#### **Experiment No: 08**

**Aim**: To code and register a service worker, and complete the install and activation process for a new service worker for the Testmate PWA.

- A Service Worker (SW) is a script that runs in the background, separate from your web page.
- It enables features like offline caching, push notifications, and background sync.
- It doesn't have access to the DOM directly.
- Runs only on HTTPS (except on localhost during development).

#### Step 1: Code in main. js (Service Worker Registration)

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

• Place this script in your main.js and make sure it is **linked in your HTML file** using <script src="main.js"></script>.

# Step 2: Code in sw.js (Install + Activate + Fetch Handling)

```
// Caching important files
const filesToCache = [
   './components/Login.js',
   './components/Home.js',
];
```

```
const CACHE_NAME = 'offline';
// Install event
self.addEventListener("install", function (event) {
  event.waitUntil(
    caches.open(CACHE_NAME).then(function (cache) {
      return cache.addAll(filesToCache);
    })
 );
});
// Activate event (clean up old caches if needed)
self.addEventListener("activate", function (event) {
  console.log("Service worker activated.");
});
// Fetch event
self.addEventListener("fetch", function (event) {
  event.respondWith(
    fetch(event.request).then(function (response) {
      if (response.status !== 404) {
        return response;
      } else {
        return caches.match("offline.html");
    }).catch(function () {
      return caches.match(event.request).then(function (response) {
        return response || caches.match("offline.html");
      });
   })
  );
  event.waitUntil(
    caches.open(CACHE_NAME).then(function (cache) {
      return fetch(event.request).then(function (response) {
        return cache.put(event.request, response.clone());
      });
    })
 );
});
```

### Step 3: Run Your Project Locally

You can use a local server (HTTPS not required on localhost):

#### **%** If using VS Code:

Install Live Server Extension and run your project with Right click → "Open with Live Server"

Or use:

npx serve

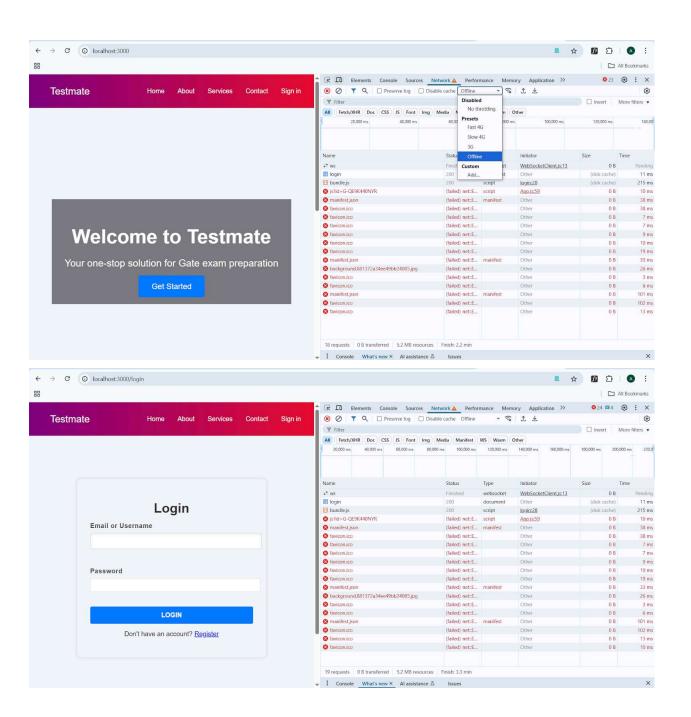
or

python3 -m http.server 8080

Then go to http://localhost:8080

## Step 4: Test It

- 1. Open DevTools  $\rightarrow$  Application tab  $\rightarrow$  Check if service worker is registered.
- 2. Go Offline (DevTools  $\rightarrow$  Network  $\rightarrow$  Offline).
- 3. Visit /home or /login  $\rightarrow$  Page should load from cache.
- 4. Try a non-cached URL  $\rightarrow$  Should show offline.html.



☐ All Bookmarks 88 Y Filter ☐ Invert More filters ▼ All Fetch/XHR Doc CSS JS Font Img Media Manifest WS Wasm Other 40,000 ms 50,000 ms 20,000 ms 70,000 ms 90,0 10,000 ms 30,000 ms Type websocket websocket websocket Initiator VM6:50632 Finished Finished Finished 200 200 ≠ ws ≠ ws ≠ ws 0 B 0 B 0 B VM6:50632 VM6:50632 11 ms 215 ms login
bundle.js script login:28 (failed) net:E... script (failed) net:E... manifest VM6:5792 Other favicon.ico
favicon.ico
favicon.ico
favicon.ico
favicon.ico **Testmate** 38 ms 7 ms 7 ms 9 ms 10 ms (failed) net:E... (failed) net::E... 0 B 0 B 0 B (failed) net:E... ➣ You're offline (failed) net:E... (failed) net:E... 08 08 08 08 (failed) net:E... (failed) net:E... manifest favicon.ico
 manifest.json Other Other 19 ms
33 ms
26 ms
3 ms
6 ms
12 ms
0 ms
1 ms
33 ms background.881372a34ee49bb24085.jpg
 favicon.ico
 favicon.ico Other Other Other (failed) net::E... (failed) net::E... 08

register
 dataimage/png/base...

 dataimage/png/base...

 register
 1 requests 0 8 transferred 5.2 MB resources

Console What's new × Al assistance ∆ Issues

(failed) net:E... document

200 png (canceled) document

chrome-error://chromeweb (memory cac...

VM9:1 Other