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:
In Progressive Web Apps (PWAs), a service worker is a script that runs in the background of the browser, separate from the web page. It helps enhance user experience by enabling features like offline access, background sync, and push notifications. The service worker works using a lifecycle with three main stages:

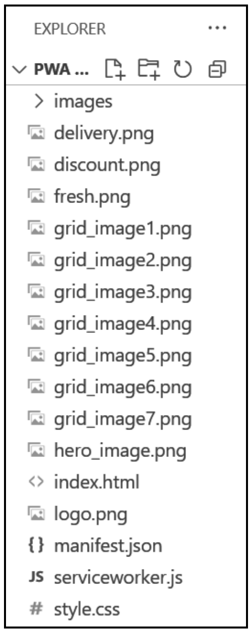
1. Install – This is when the service worker is first registered and used to cache important files.
2. Activate – After installation, the service worker takes control and clears old caches if needed.
3. Fetch – It intercepts network requests and serves responses from the cache when offline.

What We Implemented in This Experiment
In our E-commerce PWA project, we performed the following:

- Registered a Service Worker in index.html using JavaScript. This ensures that the browser knows where the service worker file (serviceworker.js) is located.
- In serviceworker.js:
  - We defined a cache name and listed essential files (index.html, style.css, manifest.json, and icons) that need to be stored in cache.
  - During the install event, we used cache.addAll() to store these files so that they can load offline.
  - In the fetch event, the service worker checks if a requested file is in the cache. If it is, it serves the cached version; if not, it fetches from the network.

This implementation allows the app to load essential content even without an internet connection, improving reliability and user experience.

Folder Structure:



```
> images
delivery.png
discount.png
fresh.png
grid_image1.png
grid_image2.png
grid_image3.png
grid_image4.png
grid_image5.png
grid_image6.png
grid_image7.png
hero_image.png
index.html
logo.png
manifest.json
serviceworker.js
style.css
```

Code:

index.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0"
/>
    <meta name="theme-color" content="#007BFF">
    <title>Restaurant Website</title>
    <link rel="stylesheet" href="style.css" />
    <link rel="manifest" href="manifest.json">
    <script
      src="https://kit.fontawesome.com/7a4b62b0a4.js"
      crossorigin="anonymous"
    ></script>
  </head>
  <body>

    <script>
        if ('serviceWorker' in navigator) {
            navigator.serviceWorker.register('serviceworker.js')
                .then(() => console.log("Service Worker Registered"))
                .catch(error => console.log("Service Worker Registration
Failed", error));
        }
    </script>
  </body>
</html>
```

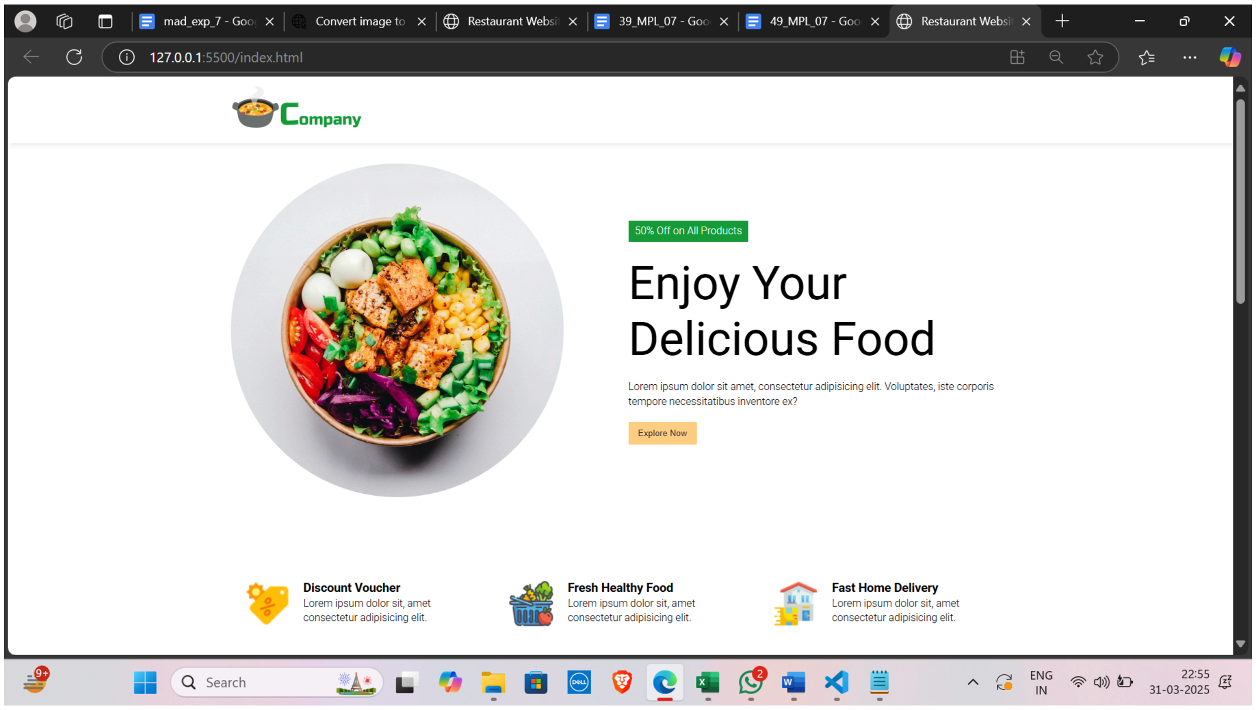
serviceworker.js

```
const CACHE_NAME = "cooking-cache-v1";
const FILES_TO_CACHE = [
    "index.html",
    "style.css",
    "manifest.json",
    "images/app.png",
    "images/big.png"
];

self.addEventListener("install", event => {
    event.waitUntil(
        caches.open(CACHE_NAME).then(cache => {
            return cache.addAll(FILES_TO_CACHE);
        })
    );
});

self.addEventListener("fetch", event => {
    event.respondWith(
        caches.match(event.request).then(response => {
            return response || fetch(event.request);
        })
    );
});
```

Screenshot:



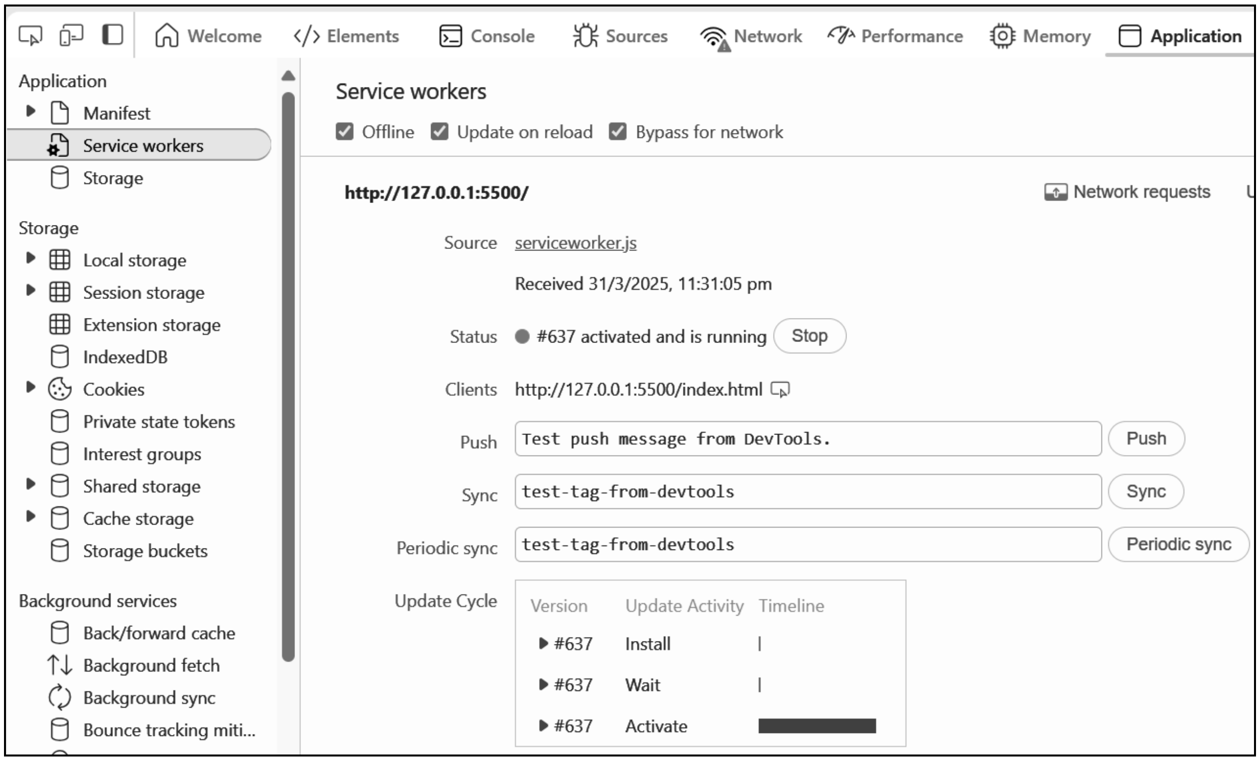
Enjoy Your Delicious Food

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Voluptates, iste corporis tempore necessitatibus inventore ex?

Discount Voucher
Lorem ipsum dolor sit, amet consectetur adipisicing elit.

Fresh Healthy Food
Lorem ipsum dolor sit, amet consectetur adipisicing elit.

Fast Home Delivery
Lorem ipsum dolor sit, amet consectetur adipisicing elit.



Service workers

☑ Offline ☑ Update on reload ☑ Bypass for network

http://127.0.0.1:5500/

Source serviceworker.js

Received 31/3/2025, 11:31:05 pm

Status ● #637 activated and is running

Clients http://127.0.0.1:5500/index.html

Push `Test push message from DevTools.`

Sync `test-tag-from-devtools`

Periodic sync `test-tag-from-devtools`

Update Cycle

Version	Update Activity	Timeline
#637	Install	|
#637	Wait	|
#637	Activate	▬

Conclusion

In this experiment, we successfully implemented and registered a service worker to enable offline support for our E-commerce website by caching essential files. Initially, we faced issues with incorrect file paths and browser caching, but we resolved them by verifying resource locations and updating the cache version for proper service worker activation.