**Name: Aastha Jajoo** **Class: D15B** **Roll No: 23**

# Experiment No: 8

**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 :**
A service worker is a type of web worker that runs in the background of a web application, separate from the main browser thread. It is written in JavaScript and provides a way to run scripts in the background, independent of the user interface, allowing web applications to support offline functionality, push notifications, background sync, and more.
Here are some key points about service workers:

1. **Offline Support:** Service workers enable web applications to work offline by caching resources like HTML, CSS, JavaScript, and images. When the user is offline, the service worker can intercept network requests and serve cached content, providing a seamless offline experience.

2. **Push Notifications:** Service workers can receive push notifications from a server even when the web application is not open in the browser. This allows web apps to send notifications to users, similar to native mobile apps.

3. **Background Sync:** Service workers can schedule background syncs to send data to a server when the device is online, even if the web app is not actively being used. This is useful for applications that need to periodically sync data with a server.

4. **Improved Performance:** By offloading tasks to a separate thread, service workers can improve the performance of web applications. They can handle resource-intensive operations without blocking the main thread, leading to smoother user experiences.

5. **Security:** Service workers run in a separate context from the main page, which enhances security by preventing malicious scripts from accessing sensitive information or modifying the DOM directly.

**Name: Aastha Jajoo** **Class: D15B** **Roll No: 23**

**Steps :**

1. **Create a Service Worker File:**

- Create a new JavaScript file for your service worker, for example, service-worker.js.
- Define the service worker code inside this file. The code will include event listeners for install and activate events, as well as caching strategies.

```
JS serviceworker.js > ...
1   var staticCacheName = "pwa";
2
3   self.addEventListener("install", function (e) {
4   e.waitUntil(
5       caches.open(staticCacheName).then(function (cache) {
6       return cache.addAll(["/"]);
7       })
8   );
9   });
10
11  self.addEventListener("fetch", function (event) {
12  console.log(event.request.url);
13
14  event.respondWith(
15      caches.match(event.request).then(function (response) {
16      return response || fetch(event.request);
17      })
18  );
19  });
```

2. **Register the Service Worker:**
   In your main HTML file (e.g., index.html), add a script tag to register the service worker. Place the following code inside your HTML **file:**

```
<script>
    if ('serviceWorker' in navigator) {
      window.addEventListener('load', function() {
```

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

3. **Serviceworker code :**

```
JS serviceworker.js > ...
1   var staticCacheName = "pwa";
2
3   self.addEventListener("install", function (e) {
4   e.waitUntil(
5       caches.open(staticCacheName).then(function (cache) {
6       return cache.addAll(["/"]);
7       })
8   );
9   });
10
11  self.addEventListener("fetch", function (event) {
12  console.log(event.request.url);
13
14  event.respondWith(
15      caches.match(event.request).then(function (response) {
16      return response || fetch(event.request);
17      })
18  );
19  });
```

4. **Testing:**

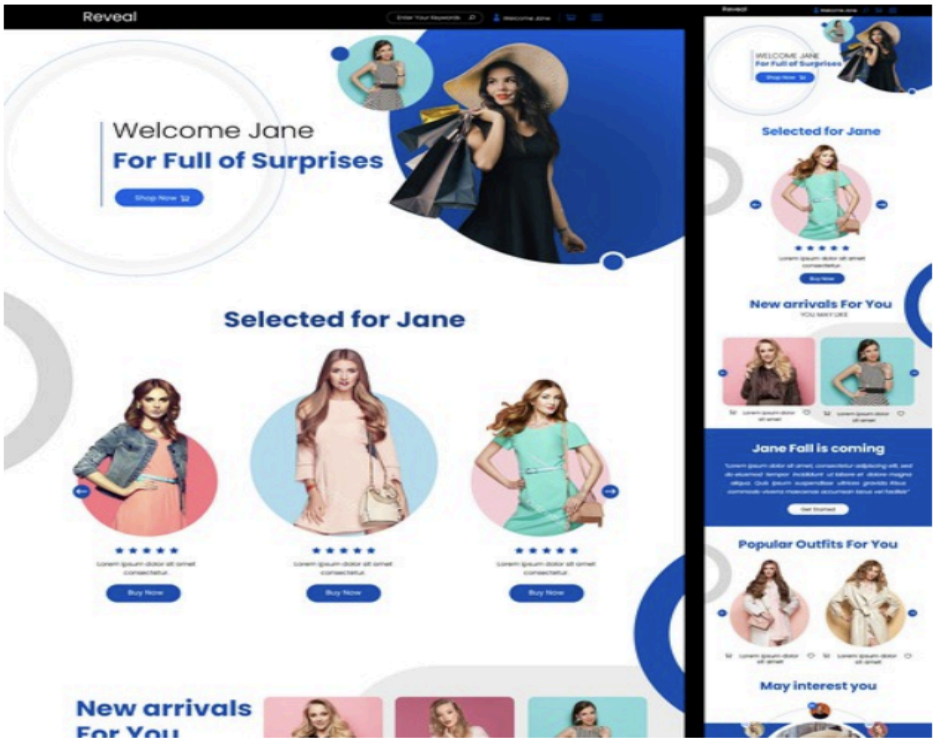
Save all your files and load your E-commerce PWA in a browser that supports service workers (e.g., Chrome, Firefox).

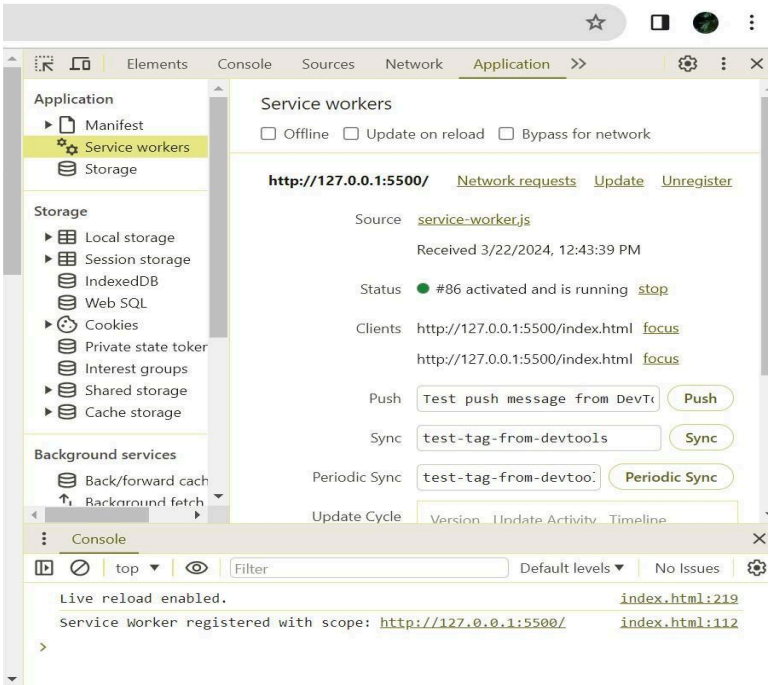
Open the developer tools (F12 or Ctrl+Shift+I), go to the Application tab, and check the Service Workers section to see if your service worker is registered and active.

Test the offline functionality by disconnecting from the internet and reloading the page. Ensure that cached content is served when offline.

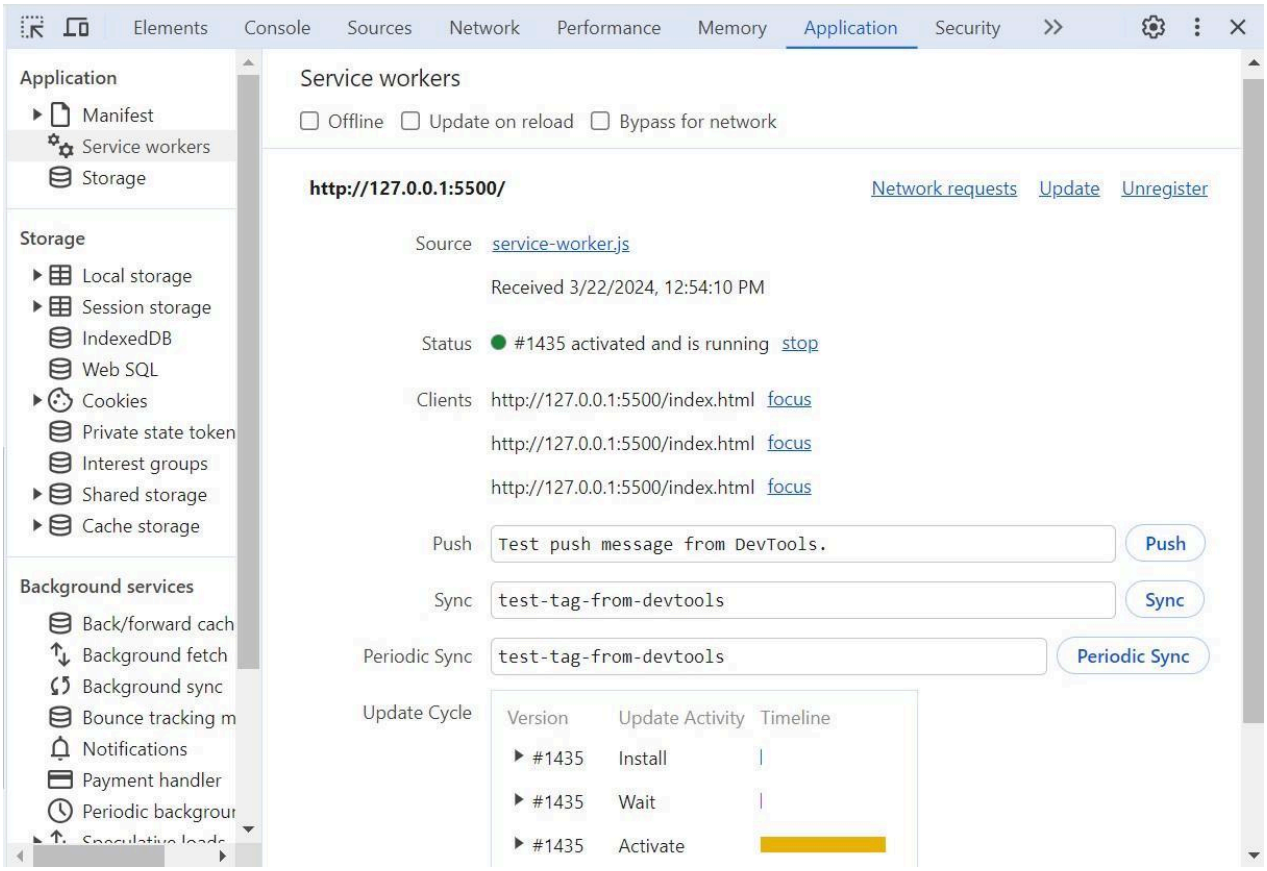
**Name: Aastha Jajoo** **Class: D15B** **Roll No: 23**

# Clothes Ecommerce Shop



**Name: Aastha Jajoo** **Class: D15B** **Roll No: 23**




**CONCLUSION :** Hence we have understood the working of PWA applications and learnt how to deploy Progressive web applications. Also we have coded and registered a service worker, and completed the installation and activation process for a new service worker for the Clothes E-commerce PWA.