

Assignment No.: 08

Title: Service Worker Integration for Phone Validator PWA

Name: Harshit Raheja

Class: D15B

Roll Number: 45

**Aim:** To implement, register, and activate a service worker to enable offline functionality for the Phone Validator App PWA.

**Theory:** A service worker is a script that runs independently in the browser background, enabling advanced capabilities like caching, offline access, push notifications, and network request interception. Service workers only operate over HTTPS and go through three main lifecycle stages: registration, installation, and activation.

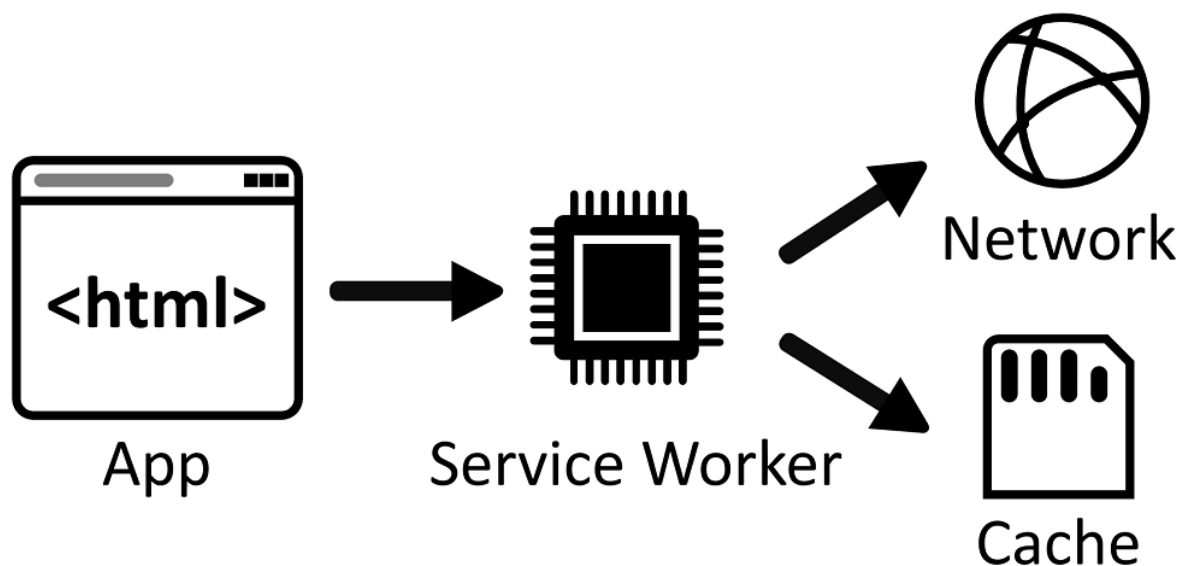
**Functionality of Service Workers:** 1. Intercept and modify network requests. 2. Cache essential resources for offline access. 3. Support background sync and push notifications.

**Limitations:** 1. Cannot access the DOM directly. 2. Runs only on HTTPS or localhost. 3. Limited to scope defined during registration.

**Service Worker Lifecycle: Registration:** Done in window.onload using navigator.serviceWorker.register().

**Installation:** Occurs when the SW is new or updated; uses install event to cache files.

**Activation:** Triggers activate event to finalize setup and remove old caches if needed.



## Code:

### 1. serviceworker.js

```
var staticCacheName = "phone-validator-pwa-v1";
self.addEventListener("install", function (e) {
  e.waitUntil(
    caches.open(staticCacheName).then(function (cache) {
      return cache.addAll([
        "/",
        "/index.html",
        "/manifest.json",
        "/serviceworker.js",
        "/images/phone-icon-192.png",
        "/images/phone-icon-512.png"
      ]);
    })
  );
});
self.addEventListener("fetch", function (event) {
  event.respondWith(
    caches.match(event.request).then(function (response) {
      return response || fetch(event.request);
    })
  );
});
```

### 2. manifest.json (Excerpt)

```
{
  "name": "Phone Validator App",
  "short_name": "PhoneCheck",
  "start_url": "index.html",
  "display": "standalone",
  "background_color": "#f4f6f8",
  "theme_color": "#2c3e50",
  "scope": ".",
  "description": "Check if a phone number is valid using Abstract API.",
  "icons": [
    {"src": "images/phone-icon-192.png", "sizes": "192x192", "type": "image/png"},
    {"src": "images/phone-icon-512.png", "sizes": "512x512", "type": "image/png"}
  ]
}
```

```
}
```

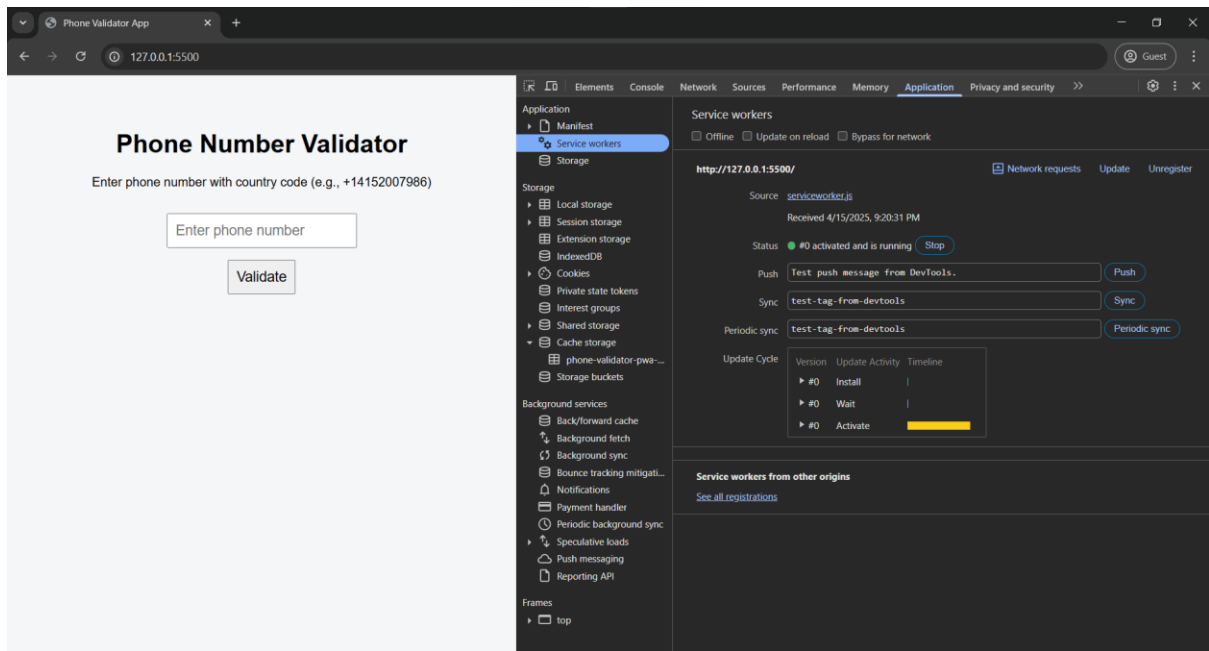
### 3. index.html (Excerpt)

```
<script>
window.addEventListener('load', () => {
  if ('serviceWorker' in navigator) {
    navigator.serviceWorker.register('serviceworker.js').catch(err => console.error('SW
registration failed:', err));
  }
});
</script>
```

### Screenshots:

The screenshot shows a web browser with the 'Phone Validator App' open. The app's interface includes a title 'Phone Number Validator', a subtitle 'Enter phone number with country code (e.g., +14152007986)', a text input field labeled 'Enter phone number', and a 'Validate' button. The browser's developer tools are open, displaying the 'Application' tab. The left sidebar shows the 'Storage' section expanded, with 'phone-validator-pw...' selected. The main pane shows the 'Cache storage' section, which is currently empty, displaying 'No cache entry selected' and 'Select a cache entry above to preview'. A table of cache entries is visible at the bottom of the pane.

#	Name	Response...	Content-T...	Content-T...	Time Cac...	Vary Head...
0	/	basic	text/html	2,494	4/15/202...	
1	/images/phone-icon-192.png	basic	image/png	2,990	4/15/202...	
2	/images/phone-icon-512.png	basic	image/png	9,588	4/15/202...	
3	/index.html	basic	text/html	2,494	4/15/202...	
4	/manifest.json	basic	applicatio...	982	4/15/202...	
5	/serviceworker.js	basic	applicatio...	697	4/15/202...	



**Output:** The service worker registers and pre-caches critical app files. If the user opens the app offline, cached files like index.html and icons are served. The app provides phone number validation using the Abstract API, with graceful offline behavior enabled by the service worker.