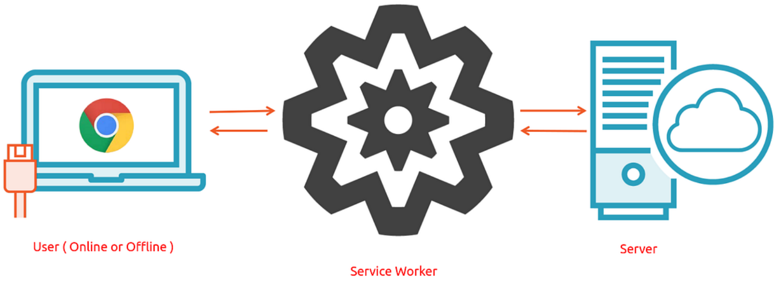
Experiment No.9

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

- **Theory:**

Service worker:

Service workers play a crucial role in Progressive Web Applications (PWAs) by enabling offline functionality, improving performance, and providing features like push notifications. These scripts run on a separate thread, intercepting network requests, caching resources, and ensuring the application works offline. Service workers need to be registered for a PWA to control network requests, manage caching, and keep the app up to date. They are essential for creating reliable PWAs that can function seamlessly even when the user is offline. Additionally, service workers can handle notifications, perform heavy calculations, and control network requests, making them a powerful tool for enhancing web applications



Fetch Event:

1. The fetch event is triggered whenever the browser makes a network request, such as fetching resources like HTML, CSS, JavaScript files, images, or API calls.
2. Service workers intercept these fetch requests, allowing them to modify the request or response, or even serve cached responses from the cache storage.
3. This event is commonly used to implement caching strategies, including serving cached resources when the network is slow or offline, or fetching fresh content from the network when available.

Sync Event:

1. The sync event is a special event available in service workers that allows you to perform background synchronization tasks.
2. It enables your PWA to synchronize data with the server even when the app is not actively in use or when the device is offline.
3. Sync events are useful for tasks like sending queued data to the server, updating content in the background, or refreshing cached resources periodically.

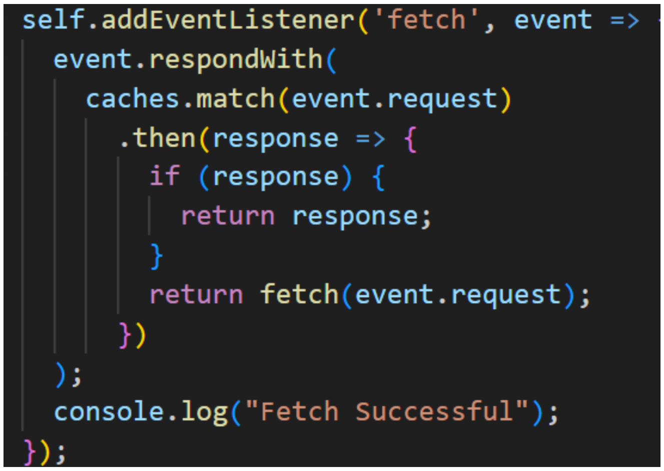
Push Event:

1. The push event occurs when the service worker receives a push notification from a server, typically initiated by a backend server or another client application.
2. This event allows service workers to display notifications to users, even when the web app is not currently open or active.
3. Push notifications are effective for engaging users with timely updates, notifications, or alerts, keeping them informed about new content, messages, or events related to the PWA.

- ## Code:-

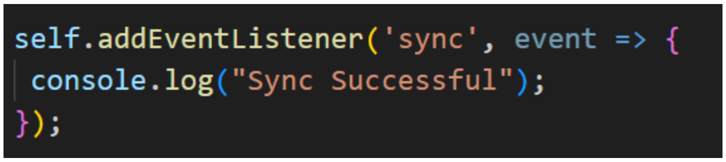
service-worker.js:

Fetch Event: The fetch event is triggered whenever a resource is fetched from the network. In the service worker, you can intercept network requests and respond with cached resources or fetch resources from the network.



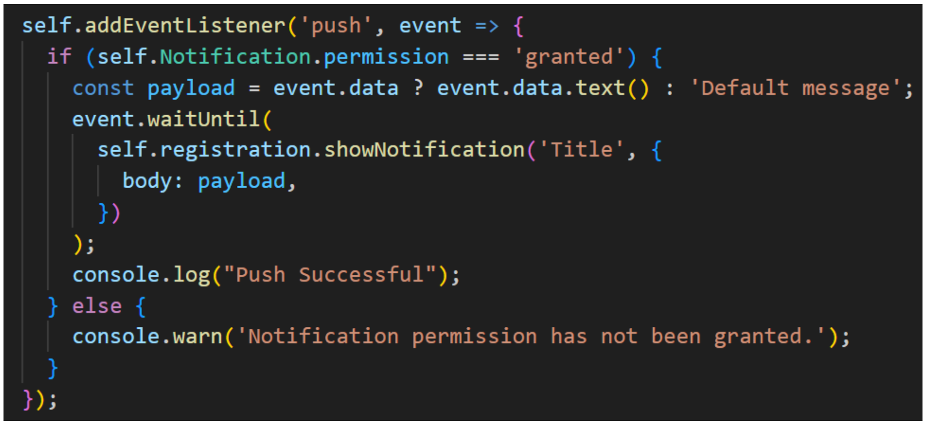
```
self.addEventListener('fetch', event =>
  event.respondWith(
    caches.match(event.request)
      .then(response => {
        if (response) {
          return response;
        }
        return fetch(event.request);
      })
  );
  console.log("Fetch Successful");
});
```

Sync Event: The sync event allows you to perform background synchronization tasks, such as sending data to a server when the network connection is available.



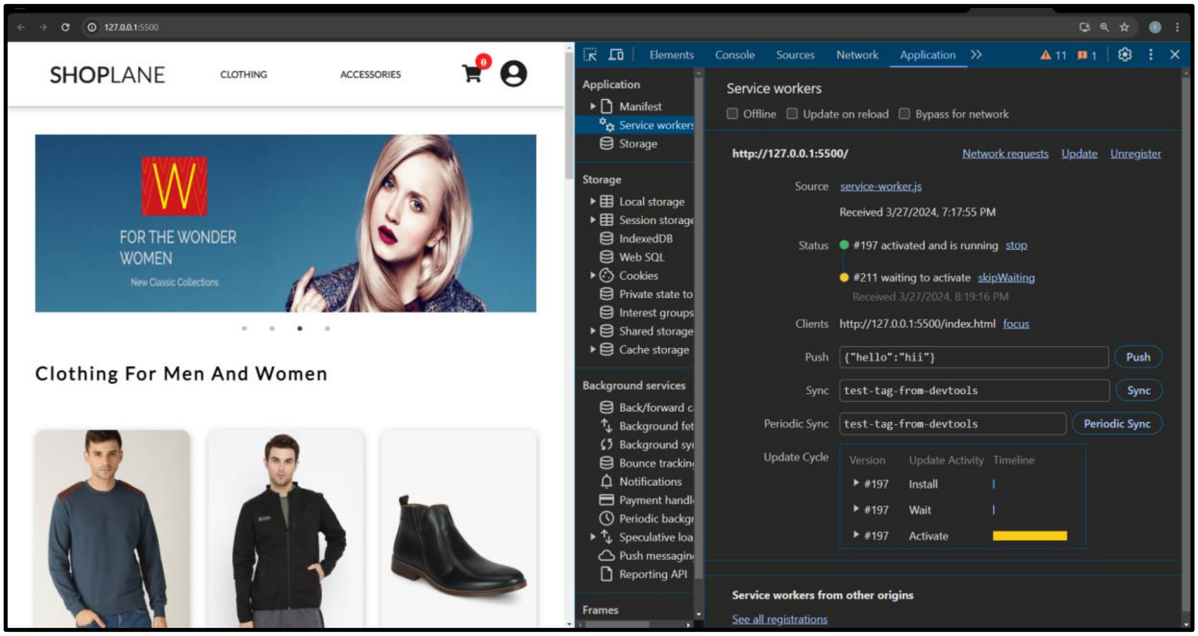
```
self.addEventListener('sync', event => {
  console.log("Sync Successful");
});
```

Push Event: The push event is triggered when a push notification is received by the service worker. You can handle this event to display the received notification to the user or perform any other action.



```
self.addEventListener('push', event => {
  if (self.Notification.permission === 'granted') {
    const payload = event.data ? event.data.text() : 'Default message';
    event.waitUntil(
      self.registration.showNotification('Title', {
        body: payload,
      })
    );
    console.log("Push Successful");
  } else {
    console.warn('Notification permission has not been granted.');
  }
});
```

- **Output:-**



Service workers

□ Offline □ Update on reload □ Bypass for network

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

Source service-worker.js

Received 3/27/2024, 7:17:55 PM

Status ● #197 activated and is running stop

● #211 waiting to activate skipWaiting

Received 3/27/2024, 8:19:16 PM

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

Push {"hello":"hii"} Push

Sync test-tag-from-devtools Sync

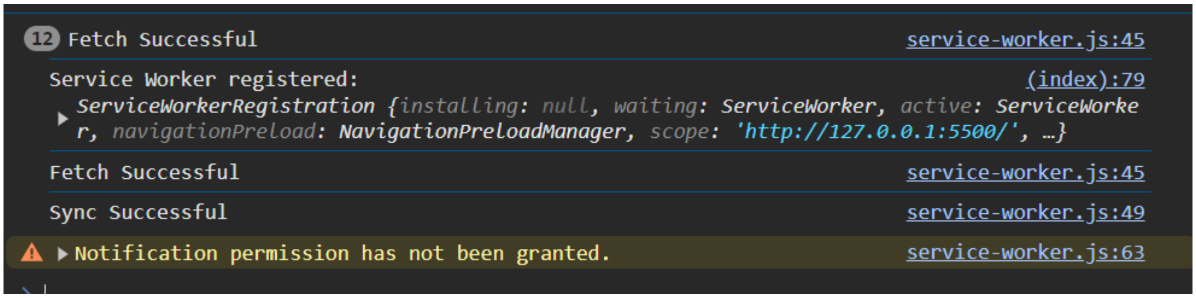
Periodic Sync test-tag-from-devtools Periodic Sync

Update Cycle

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

Service workers from other origins

See all registrations



```
12 Fetch Successful                                              service-worker.js:45
Service Worker registered:                                              (index):79
▸ ServiceWorkerRegistration {installing: null, waiting: ServiceWorker, active: ServiceWorke
  r, navigationPreload: NavigationPreloadManager, scope: 'http://127.0.0.1:5500/', …}
Fetch Successful                                                 service-worker.js:45
Sync Successful                                                  service-worker.js:49
⚠ ▸Notification permission has not been granted.                 service-worker.js:63
```

- **Conclusion:**

In conclusion, by implementing service worker events like fetch, sync, and push for the E-commerce PWA, we've enhanced its functionality. This enables features such as offline caching, background synchronization, and push notifications, ensuring a seamless and engaging user experience across various devices and network conditions.