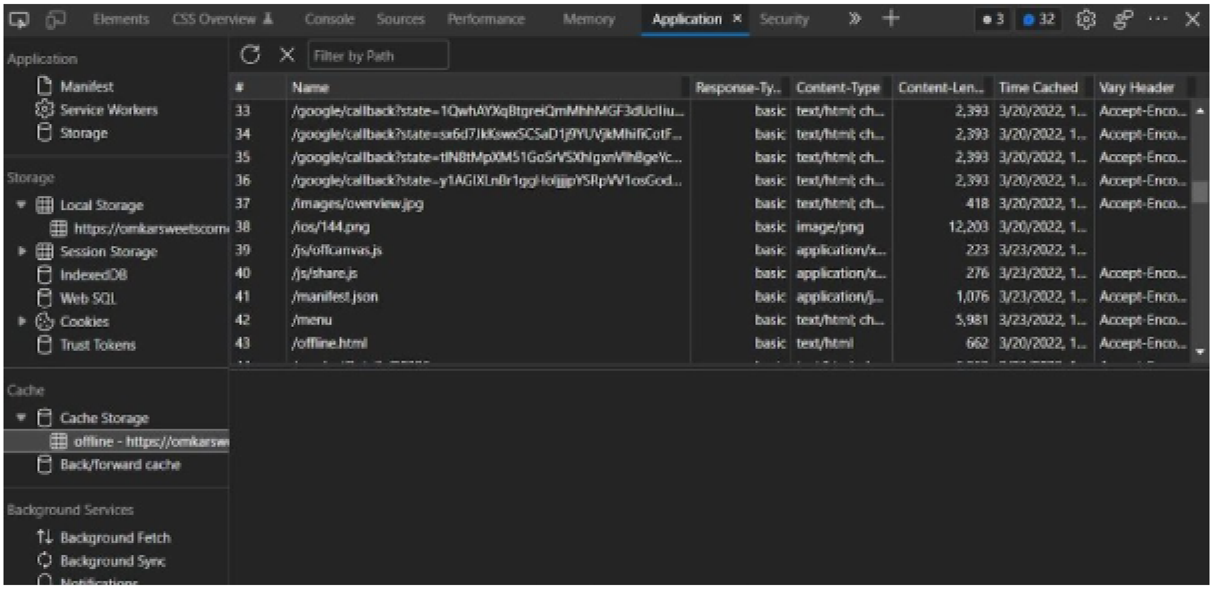
Sync `test-tag-from-devtools` Sync

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

Update Cycle

Version	Update Activity	Timeline
#2466	Install	
#2466	Wait	
#2466	Activate	

What's new in DevTools 134



#	Name	Response-Ty...	Content-Type	Content-Len...	Time Cached	Vary Header
33	/google/callback?state=1QwhAYXqBtgreiQmMhhMGF3dUcIiu...	basic	text/html; ch...	2,393	3/20/2022, 1...	Accept-Enco...
34	/google/callback?state=sx6d7JkKswxSCSaD1j9YUVjkMhifiCotF...	basic	text/html; ch...	2,393	3/20/2022, 1...	Accept-Enco...
35	/google/callback?state=tlN8tMpXM51GoSrVSXhlgxnVlhBgeYc...	basic	text/html; ch...	2,393	3/20/2022, 1...	Accept-Enco...
36	/google/callback?state=y1AGIXLnBr1ggHoljjpYSRpVV1osGod...	basic	text/html; ch...	2,393	3/20/2022, 1...	Accept-Enco...
37	/images/overview.jpg	basic	text/html; ch...	418	3/20/2022, 1...	Accept-Enco...
38	/ios/144.png	basic	image/png	12,203	3/20/2022, 1...	
39	/js/offcanvas.js	basic	application/x...	223	3/23/2022, 1...	
40	/js/share.js	basic	application/x...	276	3/23/2022, 1...	Accept-Enco...
41	/manifest.json	basic	application/j...	1,076	3/23/2022, 1...	Accept-Enco...
42	/menu	basic	text/html; ch...	5,981	3/23/2022, 1...	Accept-Enco...
43	/offline.html	basic	text/html	662	3/20/2022, 1...	Accept-Enco...

4. Conclusion

By implementing a service worker in the E-Commerce PWA, we have added offline capabilities and enhanced loading speed, making the web application more reliable and user-friendly. This experiment demonstrates how modern web technologies improve user experience in real-world scenarios like online shopping platforms.