**Experiment No 10**

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

**Code**:

***index.html***

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>GitHub page as PWA template</title>

<link rel="canonical" href="https://Dhotre12.github.io/github-page-pwa/" />

<link rel="manifest" href="/github-page-pwa/manifest.webmanifest">

<link rel="stylesheet" href="css/styles.css">

</head>

<body>

<h1>GitHub page as PWA template</h1>

<p>

This is a PWA template for GitHub pages.

Make sure to check the README at

<a href="https://github.com/Dhotre12/github-page-pwa/">https://github.com/Dhotre12/github-page-pwa/</a> and

to change the necessary settings to your needs.

</p>

<script>

if (navigator.serviceWorker) {

navigator.serviceWorker.register (

'/github-page-pwa/sw.js',

{scope: '/github-page-pwa/'}

)

}

</script>

<script src="js/app.js"></script>

</body>

</html>

***sw.js***

var GHPATH = '/github-page-pwa';

var APP\_PREFIX = 'gppwa\_';

var VERSION = 'version\_002';

var URLS = [

`${GHPATH}/`,

`${GHPATH}/index.html`,

`${GHPATH}/css/styles.css`,

`${GHPATH}/img/icon.png`,

`${GHPATH}/js/app.js`

]

var CACHE\_NAME = APP\_PREFIX + VERSION

self.addEventListener('fetch', function (e) {

console.log('Fetch request : ' + e.request.url);

e.respondWith(

caches.match(e.request).then(function (request) {

if (request) {

console.log('Responding with cache : ' + e.request.url);

return request

} else {

console.log('File is not cached, fetching : ' + e.request.url);

return fetch(e.request)

}

})

)

})

self.addEventListener('install', function (e) {

e.waitUntil(

caches.open(CACHE\_NAME).then(function (cache) {

console.log('Installing cache : ' + CACHE\_NAME);

return cache.addAll(URLS)

})

)

})

self.addEventListener('activate', function (e) {

e.waitUntil(

caches.keys().then(function (keyList) {

var cacheWhitelist = keyList.filter(function (key) {

return key.indexOf(APP\_PREFIX)

})

cacheWhitelist.push(CACHE\_NAME);

return Promise.all(keyList.map(function (key, i) {

if (cacheWhitelist.indexOf(key) === -1) {

console.log('Deleting cache : ' + keyList[i] );

return caches.delete(keyList[i])

}

}))

})

)

})

***manifest.webmanifest***

{

"name": "Github Page PWA",

"short\_name": "GPPWA",

"description": "Github Page as a Progressive Web App",

"scope": "/github-page-pwa/",

"start\_url": "/github-page-pwa/",

"background\_color": "#ffffff",

"theme\_color": "#ffffff",

"display": "standalone",

"icons": [

{

"src": "/github-page-pwa/img/icon.png",

"type": "image/png",

"sizes": "700x700"

}

]

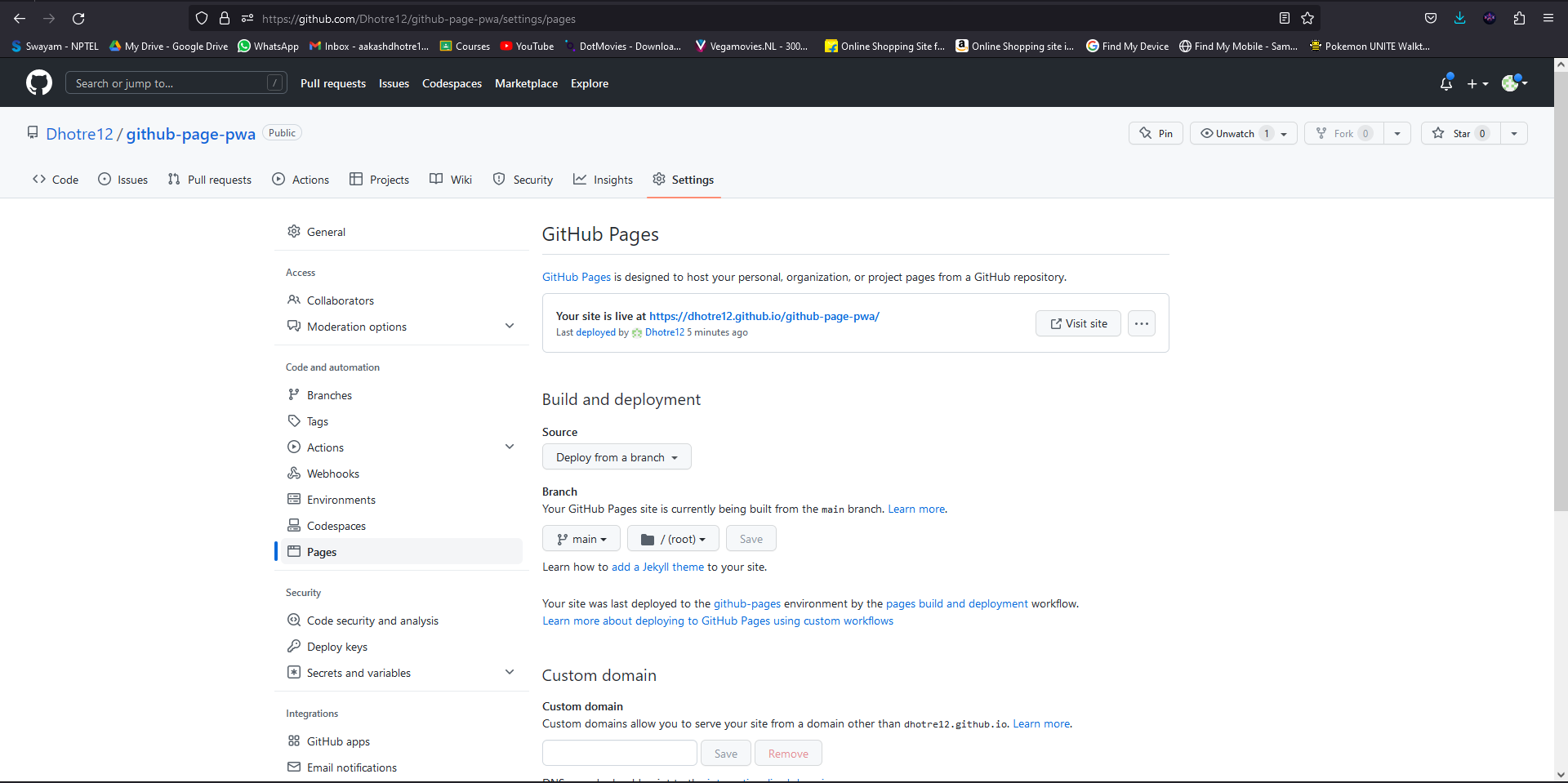
}

***js folder***

***app.js***

console.log('I am running!');

**Output**:

****

