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

A Service Worker is a script that runs in the background of a web application and helps improve performance and offline functionality. It allows features like caching files, background syncing, and push notifications. For PWAs (Progressive Web Apps), service workers are essential for creating a reliable and fast experience, even without internet.

In this experiment, we focused on coding and registering a new service worker for our E-commerce PWA "EcoGadgetHub". Here's what we did:

1. Registered the Service Worker in index.html:
    - We used JavaScript to check if the browser supports service workers.
    - If supported, the service worker (serviceworker.js) is registered when the page loads.

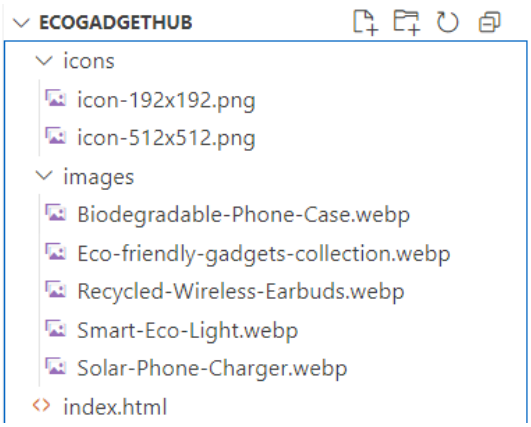
2. Installation Phase:
    - In the install event, we created a cache named 'eco-gadget-hub-v1'.
    - We preloaded and stored important files like HTML, images, and icons in this cache so they can be accessed offline.

3. Activation Phase:
    - In the activate event, we checked for older cache versions.
    - If any old cache is found, it is deleted to keep the storage clean and updated.

4. Fetch Handling:
    - For every request, the service worker checks if the file is in the cache.
    - If it is, the cached file is returned (faster and works offline).
    - If not, it fetches the file from the internet.

By doing this, our E-commerce PWA now works smoothly even without internet access, and loads faster due to cached resources.

Folder Structure



index.html

```
<!DOCTYPE html>
<html lang="en">
 <head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1" />
  <title>EcoGadgetHub – Sustainable Tech Gadgets</title>
  <link rel="manifest" href="manifest.json">
  </style>
 </head>
 <body>
  <script>
   if ("serviceWorker" in navigator) {
    window.addEventListener("load", () => {
     navigator.serviceWorker
      .register("/serviceworker.js")
      .then((registration) => {
       console.log(
        "✅ Service Worker registered! Scope:",
        registration.scope
       );
      })
      .catch((error) => {
       console.log("❌ Service Worker registration failed:", error);
      });
    });
   }
  </script>
 </body>
</html>
```

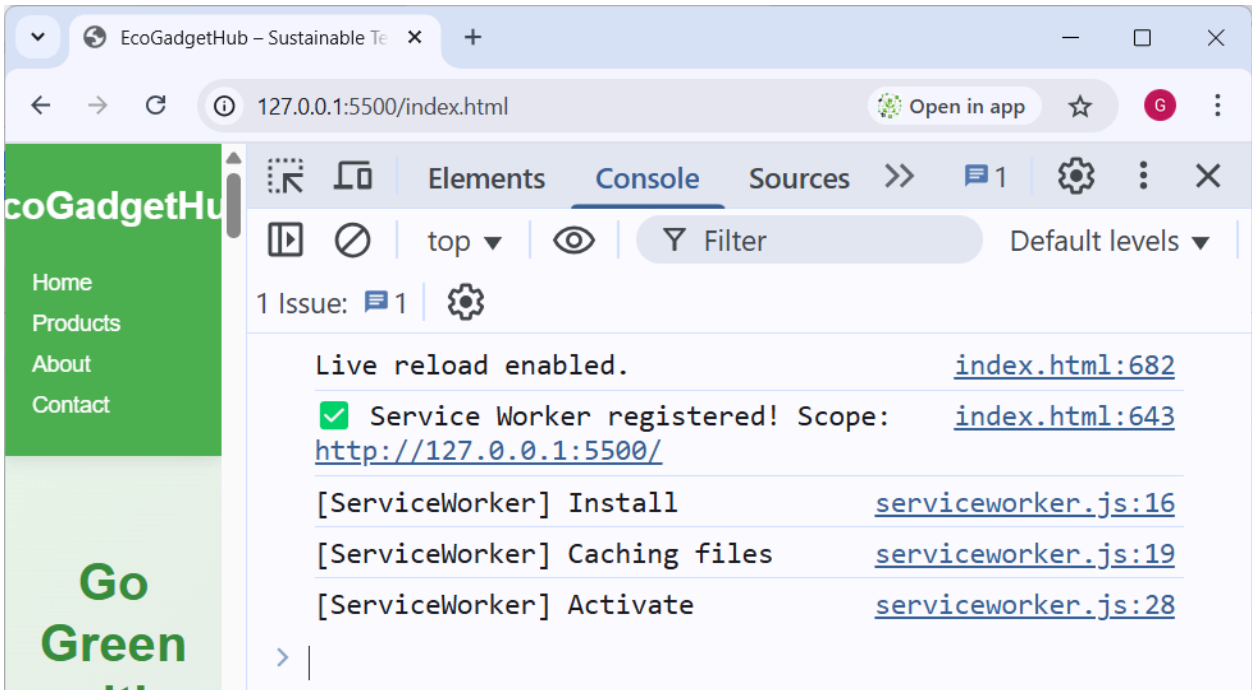
serviceworker.js

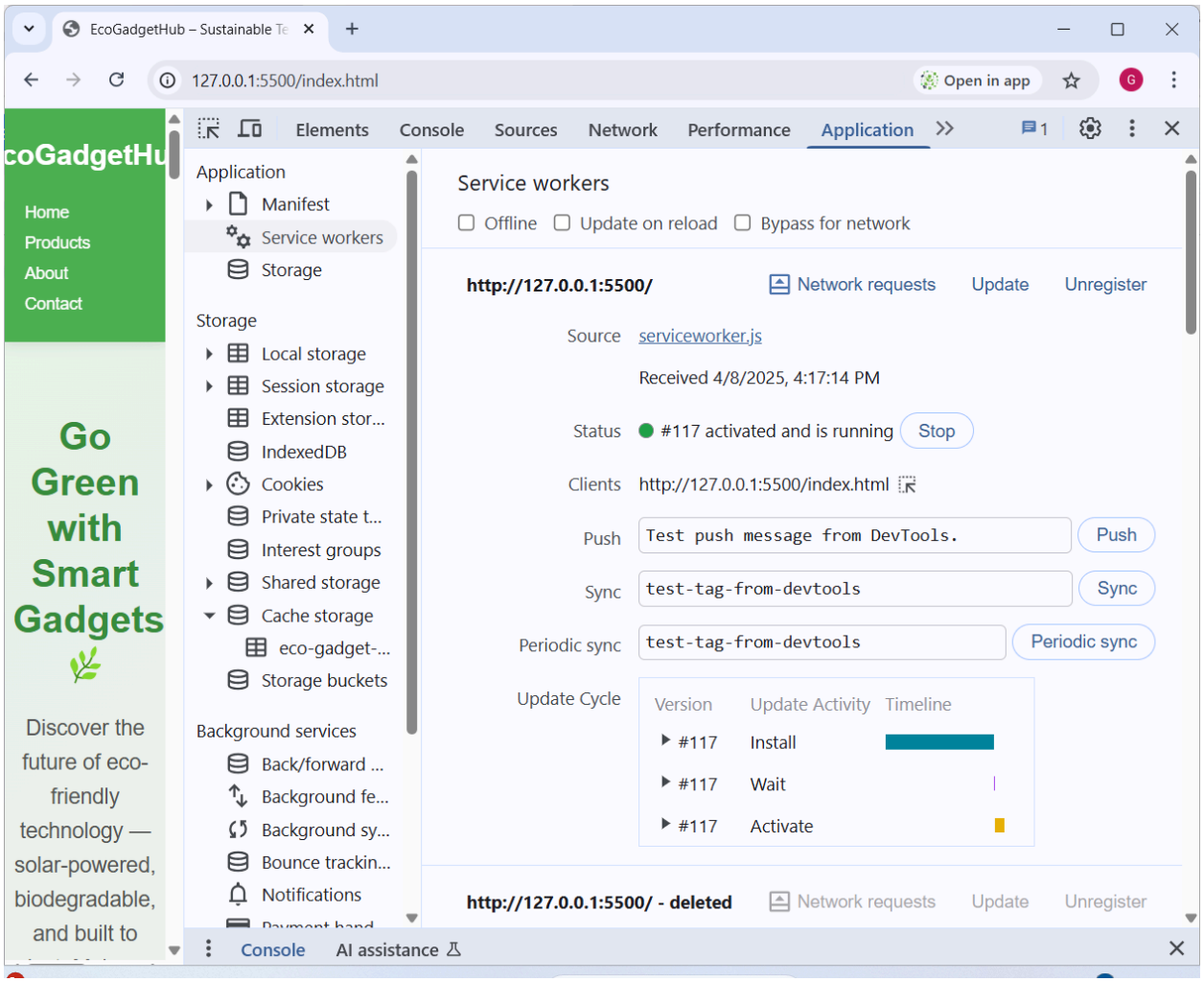
```
const CACHE_NAME = 'eco-gadget-hub-v1';
const urlsToCache = [
 '/',
 '/index.html',
 'icons/icon-192x192.png',
 'icons/icon-512x512.png',
 'images/Biodegradable-Phone-Case.webp',
 'images/Eco-friendly-gadgets-collection.webp',
```

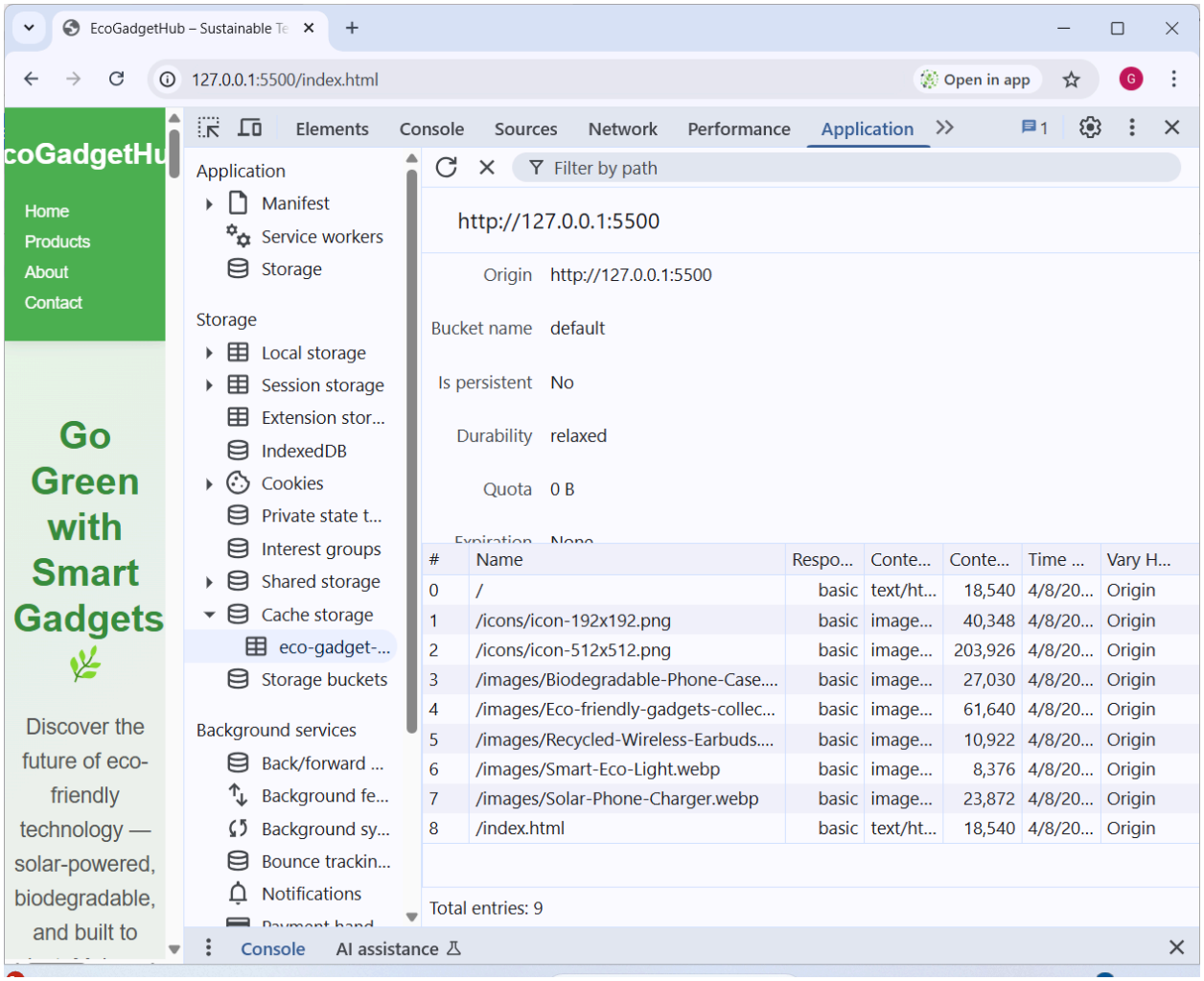
```
  'images/Recycled-Wireless-Earbuds.webp',
  'images/Smart-Eco-Light.webp',
  'images/Solar-Phone-Charger.webp',
];

// Install Event
self.addEventListener("install", (event) => {
  console.log("[ServiceWorker] Install");
  event.waitUntil(
    caches.open(CACHE_NAME).then((cache) => {
      console.log("[ServiceWorker] Caching files");
      return cache.addAll(urlsToCache);
    })
  );
});

// Activate Event
self.addEventListener("activate", (event) => {
  console.log("[ServiceWorker] Activate");
  event.waitUntil(
    caches.keys().then((keyList) =>
      Promise.all(
        keyList.map((key) => {
          if (key !== CACHE_NAME) {
            console.log("[ServiceWorker] Removing old cache", key);
            return caches.delete(key);
          }
        })
      )
    )
  );
  return self.clients.claim();
});

// Fetch event
self.addEventListener('fetch', (event) => {
  event.respondWith(
    caches.match(event.request)
      .then((response) => response || fetch(event.request))
```

```
  );
});
```

Screenshot


EcoGadgetHub – Sustainable Te
127.0.0.1:5500/index.html
Open in app
Elements
Console
Sources
top
Filter
Default levels
1 Issue: 1
Live reload enabled. index.html:682
Service Worker registered! Scope: http://127.0.0.1:5500/ index.html:643
[ServiceWorker] Install serviceworker.js:16
[ServiceWorker] Caching files serviceworker.js:19
[ServiceWorker] Activate serviceworker.js:28
Home
Products
About
Contact
Go Green


EcoGadgetHub – Sustainable Te
127.0.0.1:5500/index.html
Open in app
Elements
Console
Sources
Network
Performance
Application
Application
Manifest
Service workers
Storage
Storage
Local storage
Session storage
Extension stor...
IndexedDB
Cookies
Private state t...
Interest groups
Shared storage
Cache storage
eco-gadget-...
Storage buckets
Background services
Back/forward ...
Background fe...
Background sy...
Bounce trackin...
Notifications
Service workers
Offline
Update on reload
Bypass for network
http://127.0.0.1:5500/
Network requests
Update
Unregister
Source serviceworker.js
Received 4/8/2025, 4:17:14 PM
Status #117 activated and is running
Stop
Clients http://127.0.0.1:5500/index.html
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
#117 Install
#117 Wait
#117 Activate
http://127.0.0.1:5500/ - deleted
Network requests
Update
Unregister
Console
AI assistance
Home
Products
About
Contact
Go Green with Smart Gadgets
Discover the future of eco-friendly technology — solar-powered, biodegradable, and built to



Conclusion

In this experiment, we successfully implemented and registered a service worker to cache essential files and enable offline access for EcoGadgetHub. Initially, we faced an issue where the service worker wasn't activating due to incorrect file paths, but we resolved it by verifying the file locations and correcting the URLs in the cache list.