Experiment No.8
MAD & PWA LAB

- **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 –

A service worker is a script that runs independently in the browser background. On the user side, it can intercept its network requests and decide what to load (fetch).
Service workers mainly serve features like background sync, push notifications and they are commonly used for'offline first' applications, giving the developers the opportunity to take complete control over the user experience.

Before it's time there has been API called AppCache, which has been trying to serve the offline experience feature. However, there have been numerous problems in the interface of the AppCache API and Service Workers are here, going over them.

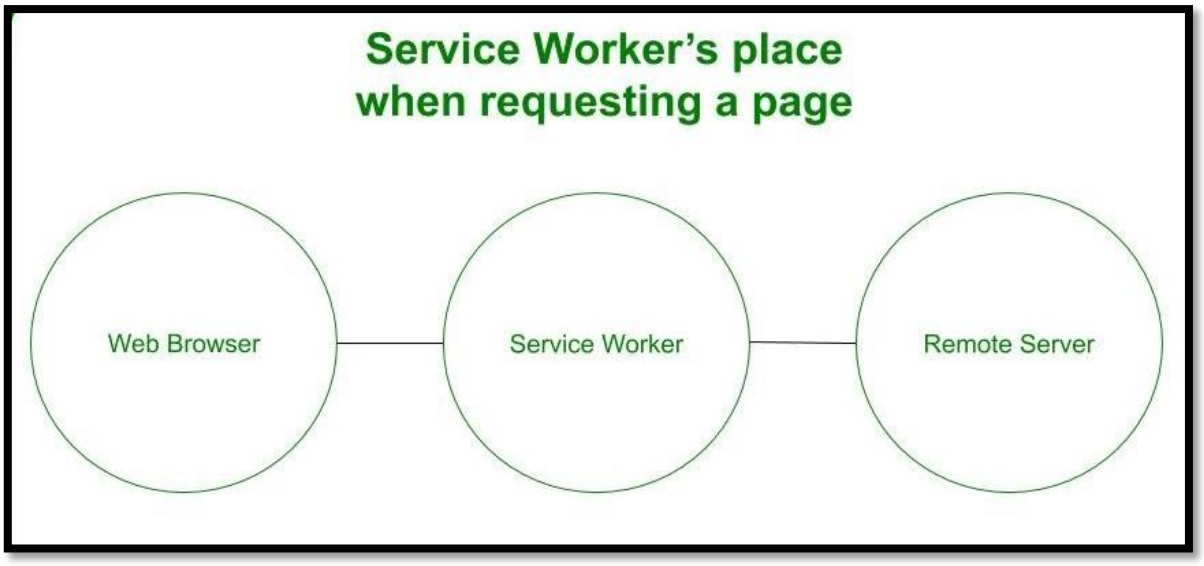
→ The service worker life cycle –

The service worker lifecycle is completely separate from the web page. It's a programmable network proxy, which is terminated when it's not used and restarted when it's next needed. Service Workers heavily rely on the use of Javascript Promises , so it's good to go over them if they are new to you.

During installation, the service worker can cache some static assets like web pages. If the browser cache the files successfully, the service worker gets installed.

Afterward, the worker needs to be activated. During activation the service worker can manage and decide what to happen to the old caches, typically they are being deleted and replaced with the new versions.

Lastly, after activation, the service worker takes control over all pages in his scope, without the page which initially registered the service worker, which needs to be refreshed. The service worker is smart in terms of memory usage and will be terminated if there is nothing to fetch and there are no message events occurring.



→ **Prerequisites :**

1. HTTPS unless on localhost
   - Service workers require the use of HTTPS connection. Before deployment, the workers does work under the localhost server but if you want to upload it to the internet you need to have the HTTPS setup on your server. One good place to host free demos are the GitHub Pages, which are server over HTTPS.
2. Browser support
   - Service Workers are highly supported over the internet by Chrome, Firefox, Opera, Safari and Edge, which makes them worthy for deployment.

- **Implementation:**

→ **service-worker.js [code] –**

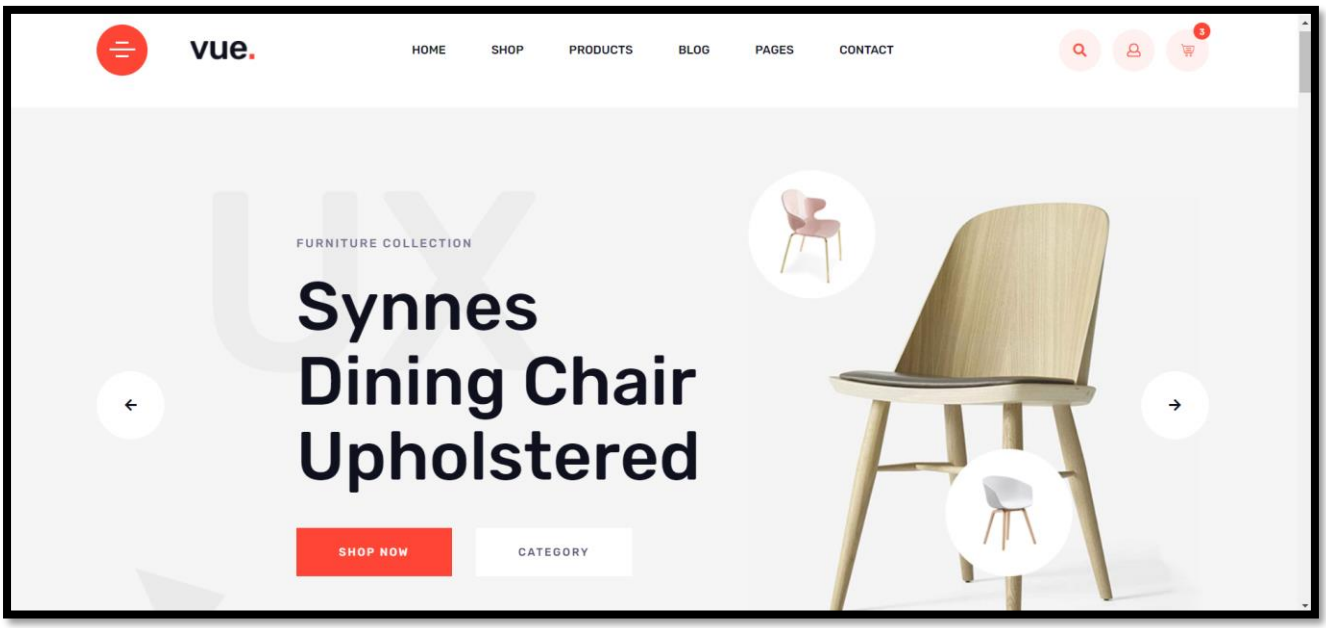
```
const CACHE_NAME = 'ecommerce-pwa-v1';
const urlsToCache = [
'/',
'index.html',
'index-2.html',
'index-3.html',
'index-4.html',
'index-5.html',
'login.html',
'cart.html',
'blog.html',
'about.html',
];
self.addEventListener('install', event => {
event.waitUntil(
caches.open(CACHE_NAME)
.then(cache => {
return cache.addAll(urlsToCache);
})
);
});
self.addEventListener('activate', event => {
event.waitUntil(
caches.keys().then(cacheNames => {
return Promise.all(
cacheNames.map(cacheName => {
if (cacheName !== CACHE_NAME) {
return caches.delete(cacheName);
}}));})
);});
self.addEventListener('fetch', event => {
```

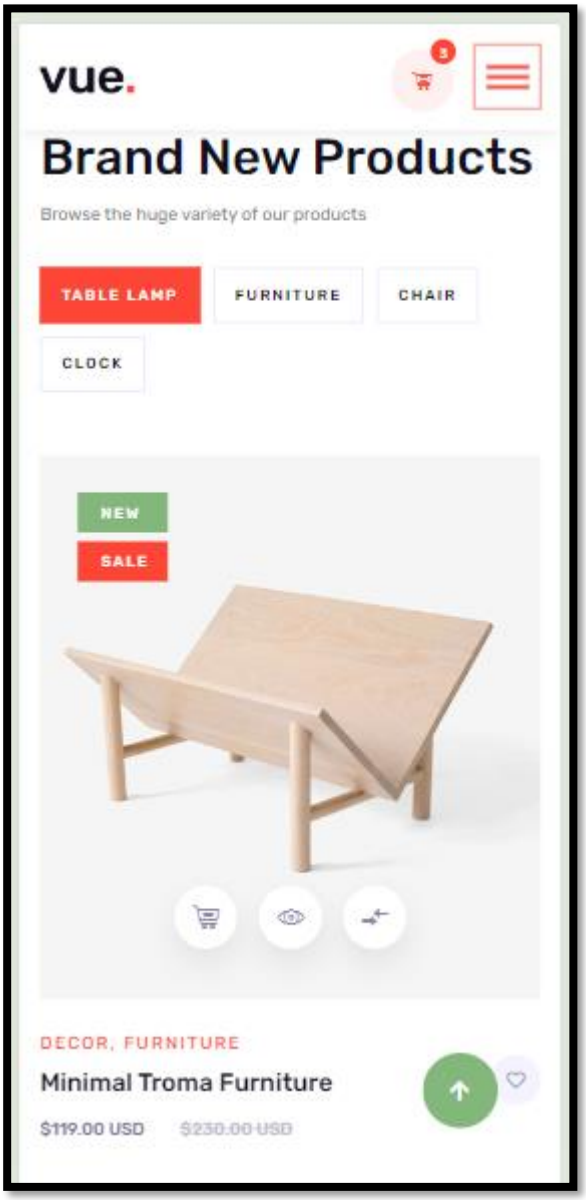
```
event.respondWith(
caches.match(event.request)
.then(response => {
if (response) {
return response;
}
return fetch(event.request)
.then(response => {
if (!response || response.status !== 200 ||
response.type !== 'basic') {
return response;
}
const responseToCache = response.clone();
caches.open(CACHE_NAME)
.then(cache => {
cache.put(event.request, responseToCache);
});
return response;
});}));});
```

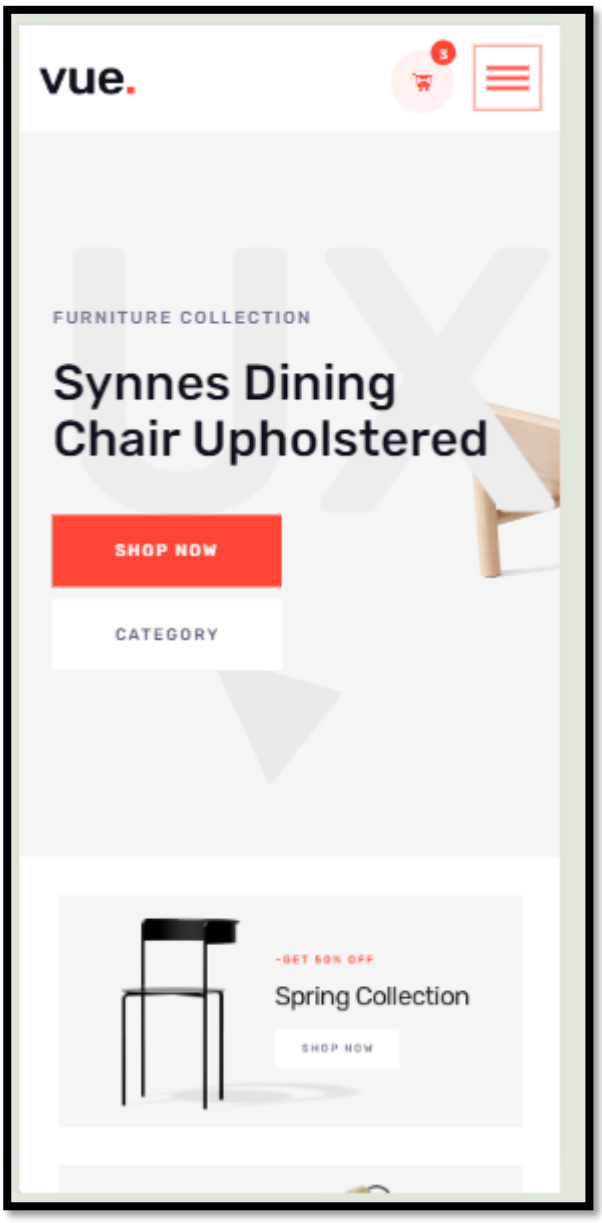
→ **index.html [code] –**

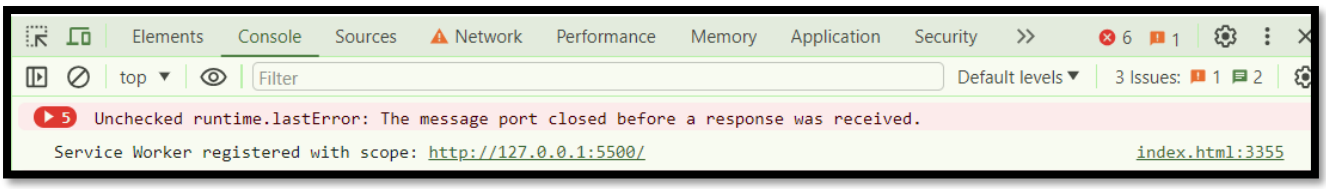
```
<script>
        if ('serviceWorker' in navigator) {
            window.addEventListener('load',
function() {
                navigator.serviceWorker.register('
/service-worker.js')
                    .then(function(registration) {
                        console.log('Service Worker
registered with scope:', registration.scope);
                    })
                    .catch(function(error) {
                        console.error('Service
Worker registration failed:', error);
                    });
            });
        }
    </script>
```

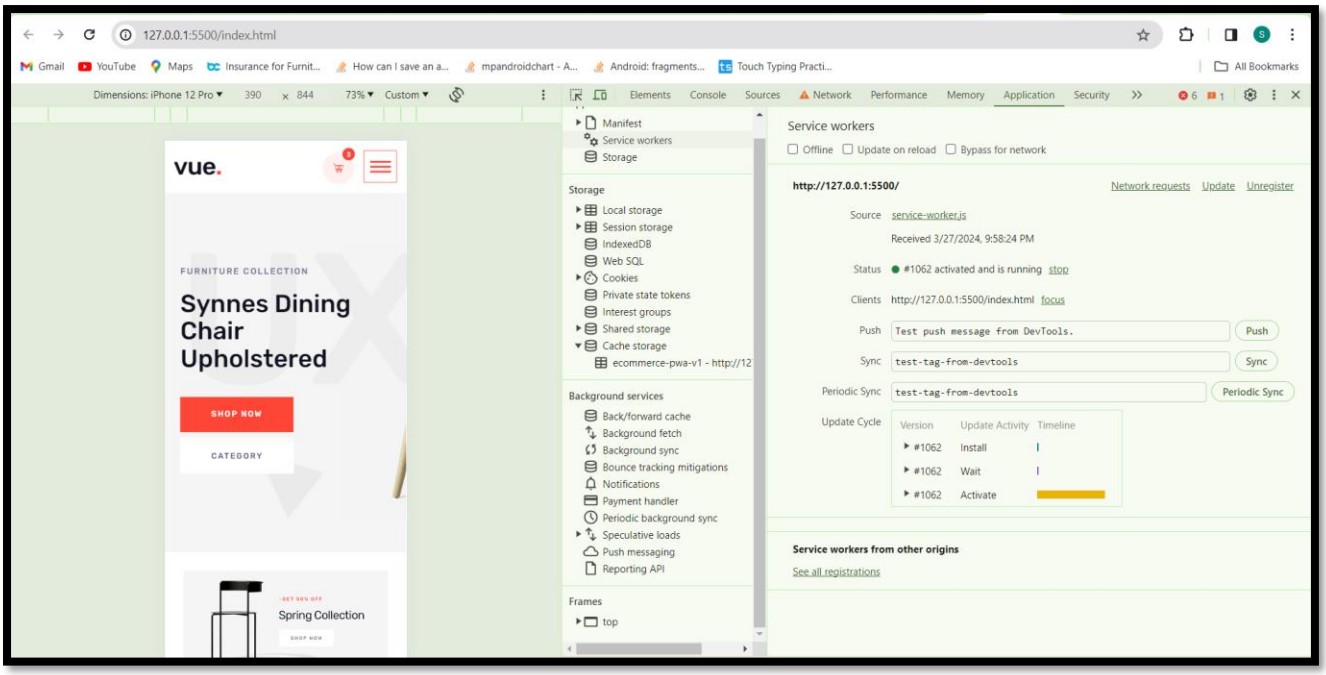
→ **Website SnapShots –**

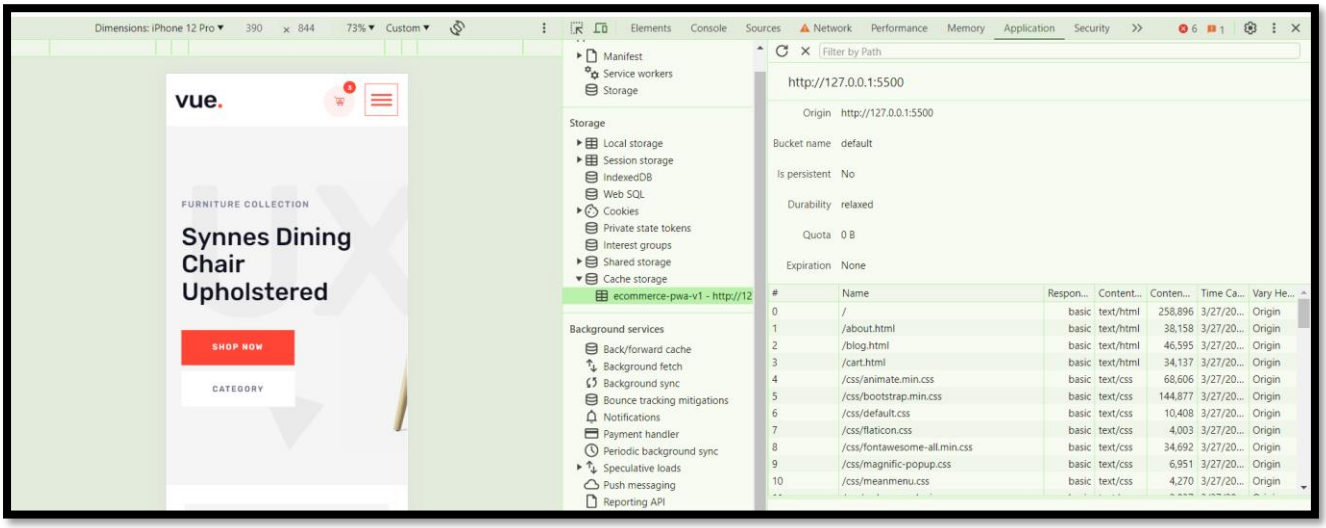



vue.
Brand New Products
Browse the huge variety of our products
TABLE LAMP
FURNITURE
CHAIR
CLOCK
NEW
SALE
DECOR, FURNITURE
Minimal Troma Furniture
$119.00 USD
$230.00 USD


vue.
FURNITURE COLLECTION
Synnes Dining Chair Upholstered
SHOP NOW
CATEGORY
Spring Collection
SHOP NOW


Elements
Console
Sources
Network
Performance
Memory
Application
Security
top
Filter
Default levels
3 Issues:
Unchecked runtime.lastError: The message port closed before a response was received.
Service Worker registered with scope: http://127.0.0.1:5500/
index.html:3355





- **Conclusion:**

  Hence, successfully implemented a service worker for the E-commerce Progressive Web App (PWA) entails crafting the service worker code to manage caching and network requests, integrating the service worker registration into the primary JavaScript file of the application, and ensuring the successful execution of the install and activation phases.