Experiment 08

Name	Roshan Bhagtani
Roll no.	4
Class	D15C
DOP	
DOS	
Grade	
Sign	

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 (Progressive Web App).

Theory: A **Service Worker** is a background script that acts as a proxy between a web application, the browser, and the network. It enables advanced features like **offline access**, **caching**, **push notifications**, and **background sync**, which are essential for building **Progressive Web Apps (PWAs)**.

Key features of a PWA include:

- Offline support
- Improved performance through caching
- App-like behavior
- Ability to work independently of network conditions

Service workers follow a **lifecycle**, consisting of the following main events:

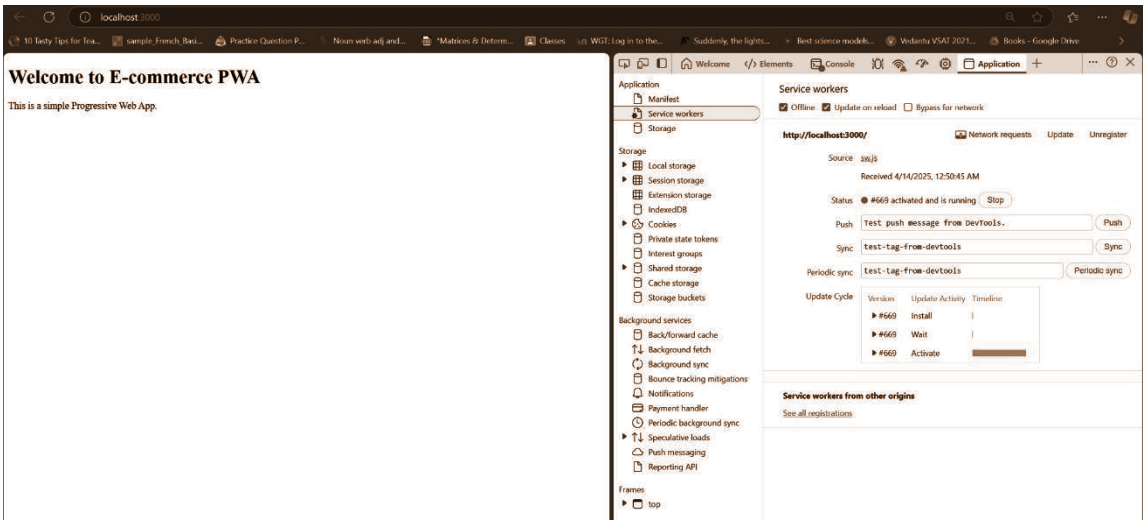
1. **Install**
2. **Activate**
3. **Fetch**

Implementation:

1. **`navigator.serviceWorker.register()`:** This method registers the service worker when the page loads. It checks if service workers are supported in the browser and then registers `sw.js` located at the root level.

2. **`self.addEventListener('install')`:** Triggered when the service worker is first installed. This is where assets are typically cached for offline use. In this case, we simply log the installation and immediately activate using `self.skipWaiting()`.

Output:



Conclusion:

Through this implementation, we successfully:

- Registered and activated a service worker
- Understood and used lifecycle events: install, activate, fetch
- Observed basic PWA behavior through browser DevTools
- Laid the foundation for caching strategies and offline support in E-commerce PWAs

This setup ensures that our PWA behaves more like a native app and provides a better user experience, especially in low or no connectivity environments.