Gungun. Kamlesh. Panjabi
DI5B/ 64

PWA EXP NO 08

Aim : To create and register a service worker, and complete the install and activate process for a new service worker for the E - commerce PWA

Theory :

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 Manage push notification and develop 'offline trust' web application with cache API

Service worker cycle.

A service worker get through three steps in the lifecycle
- Registration
- Installation
- Activation

what can we do with service workers?
- you can dominate Network Traffic
- you can cache
- you can Manage Push Notification
- you can continue
- what can't we do this with Service worker

Ans: you can't access the window
you can't work it on so part

Activation : The activate event ensures that old
are cleaned up with a new version of the
service worker is installed

caching strategy : The service worker uses a cac
strategy (such as cache first or network-first
to device how to serve resource to user
when they're offline

conclusion :- In conclusion the process of coding
registering a service worker for my fundraise
webapp involves serval important step to en
your app function offline and deliver enha
performances.

Name: GUNGUN PANJABI

Div:D15B

Roll No: 64

Mpl Practical 8

Steps to Make Your Vite React App a PWA

1 Register the Service Worker
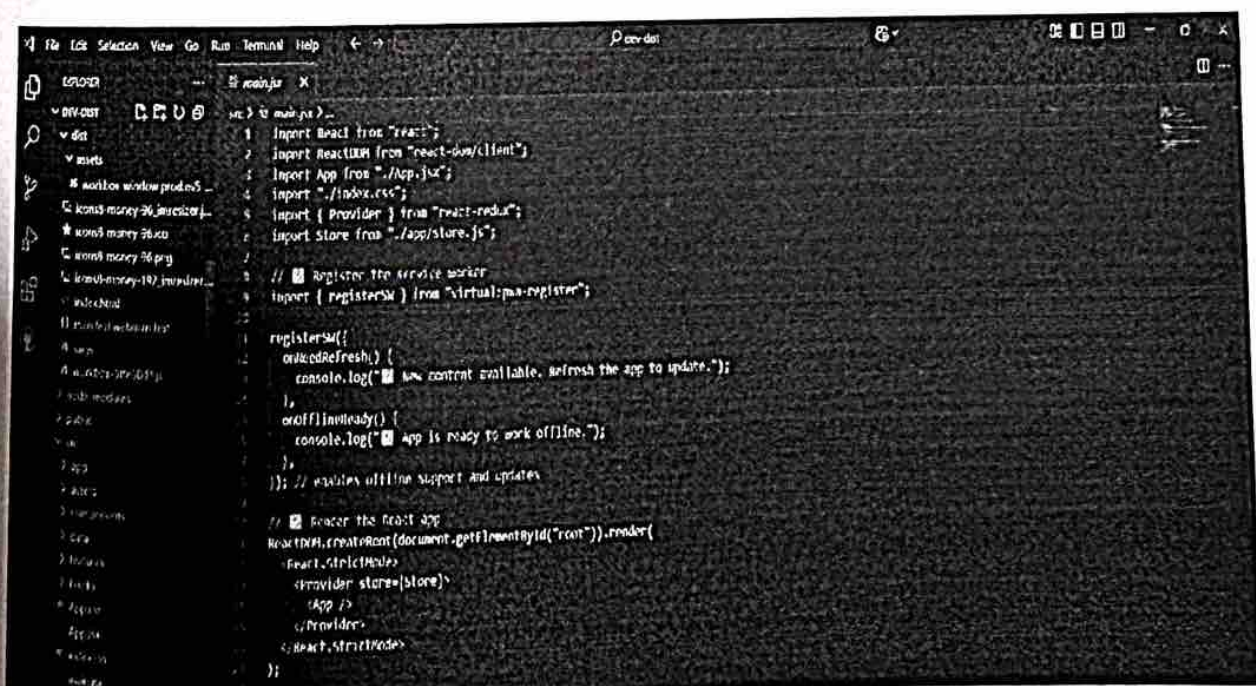
1. Open main.jsx (or wherever React renders the app). 2. Import the registerSW function from virtual:pwa-register. 3. Call registerSW() to activate the service worker.

4. This ensures that the app works offline and updates correctly.

2 Verify PWA Installation

1. Start the Vite development server.

2. Open Chrome DevTools (F12) → Go to Application → Manifest.

3. Check if the Install button appears in the address bar. 4. Install the app and check if it works like a standalone application.

3 Test Service Worker Activation

1. Open Chrome DevTools → Application → Service Workers.

2. Check if the service worker is registered.

3. Enable Offline mode and refresh the page to test caching.

4. If the app loads, the service worker is working successfully!

dev-dist

- JS registerSW.js
- JS sw.js
- JS workbox-417a7fbf.js

File Edit Selection View Go Run Terminal Help — dev-dist

EXPLORER — JS registerSW.js ✕

dev-dist > JS registerSW.js > …

```
1  if('serviceWorker' in navigator) navigator.serviceWorker.register("/dev-sw.js?dev-sw", { scope: "/", type: "classic" })
```

File Edit Selection View Go Run Terminal Help — dev-dist

EXPLORER — JS sw.js ✕

dev-dist > JS sw.js > …

```
1  /*
2  * Copyright 2018 Google Inc. All Rights Reserved.
3  * Licensed under the Apache License, Version 2.0 (the "License");
4  * you may not use this file except in compliance with the License.
5  * You may obtain a copy of the License at
6  *      http://www.apache.org/licenses/LICENSE-2.0
7  * Unless required by applicable law or agreed to in writing, software
8  * distributed under the License is distributed on an "AS IS" BASIS,
9  * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
10 * See the License for the specific language governing permissions and
11 * limitations under the License.
12 */
13
14 // If the loader is already loaded, just stop.
15 if (!self.define) {
16   let registry = {};
17
18   // Used for `eval` and `importScripts` where we can't get script URL by other means.
19   // In both cases, it's safe to use a global var because those functions are synchronous.
20   let nextDefineUri;
21
22   const singleRequire = (uri, parentUri) => {
23     uri = new URL(uri + ".js", parentUri).href;
24     return registry[uri] || (
25
26       new Promise(resolve => {
27         if ("document" in self) {
28           const script = document.createElement("script");
29           script.src = uri;
30           script.onload = resolve;
31           document.head.appendChild(script);
32         } else {
```

Explorer files: dev-dist, registerSW.js, sw.js, workbox-417a7fbf.js, dist, node_modules, public, src, app, assets, components, data, features, hooks, App.css, App.jsx, index.css, main.jsx, sw.js, .eslintrc.cjs, .gitignore, index.html, package-lock.json, package.json, README.md, vite.config.js

---

Install App

$ CURRENCY CONVERTER .99

From USD
United States Dollar

USA

USD United States

Enter Amount

⇅

To INR
Indian Rupee

Application
- Manifest
- Service workers
- Storage

Storage
- Local storage
- Session storage
- Extension storage
- IndexedDB
- Cookies
- Private state tokens
- Interest groups
- Shared storage
- Cache storage
- Storage buckets

Background services
- Back/forward cache
- Background fetch
- Background sync
- Bounce tracking mit...
- Notifications
- Payment handler
- Periodic background...
- Speculative loads
- Push messaging

Service workers
☐ Offline ☐ Update on reload

http://localhost:5173/

Source: service-w...
received 4/30/2025, 11:39:34 AM

Status: ● #4035 activated and is running   stop

Push: test push message from devtools

Sync: test-tag-from-devtools

Periodic sync: test-tag-from-devtools

Update Cycle:
Version	Update Activity	Timeline
#4035	Install	
#4035	Wait	
#4035	Activate	

Service workers from other origins