PRACTICAL 8

Name : Prachit P. Paralikar Class: D15A Roll No: 43

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:

Service Worker

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 page, manage push notifications and develop “offline first” web applications with Cache API.

Things to note about Service Worker:

- A service worker is a programmable network proxy that lets you control how network requests from your page are handled.
- Service workers only run over HTTPS. Because service workers can intercept network requests and modify responses, "man-in-the-middle" attacks could be very bad.
- The service worker becomes idle when not in use and restarts when it's next needed. You cannot rely on a global state persisting between events. If there is information that you need to persist and reuse across restarts, you can use IndexedDB databases.

What can we do with Service Workers?

- You can dominate Network Traffic
  You can manage all network traffic of the page and do any manipulations. For example, when the page requests a CSS file, you can send plain text as a response or when the page requests an HTML file, you can send a png file as a response. You can also send a true response too.

- You can Cache
  You can cache any request/response pair with Service Worker and Cache API and you can access these offline content anytime.

- You can manage Push Notifications
  You can manage push notifications with Service Worker and show any information message to the user.

- You can Continue
  Although Internet connection is broken, you can start any process with Background Sync of Service Worker.

What can’t we do with Service Workers?

- You can’t access the Window
  You can’t access the window, therefore, You can’t manipulate DOM elements.

But, you can communicate to the window through post Message and manage processes that you want.

- You can’t work it on 80 Port
  Service Worker just can work on HTTPS protocol. But you can work on localhost during development.

Service Worker Cycle

A service worker goes through three steps in its life cycle:

- Registration
- Installation
- Activation

Registration
To install a service worker, you need to register it in your main JavaScript code. Registration tells the browser where your service worker is located, and to start installing it in the background. Let's look at an example:

main.js

```
if ('serviceWorker' in navigator) {
navigator.serviceWorker.register('/service-worker.js')
.then(function(registration) {
console.log('Registration successful, scope is:', registration.scope);
})
.catch(function(error) {
console.log('Service worker registration failed, error:', error);
});
}
```

This code starts by checking for browser support by examining navigator.serviceWorker. The service worker is then registered with navigator.serviceWorker.register, which returns a promise that resolves when the service worker has been successfully registered. The scope of the service worker is then logged with registration.scope. If the service worker is already installed, navigator.serviceWorker.register returns the registration object of the currently active service worker.

The scope of the service worker determines which files the service worker controls, in other words, from which path the service worker will intercept requests. The default scope is the location of the service worker file, and extends to all directories below. So if service-worker.js is located in the root directory, the service worker will control requests from all files at this domain.

You can also set an arbitrary scope by passing in an additional parameter when registering. For example:
main.js

navigator.serviceWorker.register('/service-worker.js', { scope: '/app/'
});

In this case we are setting the scope of the service worker to /app/, which means the service worker will control requests from pages like /app/, /app/lower/ and /app/lower/lower, but not from pages like /app or /, which are higher.

If you want the service worker to control higher pages e.g. /app (without the trailing slash) you can indeed change the scope option, but you'll also need to set the Service-Worker-Allowed HTTP Header in your server config for the request serving the service worker script.

main.js

navigator.serviceWorker.register('/app/service-worker.js', { scope: '/app'
});

Installation
Once the browser registers a service worker, installation can be attempted. This occurs if the service worker is considered to be new by the browser, either because the site currently doesn't have a registered service worker, or because there is a byte difference between the new service worker and the previously installed one.

A service worker installation triggers an install event in the installing service worker. We can include an install event listener in the service worker to perform some task when the service worker installs. For instance, during the install, service workers can precache parts of a web app so that it loads instantly the next time a user opens it (see caching the application shell). So, after that first load, you're going to benefit from instant repeat loads and your time to interactivity is going to be even better in those cases. An example of an installation event listener looks like this:

service-worker.js

// Listen for install event, set callback
self.addEventListener('install', function(event) {
// Perform some task
});

Activation
Once a service worker has successfully installed, it transitions into the activation stage. If there are any open pages controlled by the previous service worker, the new service worker enters a waiting state. The new service worker only activates when there are no longer any pages loaded that are still using the old service worker. This ensures that only one version of the service worker is running at any given time.

When the new service worker activates, an activate event is triggered in the activating service worker. This event listener is a good place to clean up outdated caches (see the Offline Cookbook for an example).

service-worker.js

self.addEventListener('activate', function(event) {
// Perform some task
});

Once activated, the service worker controls all pages that load within its scope, and starts listening for events from those pages. However, pages in your app that were loaded before the service worker activation will not be under service worker control. The new service worker will only take over when you close and reopen your app, or if the service worker calls clients.claim(). Until then, requests from this page will not be intercepted by the new service worker. This is intentional as a way to ensure consistency in your site.

CODE AND OUTPUT:

1) Index.html

```
<!DOCTYPE html>
<html>
   <head>
      <title>Spa Bonjour</title>
      <meta name="viewport" content="width=device-width, initial-scale=1.0">
      <link rel="preconnect" href="https://fonts.gstatic.com">
      <link
href="https://fonts.googleapis.com/css2?family=Kaushan+Script&family=Poppins:wg
ht@500&display=swap" rel="stylesheet">
      <script
src="https://cdn.jsdelivr.net/gh/cferdinandi/smooth-scroll/dist/smooth-scroll.polyfills.m
in.js"></script>
      <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.c
ss"/>
      <link rel="stylesheet" href="styles.css">

  </head>
  <script>
        // Add event listener to execute code when page loads
        window.addEventListener('load', () => {
                // Call registerSW function when page loads
                registerSW();
        });

        // Register the Service Worker
        async function registerSW() {
                // Check if browser supports Service Worker
                if ('serviceWorker' in navigator) {
                        try {
                                // Register the Service Worker named 'serviceworker.js'
                                await navigator.serviceworker.register('serviceworker.js');
```

```
                        }
                        catch (e) {
                                // Log error message if registration fails
                                console.log('SW registration failed');
                        }
                }
        }
</script>

<!-- Google tag (gtag.js) -->
<script async
src="https://www.googletagmanager.com/gtag/js?id=G-0PRXPB85YQ"></script>
<script>
  window.dataLayer = window.dataLayer || [];
  function gtag(){dataLayer.push(arguments);}
  gtag('js', new Date());

  gtag('config', 'G-0PRXPB85YQ');
</script>
   <body>
      <!---1. Banner/Home Section-->
      <section id="banner">
        <img src="images/logo.png" alt="logo" class="logo">
        <div class="banner-text">
        <h1>Hair Salon</h1>
        <p>Style your hair is Style your life</p>
        <div class="banner-btn">
          <a href="#"><span></span> Find Out</a>
          <a href="#"><span></span> Read More</a>
        </div>
        </div>
        <div id="sideNav">
           <nav>
              <ul>
                 <li><a href="#banner">HOME</a></li>
                 <li><a href="#feature">FEATURES</a></li>
                 <li><a href="#service">SERVICES</a></li>
                 <li><a href="#testimonial">TESTIMONIALS</a></li>
                 <li><a href="#contact">CONTACT US</a></li>
              </ul>
           </nav>
        </div>
        <div id="menuBtn">
           <img src="images/menu.png" alt="menu logo" id="menu">
        </div>
      </section>

      <!--- 2. features Section-->
      <section id="feature">
```

```
        <div class="title-text">
          <p>Features</p>
          <h1>WHY CHOOSE US</h1>
        </div>
        <div class="feature-box">
          <div class="features">
            <h1>Experienced Staff</h1>
            <div class="features-desc">
              <div class="feature-icon"><i class="fa fa-shield"></i></div>
              <div class="feature-text"><p>Lorem ipsum dolor sit amet, consectetur
adipisicing elit.
                  Necessitatibus cumque harum tenetur nam nisi, ducimus, a totam
veniam magni
                  atque nihil error, sapiente quibusdam at corporis sit quae dolore
quidem!</p>
               </div>
            </div>
            <h1>Pre-Booking Online</h1>
            <div class="features-desc">
              <div class="feature-icon"><i class="fa fa-check-square-o"></i></div>
              <div class="feature-text"><p>Lorem ipsum dolor sit amet, consectetur
adipisicing elit.
                  Necessitatibus cumque harum tenetur nam nisi, ducimus, a totam
veniam magni
                  atque nihil error, sapiente quibusdam at corporis sit quae dolore
quidem!</p>
               </div>
            </div>
            <h1>Affordable Cost</h1>
            <div class="features-desc">
              <div class="feature-icon"><i class="fa fa-inr"></i></div>
              <div class="feature-text"><p>Lorem ipsum dolor sit amet, consectetur
adipisicing elit.
                  Necessitatibus cumque harum tenetur nam nisi, ducimus, a totam
veniam magni
                  atque nihil error, sapiente quibusdam at corporis sit quae dolore
quidem!</p>
               </div>
            </div>
          </div>
          <div class="feature-img">
            <img src="images/barber-man.jpg" alt="barber-man">
          </div>
        </div>
      </section>

      <!--- 3. Service Section-->
      <section id="service">
        <div class="title-text">
```

```
            <p>Services</p>
            <h1>WE PROVIDE BETTER</h1>
          </div>
          <div class="service-box">
            <div class="single-service"><img id="img" src="images/pic-1.jpg"
alt="pic1">
                <div class="overlay"></div>
                <div class="service-desc">
                  <h3>Hair Cutting</h3><hr>
                  <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Ullam
ipsa quod sit! Earum quis fuga ut ratione amet necessitatibus minus voluptates.</p>
                </div>
            </div>
            <div class="single-service"><img id="img" src="images/pic-2.jpg"
alt="pic1">
                <div class="overlay"></div>
                <div class="service-desc">
                  <h3>Hair Styling</h3><hr>
                  <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Ullam
ipsa quod sit! Earum quis fuga ut ratione amet necessitatibus minus voluptates.</p>
                </div>
            </div>
            <div class="single-service"><img id="img" src="images/pic-3.jpg"
alt="pic1">
                <div class="overlay"></div>
                <div class="service-desc">
                  <h3>Hair Dying</h3><hr>
                  <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Ullam
ipsa quod sit! Earum quis fuga ut ratione amet necessitatibus minus voluptates.</p>
                </div>
            </div>
            <div class="single-service"><img id="img" src="images/pic-4.jpg"
alt="pic1">
                <div class="overlay"></div>
                <div class="service-desc">
                  <h3>Dry Shampooing</h3><hr>
                  <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Ullam
ipsa quod sit! Earum quis fuga ut ratione amet necessitatibus minus voluptates.</p>
                </div>
            </div>
          </div>
        </section>

        <!---4. Testimonials Section-->
        <section id="testimonial">
          <div class="title-text">
            <p>Testimonials</p>
            <h1>WHAT OUR CLIENTS SAYS</h1>
          </div>
```

```
        <div class="testimonial-row">
          <div class="testimonial-col">
            <div class="user">
              <img src="images/img-1.jpg" src="image">
              <div class="user-info">
                <h4>Ken Pollard<i class="fa fa-twitter"></i></h4>
                <small>@kenpollard</small>
              </div>
            </div>
            <p>Lorem ipsum dolor sit, amet consectetur adipisicing elit.
              Voluptatum odit et ratione dolorem delectus nobis, sunt consectetur,
              quos eveniet nihil harum? Deserunt nam ex quo commodi, facilis
laudantium repellendus dolorem!</p>
                  <p>Lorem ipsum dolor sit, amet consectetur adipisicing elit.
              quos eveniet nihil harum? Deserunt nam ex quo commodi, facilis
laudantium repellendus dolorem!</p>
          </div>
          <div class="testimonial-col">
            <div class="user">
              <img src="images/img-3.jpg" src="image">
              <div class="user-info">
                <h4>Suriya<i class="fa fa-twitter"></i></h4>
                <small>@im.suriya</small>
              </div>
               </div>
      </section>

      <!---5. Contact Us section-->
      <section id="contact">

        </div>
        <div class="footer-row">
          <div class="footer-left">
            <h1>Opening Hours</h1>
            <p><i class="fa fa-clock-o"></i>Mon - Fri : 9:00 AM - 9:00 PM</p>
            <p><i class="fa fa-clock-o"></i>Sat - Sun : 10:00 AM - 11:00 PM</p>
          </div>
          <div class="footer-right">
            <h1>Get in Touch</h1>
            <p><i class="fa fa-map-marker"></i>#30 ABC Colony, XYZ City, IN</p>
            <p><i class="fa fa-paper-plane"></i>example@website.in</p>
            <p><i class="fa fa-map-marker"></i>+01 1123456782</p>
          </div>
        </div>
        <div class="social-links">
          <a href="#"><i class="fa fa-facebook"></i></a>
          <a href="#"><i class="fa fa-twitter"></i></a>
          <a href="#"><i class="fa fa-instagram"></i></a>
          <a href="#"><i class="fa fa-youtube-play"></i></a><br>
```

```
            <small> © All Copyrights are reserved to nerd__fswd.</small>
          </div>
        </section>
       <script src="script.js" type="text/javascript"></script>
<script type="module">
  // Import the functions you need from the SDKs you need
  import { initializeApp } from
"https://www.gstatic.com/firebasejs/10.8.0/firebase-app.js";
  import { getAnalytics } from
"https://www.gstatic.com/firebasejs/10.8.0/firebase-analytics.js";
  // TODO: Add SDKs for Firebase products that you want to use
  // https://firebase.google.com/docs/web/setup#available-libraries

  // Your web app's Firebase configuration
  // For Firebase JS SDK v7.20.0 and later, measurementId is optional
  const firebaseConfig = {
    apiKey: "AIzaSyCt0d-RSB3hgKMUU20QbbZA7CNRCkrXaMg",
    authDomain: "barber-985c6.firebaseapp.com",
    projectId: "barber-985c6",
    storageBucket: "barber-985c6.appspot.com",
    messagingSenderId: "3049260321",
    appId: "1:3049260321:web:b3a79fc45245ff78986a60",
    measurementId: "G-X7FB372LM2"
  };

  // Initialize Firebase
  const app = initializeApp(firebaseConfig);
  const analytics = getAnalytics(app);
</script>
   </body>

</html>
```

2) script.js

```
// Function to register the Service Worker
async function registerServiceWorker() {
    if ('serviceWorker' in navigator) {
        try {
            // Register the Service Worker named 'serviceworker.js'
            const registration = await
navigator.serviceWorker.register('serviceworker.js');
            console.log('Service Worker registered successfully:', registration);
        } catch (error) {
            console.error('Service worker registration failed:', error);
        }
    } else {
        console.log('Service workers are not supported in this browser.');
```

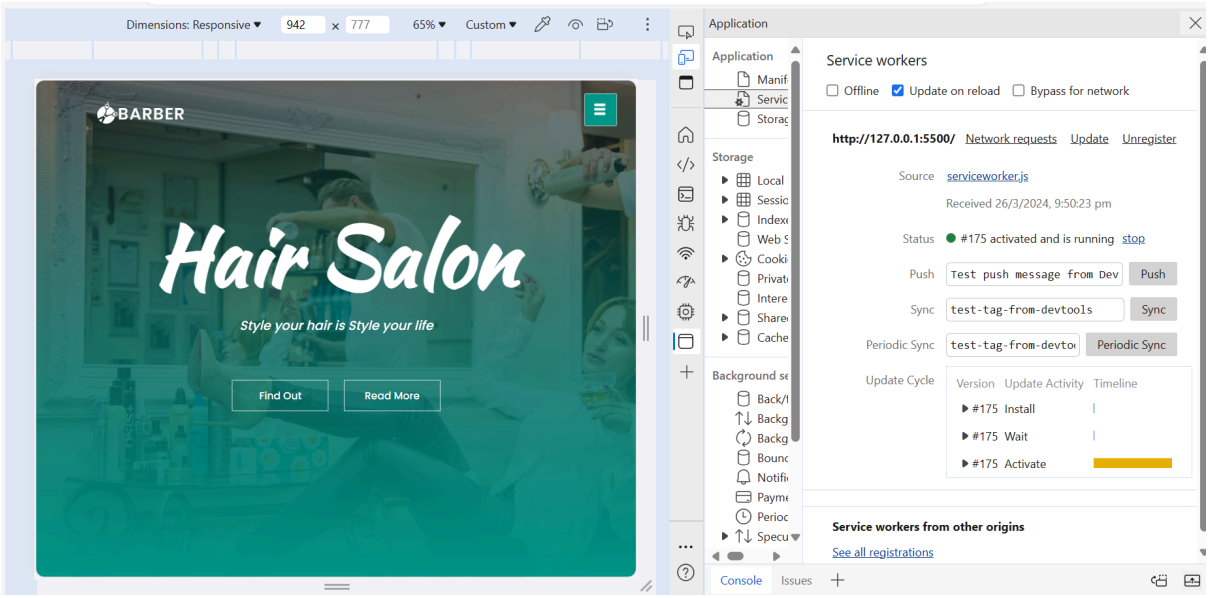
```
    }
}

// Add event listener to execute code when page loads
window.addEventListener('load', () => {
    // Call function to register Service Worker
    registerServiceWorker();

    // Rest of your existing script
    var menuBtn = document.getElementById('menuBtn');
    var sideNav = document.getElementById('sideNav');
    var menu = document.getElementById('menu');

    sideNav.style.right = "-250px";

    menuBtn.onclick = function(){
        if(sideNav.style.right == "-250px"){
            sideNav.style.right="0";
            menu.src = "images/close.png";
        }
        else{
            sideNav.style.right="-250px";
            menu.src = "images/menu.png";
        }
    }

    // All animations will take exactly 1000ms
    var scroll = new SmoothScroll('a[href*="#"]', {
        speed: 1000,
        speedAsDuration: true
    });
});
```

3) service-worker.js

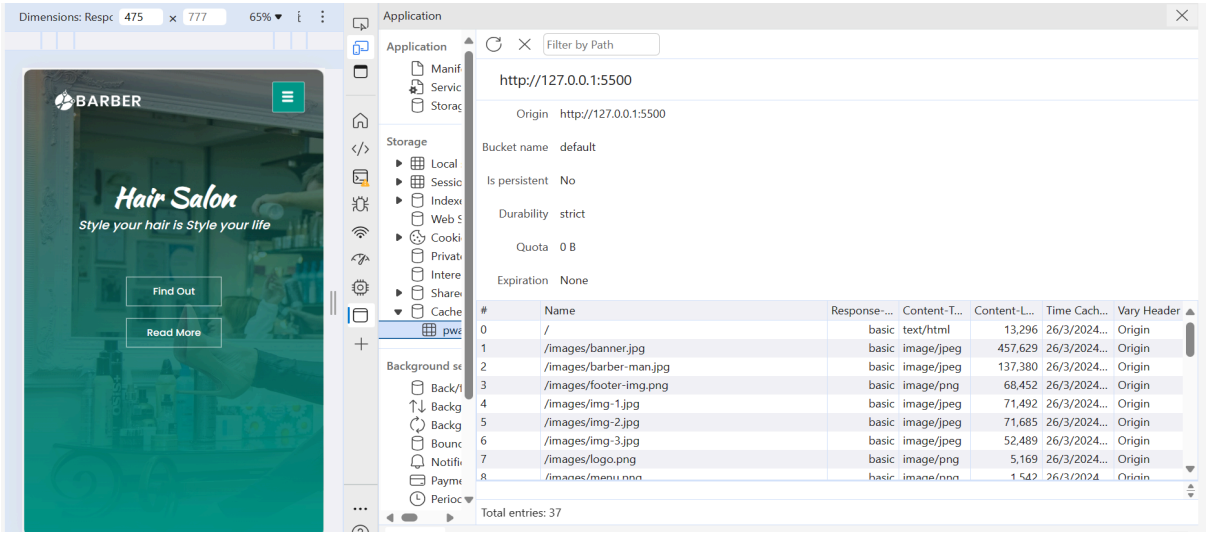
```
var staticCacheName = "pwa-v1"; // Update cache name to indicate a new version

// Installation
self.addEventListener("install", function (e) {
    console.log("Service Worker installed");
    e.waitUntil(
        caches.open(staticCacheName).then(function (cache) {
            console.log("Opened cache");
            return cache.addAll([
                "/", // Add paths to cache here
                "/index.html",
                "/styles.css",
                "/script.js",
                "/images/logo.png",
```

```
            // Add other static assets you want to cache
        ]);
    })
  );
});

// Activation
self.addEventListener("activate", function (e) {
  console.log("Service Worker activated");

  // Remove old caches
  e.waitUntil(
    caches.keys().then(function (cacheNames) {
      return Promise.all(
        cacheNames.filter(function (cacheName) {
          return cacheName !== staticCacheName;
        }).map(function (cacheName) {
          console.log("Deleting old cache:", cacheName);
          return caches.delete(cacheName);
        })
      );
    })
  );
});

// Fetch
self.addEventListener("fetch", function (event) {
  console.log("Fetch event for ", event.request.url);
  event.respondWith(
    caches.match(event.request).then(function (response) {
      if (response) {
        console.log("Found ", event.request.url, " in cache");
        return response;
      }
      console.log("Network request for ", event.request.url);
      return fetch(event.request).then(function (response) {
        // Add fetched responses to cache
        return caches.open(staticCacheName).then(function (cache) {
          console.log("Caching new resource:", event.request.url);
          cache.put(event.request.url, response.clone());
          return response;
        });
      });
    }).catch(function (error) {
      // Handle errors gracefully
      console.error("Error fetching data from cache: ", error);
    })
  );
});
```

Now Right click —> Inspect —> Application —> Service workers



Now scroll down to Cache Storage



Conclusion:

- Registered a service worker, and completed the installation and activation process for a new service worker.
- Also checked the Cache Storage