Name: **Abhinav Swaminathan**	Div-Roll no: **D15C-01**
DOP:	DOS:
Sign:	Grade:

Experiment 9

Aim: To implement Service worker events like fetch, sync and push for PWA.

Theory:

A **Service Worker** is a JavaScript file that runs in the background of a Progressive Web App (PWA). It acts as a proxy between the web app and the network, enabling features like:

- **Caching content for offline use** (`fetch` event)
- **Syncing data in the background** (`sync` event)
- **Receiving and displaying push notifications** (`push` event)

These service worker events significantly improve user experience by ensuring fast loading, real-time updates, and engagement, even in low or no internet connectivity.

Output:

1. Fetch Event

```
self.addEventListener('fetch', (event) => {
  event.respondWith(
    caches.match(event.request).then(response => {
      return response || fetch(event.request);
    })
  );
});
```

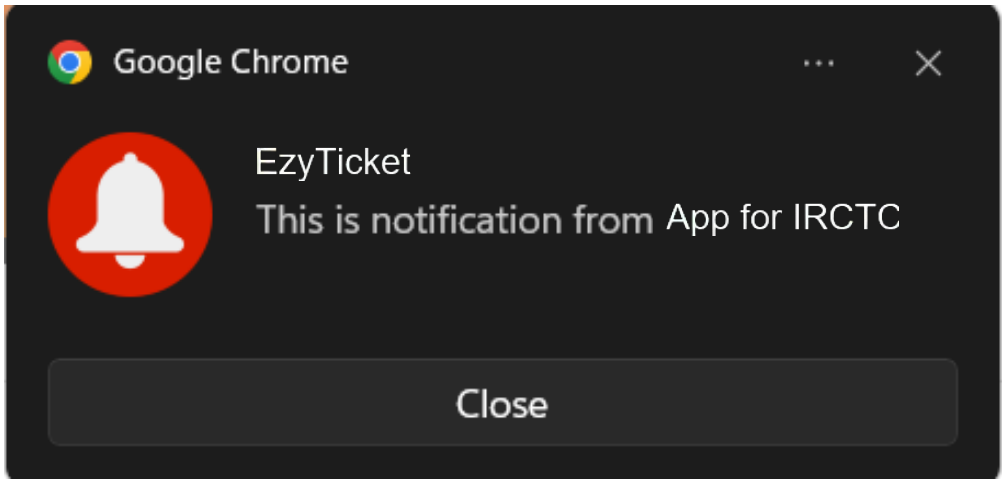
2. Sync Event

```
self.addEventListener('sync', (event) => {
  if (event.tag === 'sync-data') {
    event.waitUntil(syncDataWithServer());
  }
});
```

3. Push Event

```
self.addEventListener('push', (event) => {
  const data = event.data.json();
  self.registration.showNotification(data.title, {
    body: data.body,
    icon: 'icon.png'
  });
});
```

Notification:



Conclusion:

In this experiment, we successfully implemented the core **Service Worker events (`fetch`, `sync`, and `push`)** in the **railway PWA**. This enhanced the app's ability to:

- Work offline using cache (`fetch`)
- Automatically sync data in the background (`sync`)
- Engage users with notifications (`push`)

These features are crucial for improving **reliability**, **performance**, and **user engagement** in modern web applications.