

EXPERIMENT 08

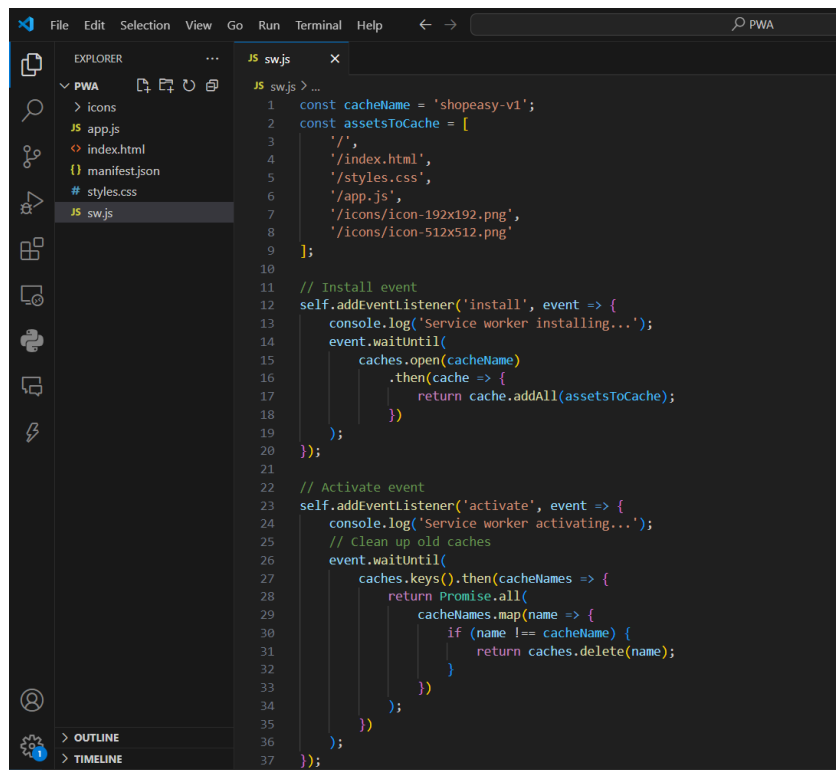
NAME : Hitesh Tanwani

CLASS : D15B

ROLL_NO : 57

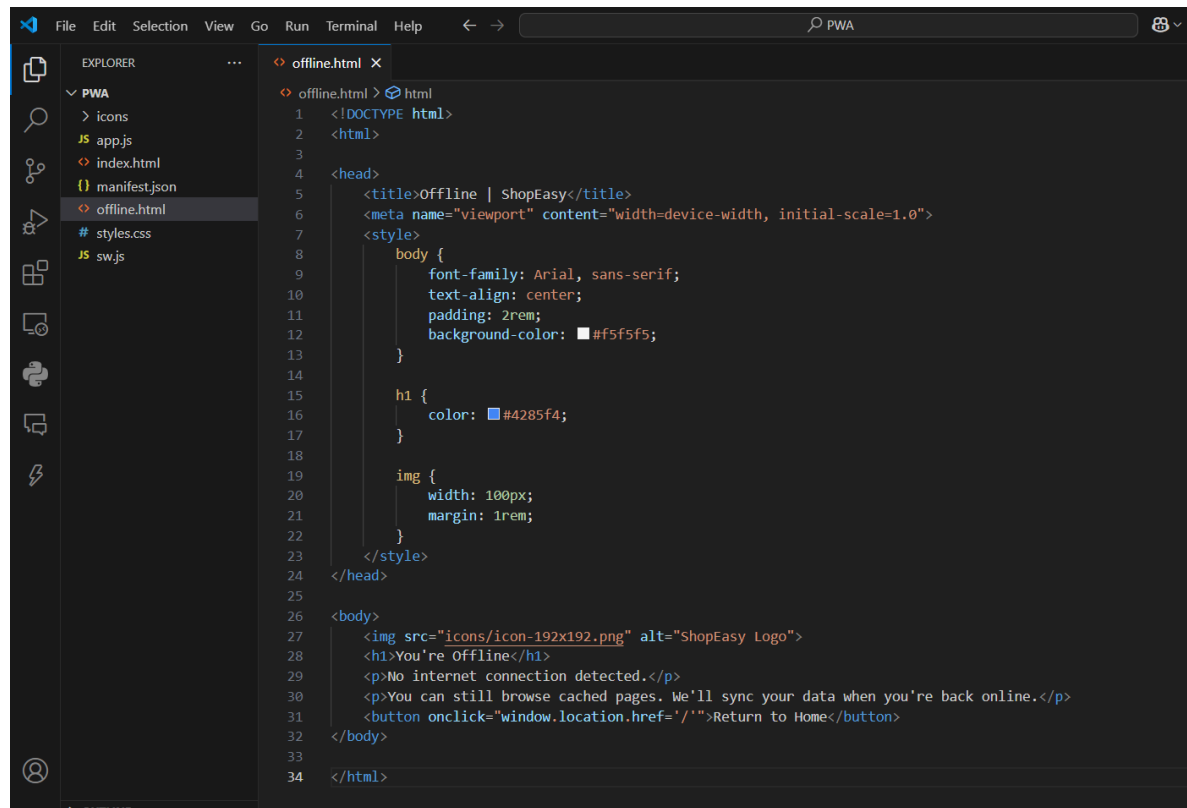
Aim: To code and register a service worker, and complete the install and activate process for a new service worker for the E-commerce PWA.

Create service-worker.js

The image shows a screenshot of the Visual Studio Code editor interface. On the left, the Explorer sidebar shows a project named 'PWA' with files including 'icons', 'app.js', 'index.html', 'manifest.json', 'styles.css', and 'sw.js'. The 'sw.js' file is selected and open in the main editor. The code in 'sw.js' defines a service worker with a cache name 'shopeasy-v1' and a list of assets to cache. It includes event listeners for 'install' and 'activate'. The 'install' event listener logs 'Service worker installing...', waits for the cache to be open, and then adds all assets to the cache. The 'activate' event listener logs 'Service worker activating...', waits for old caches to be cleaned up, and then returns. The code is as follows:

```
1  const cacheName = 'shopeasy-v1';
2  const assetsToCache = [
3    '/',
4    '/index.html',
5    '/styles.css',
6    '/app.js',
7    '/icons/icon-192x192.png',
8    '/icons/icon-512x512.png'
9  ];
10
11 // Install event
12 self.addEventListener('install', event => {
13   console.log('Service worker installing...');
14   event.waitUntil(
15     caches.open(cacheName)
16       .then(cache => {
17         return cache.addAll(assetsToCache);
18       })
19   );
20 });
21
22 // Activate event
23 self.addEventListener('activate', event => {
24   console.log('Service worker activating...');
25   // Clean up old caches
26   event.waitUntil(
27     caches.keys().then(cacheNames => {
28       return Promise.all(
29         cacheNames.map(name => {
30           if (name !== cacheName) {
31             return caches.delete(name);
32           }
33         })
34       );
35     })
36   );
37 });
```

Create a cacheable file called `offline.html` to be displayed in the absence of an internet connection.



The screenshot shows the Visual Studio Code editor with a file named `offline.html` open. The Explorer sidebar on the left shows the project structure for a PWA, including `icons`, `app.js`, `index.html`, `manifest.json`, `offline.html` (selected), `styles.css`, and `sw.js`. The main editor area displays the following HTML code:

```
1 <!DOCTYPE html>
2 <html>
3
4 <head>
5   <title>Offline | ShopEasy</title>
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <style>
8     body {
9       font-family: Arial, sans-serif;
10      text-align: center;
11      padding: 2rem;
12      background-color: #f5f5f5;
13    }
14
15    h1 {
16      color: #4285f4;
17    }
18
19    img {
20      width: 100px;
21      margin: 1rem;
22    }
23  </style>
24 </head>
25
26 <body>
27   
28   <h1>You're Offline</h1>
29   <p>No internet connection detected.</p>
30   <p>You can still browse cached pages. We'll sync your data when you're back online.</p>
31   <button onclick="window.location.href='/'">Return to Home</button>
32 </body>
33
34 </html>
```

