# Experiment No: 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:

* **Service Worker**

Service Worker is a script that works on browser background without user interaction independently. Also, It resembles a proxy that works on the user side. With this script, you can track network traffic of the page, manage push notifications and develop “offline first” web applications with Cache API.

Things to note about Service Worker:

1. A service worker is a programmable network proxy that lets you control how network requests from your page are handled.
2. Service workers only run over HTTPS. Because service workers can intercept network requests and modify responses, "man-in-the-middle" attacks could be very bad.
3. The service worker becomes idle when not in use and restarts when it's next needed. You cannot rely on a global state persisting between events. If there is information that you need to persist and reuse across restarts, you can use IndexedDB databases.

## Uses of Service Workers

* 1. **You can dominate Network Traffic:** You can manage all network traffic of the page and do any manipulations. For example, when the page requests a CSS file, you can send plain text as a response or when the page requests an HTML file, you can send a png file as a response. You can also send a true response too.
  2. **You can Cache:** You can cache any request/response pair with Service Worker and Cache API and you can access these offline content anytime.
  3. **You can manage Push Notifications:** You can manage push notifications with Service Worker and show any information message to the user.
  4. **You can Continue:** Although Internet connection is broken, you can start any process with Background Sync of Service Worker.

## Limitations of Service Workers

1. You can’t access the **Window:** You can’t access the window, therefore, You can’t manipulate DOM elements. But, you can communicate to the window through post Message and manage processes that you want.
2. You can’t work it on **80 Port:** Service Worker just can work on HTTPS protocol. But you can work on localhost during development.

## Service Worker Cycle

A service worker goes through three steps in its life cycle:

* Registration
* Installation
* Activation

**Code:**

1. **flutter\_service\_worker.js**

self.addEventListener('install', (event) => {

console.log('[Service Worker] Install event');

self.skipWaiting(); // Activate worker immediately

});

self.addEventListener('activate', (event) => {

console.log('[Service Worker] Activate event');

// Cleanup old caches here if needed

});

self.addEventListener('fetch', (event) => {

event.respondWith(

caches.match(event.request).then((response) => {

return response || fetch(event.request);

})

);

});

1. **index.html**

<!DOCTYPE html>

<html>

<head>

<!--

If you are serving your web app in a path other than the root, change the

href value below to reflect the base path you are serving from.

The path provided below has to start and end with a slash "/" in order for

it to work correctly.

For more details:

\* https://developer.mozilla.org/en-US/docs/Web/HTML/Element/base

This is a placeholder for base href that will be replaced by the value of

the `--base-href` argument provided to `flutter build`.

-->

<base href="$FLUTTER\_BASE\_HREF">

<meta charset="UTF-8">

<meta content="IE=Edge" http-equiv="X-UA-Compatible">

<meta name="description" content="A new Flutter project.">

<!-- iOS meta tags & icons -->

<meta name="apple-mobile-web-app-capable" content="yes">

<meta name="apple-mobile-web-app-status-bar-style" content="black">

<meta name="apple-mobile-web-app-title" content="activibe">

<link rel="apple-touch-icon" href="icons/Icon-192.png">

<meta name="google-signin-client\_id" content="15128894708-pqnk893c2cnslsldu7g1ghk33dech5vm.apps.googleusercontent.com">

<script src="https://accounts.google.com/gsi/client" async defer></script>

<!-- Favicon -->

<link rel="icon" type="image/png" href="favicon.png"/>

<title>activibe</title>

<link rel="manifest" href="manifest.json">

</head>

<body>

<script src="flutter\_bootstrap.js" async></script>

</body>

<script>

if ('serviceWorker' in navigator) {

window.addEventListener('load', function () {

navigator.serviceWorker.register('flutter\_service\_worker.js')

.then(function (registration) {

console.log('Service Worker registered with scope:', registration.scope);

}).catch(function (error) {

console.log('Service Worker registration failed:', error);

});

});

}

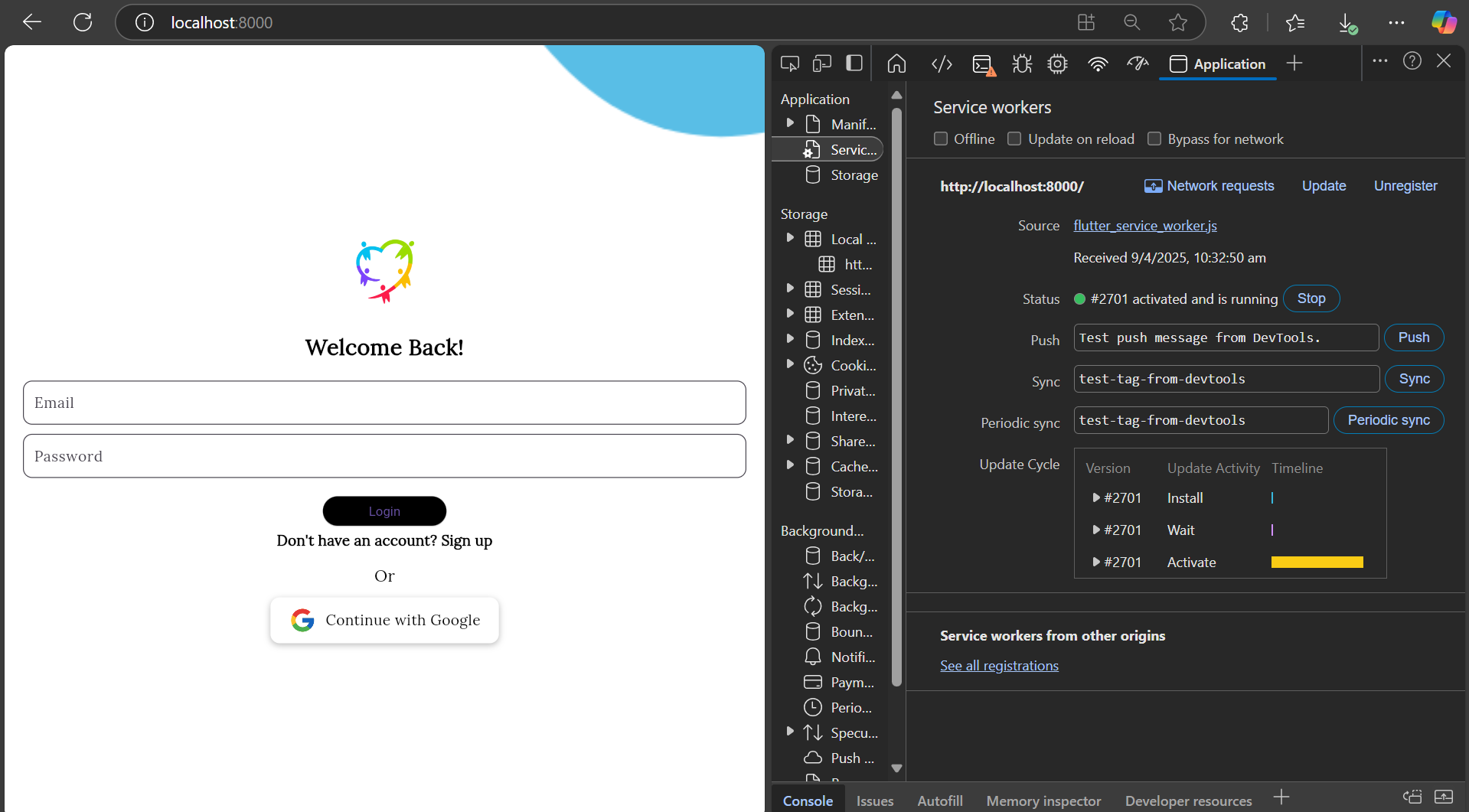
</script>

</html>

**Github Link:** [**sadneya145/acti\_web**](https://github.com/sadneya145/acti_web)

**Output:**

* After code update in flutter\_service\_worker.js a serviceworker got created in the background.
* And its status is #2701 activated and running means it is actively running in the background.

****