

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!');
  })
});
```

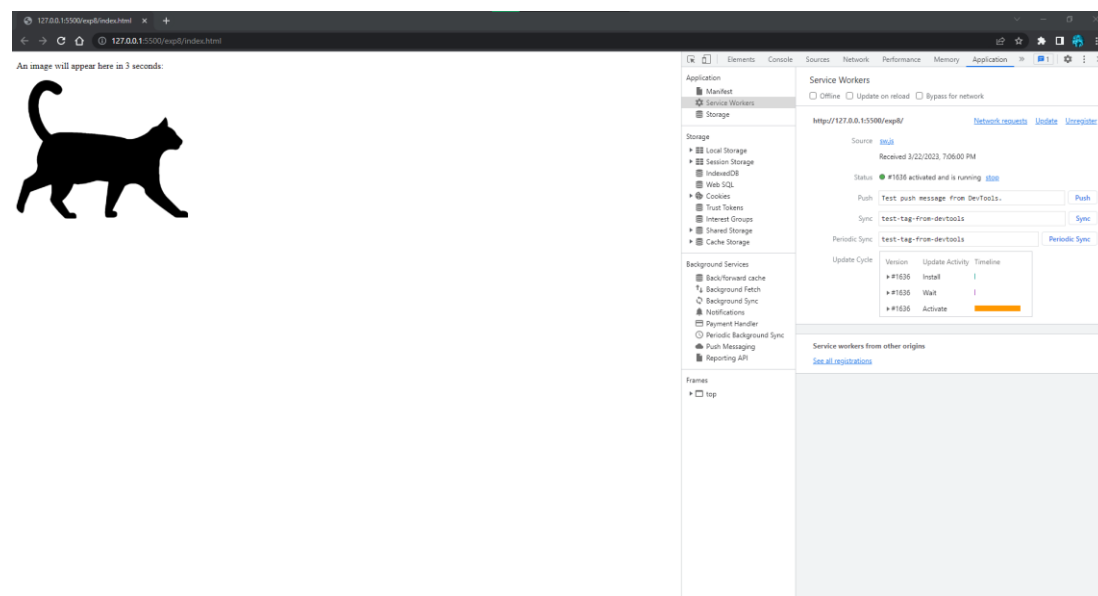
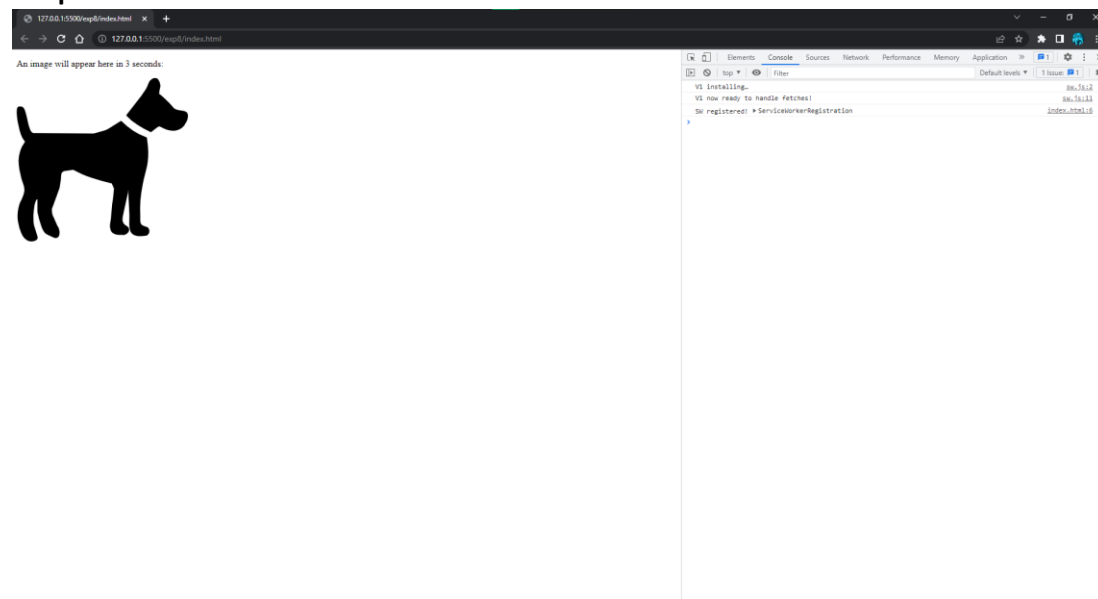
```
);
});
```

```
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:



127.0.0.1:5500/exp/ index.html

127.0.0.1:5500/exp/ index.html

An image will appear here in 3 seconds:



Application

- Manifest
- Service Workers
- Storage

Storage

- Local Storage
- Session Storage
- IndexedDB
- Web SQL
- Cookies
- Trust Tokens
- Interest Groups
- Shared Storage
- Cache Storage

Background Services

- Back/forward cache
- Background Fetch
- Background Sync
- Notifications
- Payment Handler
- Periodic Background Sync
- Push Messaging
- Reporting API

Frames

- top

Service Workers

☐ Offline ☐ Update on reload ☐ Bypass for network

http://127.0.0.1:5500/exp/ [Network requests](#) [Update](#) [Unregister](#)

Source [URL](#)

Received 3/22/2023, 7:06:00 PM

Status **#1636** activated and is running [stop](#)

#1637 waiting to activate [skicollation](#)

Received 3/22/2023, 7:15:59 PM

Push [Push](#)

Sync [Sync](#)

Periodic Sync [Periodic Sync](#)

Update Cycle

Version	Update Activity	Timeline
#1636	Install	I
#1636	Wait	I
#1636	Activate	I


Service workers from other origins

[See all registrations](#)

127.0.0.1:5500/exp/ index.html

127.0.0.1:5500/exp/ index.html

An image will appear here in 3 seconds:



Created by Peleg Red
from the Noun Project

Application

- Manifest
- Service Workers
- Storage

Storage

- Local Storage
- Session Storage
- IndexedDB
- Web SQL
- Cookies
- Trust Tokens
- Interest Groups
- Shared Storage
- Cache Storage

Background Services

- Back/forward cache
- Background Fetch
- Background Sync
- Notifications
- Payment Handler
- Periodic Background Sync
- Push Messaging
- Reporting API

Frames

- top

Service Workers

☐ Offline ☒ Update on reload ☐ Bypass for network

http://127.0.0.1:5500/exp/ [Network requests](#) [Update](#) [Unregister](#)

Source [URL](#)

Received 3/22/2023, 7:18:34 PM

Status **#1641** activated and is running [stop](#)

Push [Push](#)

Sync [Sync](#)

Periodic Sync [Periodic Sync](#)

Update Cycle

Version	Update Activity	Timeline
#1641	Install	I
#1641	Wait	I
#1641	Activate	I

Service workers from other origins

[See all registrations](#)