**Experiment 08**

**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.

**Code**:

***index.html***

<!DOCTYPE html>

<p>An image will appear here in 3 seconds:</p>

<script>

navigator.serviceWorker

.register("./sw.js")

.then((reg) => console.log("SW registered!", reg))

.catch((err) => console.log("Boo!", err));

setTimeout(() => {

const img = new Image();

img.src = "./horse.svg";

img.width = 300;

document.body.appendChild(img);

}, 3000);

</script>

***sw.js***

const expectedCaches = ['static-v2'];

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

console.log('V2 installing…');

// cache a horse SVG into a new cache, static-v2

event.waitUntil(

caches.open('static-v2').then(cache => cache.add('./horse.svg'))

);

});

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

// delete any caches that aren't in expectedCaches

// which will get rid of static-v1

event.waitUntil(

caches.keys().then(keys => Promise.all(

keys.map(key => {

if (!expectedCaches.includes(key)) {

return caches.delete(key);

}

})

)).then(() => {

console.log('V2 now ready to handle fetches!');

})

);

});

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

const url = new URL(event.request.url);

// serve the horse SVG from the cache if the request is

// same-origin and the path is '/dog.svg'

if (url.origin == location.origin && url.pathname == './dog.svg') {

event.respondWith(caches.match('./horse.svg'));

}

});

**Output**:







