

**Date :** 31 - 07 - 2020

**Morning Session :** 9 am – 11.00 PM

**By ~** Sundeep Charan Ramkumar Today

## **Topics: PWA (Progressive Web App)**

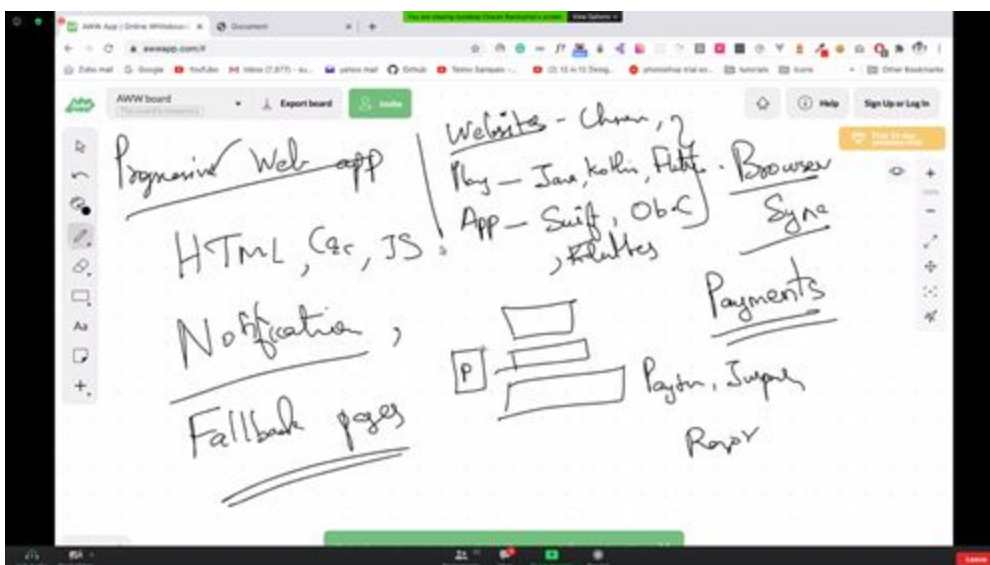
### **Progressive Web App:**

A PWA is a web application that can be “installed” on your system. It works offline when you don’t have an internet connection, leveraging data cached during your last interactions with the app.

You should make your web app into a PWA because it’ll reduce the time it takes for your app to load and it’ll give your users a better experience.

### **PWA's**

- Can be made using vanilla JS, HTML & CSS
- Accessed via a web address and not the app store
- Can be installed on the mobile home-screen
- Runs in the browser but with access to device features
- Can be used offline
- Can use web push notifications

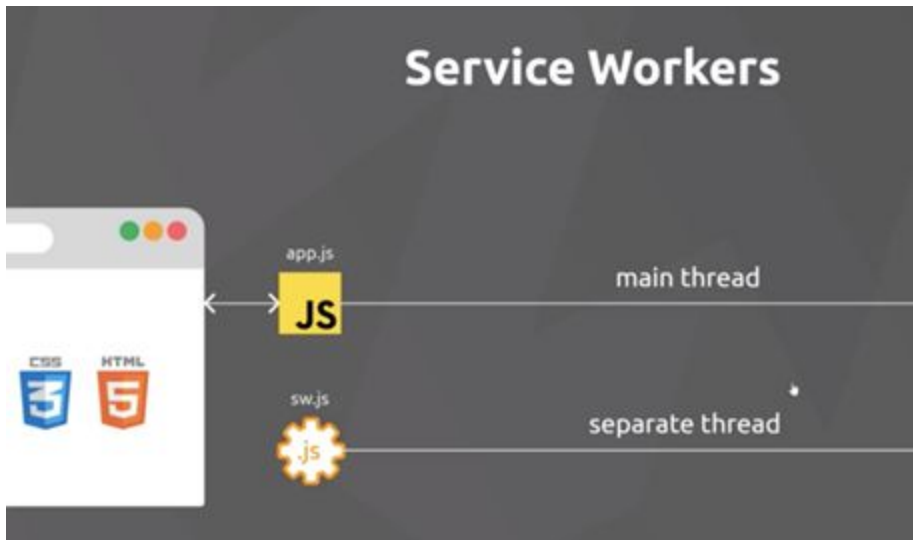


**We will try to convert this to PWA.**

## Service -Worker:

service workers run on a separate thread.

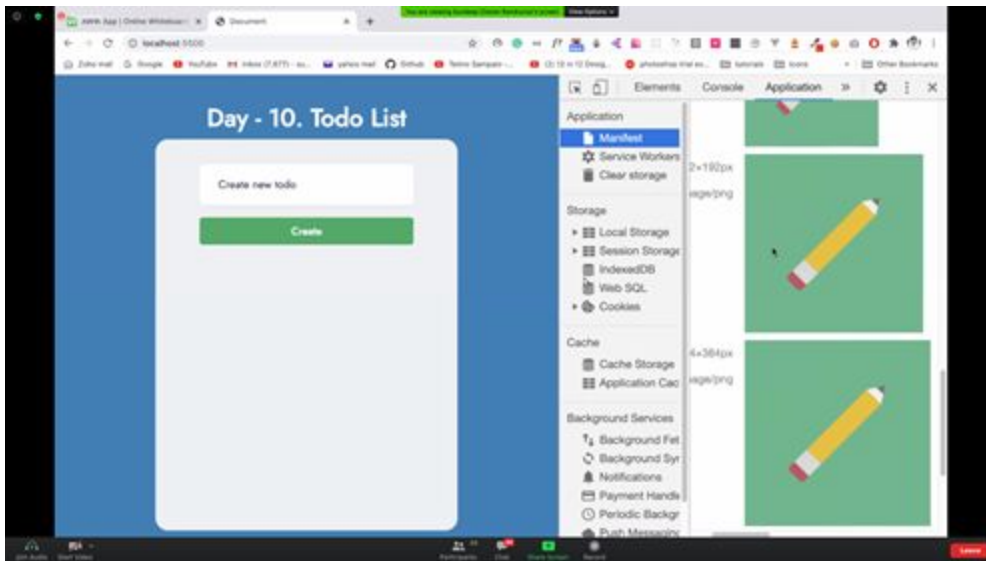
This means service workers don't have access to the DOM



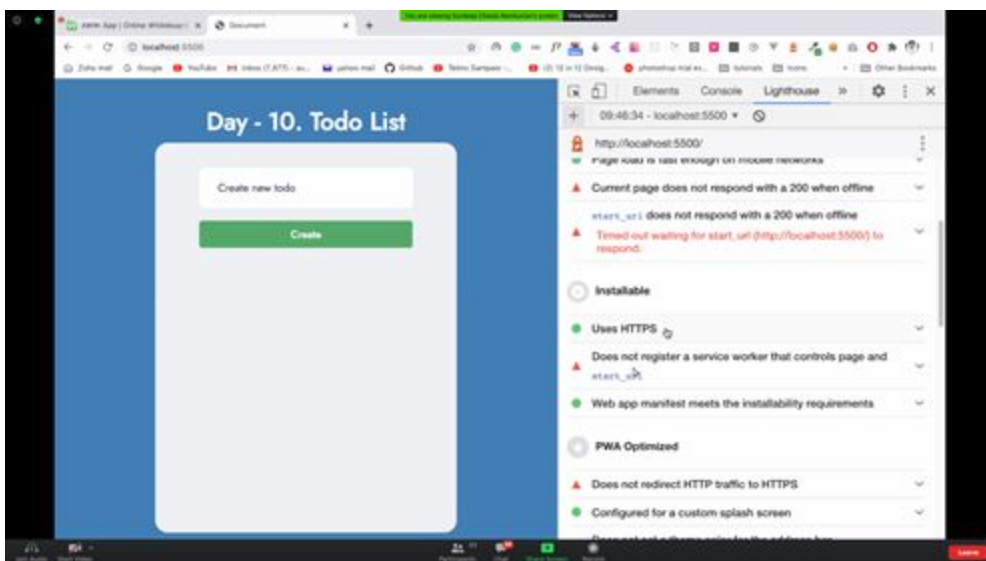
A screenshot of a code editor displaying a `manifest.json` file. The file contains the following JSON content:

```
1 {
2   "short_name": "React App",
3   "name": "Create React App Sample",
4   "icons": [
5     {
6       "src": "favicon.ico",
7       "sizes": "64x64 32x32 24x24 16x16",
8       "type": "image/x-icon"
9     },
10    {
11      "src": "logo192.png",
12      "type": "image/png",
13      "sizes": "192x192"
14    },
15    {
16      "src": "logo512.png",
17      "type": "image/png",
18      "sizes": "512x512"
19    }
20  ],
21   "start_url": ".",
22   "display": "standalone",
23   "theme_color": "#000000",
24   "background_color": "#ffffff"
25 }
```

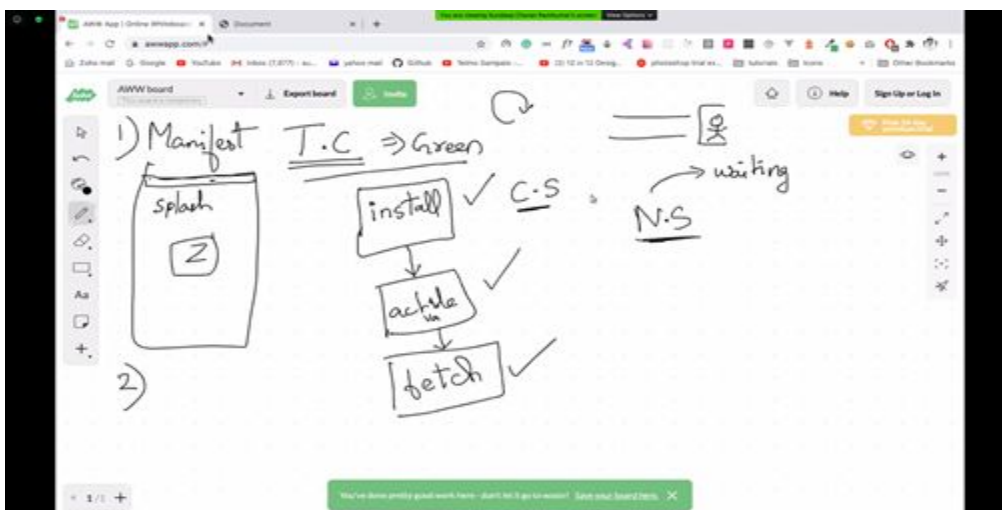
## Editing the Manifest with icons , display and stuff:

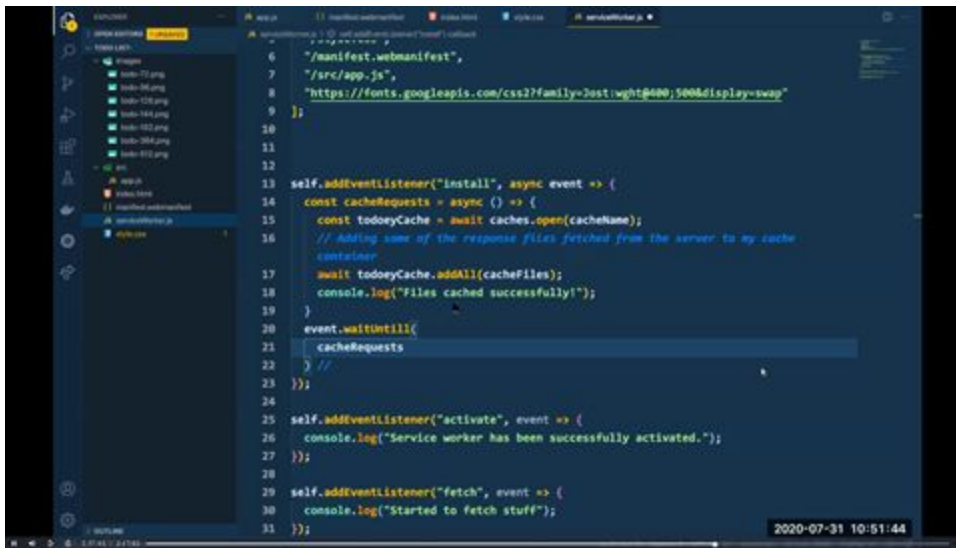


## Generate a Report:



## Service Worker and how things work:

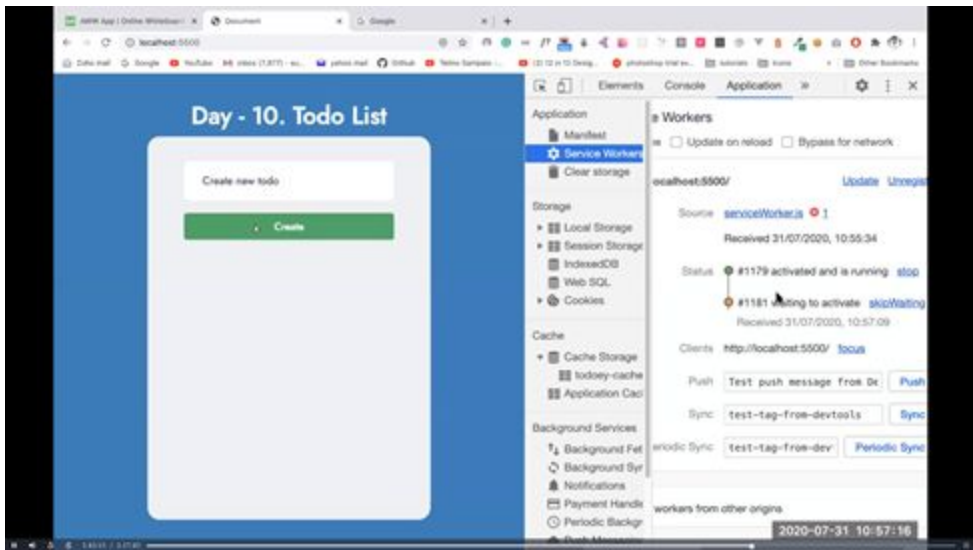




The screenshot shows a VS Code editor with a service worker file named `sw.js`. The code defines a service worker that registers a cache named `today-cache` and adds assets from the `manifest.webmanifest` and `src/app.js` to it. It also includes a `fetch` event listener that logs messages when the service worker is activated and when it starts fetching resources.

```
6  "/manifest.webmanifest",
7  "/src/app.js",
8  "https://fonts.googleapis.com/css?family=Roboto:400,500&display=swap"
9  ];
10
11
12
13 self.addEventListener("install", async event => {
14   const cacheRequests = async () => {
15     const todayCache = await caches.open(cacheName);
16     // Adding some of the response files fetched from the server to my cache
17     // container
18     await todayCache.addAll(cacheFiles);
19     console.log("Files cached successfully!");
20   }
21   event.waitUntil(
22     cacheRequests
23   );
24
25 self.addEventListener("activate", event => {
26   console.log("Service worker has been successfully activated.");
27 });
28
29 self.addEventListener("fetch", event => {
30   console.log("Started to fetch stuff");
31 });
```

Check:



Resources:

Hands down the best resource to learn PWA

<https://www.youtube.com/playlist?list=PLNYkxOF6rcIB2xHBZ7opgc2Mv009X87Hh>