Shravani Pore
D15A_45

MAD PWA Lab 7

Aim:
To write meta data of your Ecommerce PWA in a Web app manifest file to enable “add to homescreen feature”.

Theory:
A regular web app is accessed through a web browser and requires an active internet connection to function. These apps often have limited offline capabilities and may not offer a seamless user experience on mobile devices.

On the other hand, a Progressive Web App (PWA) is a type of web application that leverages modern web technologies to provide a more app-like experience to users. PWAs are designed to work across various platforms and devices, including desktops, tablets, and smartphones. They can be accessed through a web browser like regular web apps but offer additional features and capabilities, such as offline access, push notifications, and the ability to install them on the user's device.

To enable the "add to homescreen" feature for a PWA, you need to include metadata in the Web app manifest file. The Web app manifest is a JSON file that provides information about the PWA, such as its name, icons, colors, start URL, display mode, and more. By defining these properties in the manifest file, you can ensure that the PWA is displayed correctly when users add it to their device's homescreen.

Key features of a PWA that contribute to its app-like experience include:

- Responsive Design: PWAs are built with responsive design principles, ensuring that they look and work well across different screen sizes and devices.
- Offline Access: PWAs can cache assets and data, allowing users to access content even when they are offline or have a poor internet connection.
- Fast Loading: PWAs are optimized for speed and performance, delivering fast loading times and smooth navigation.
- Engagement: PWAs can send push notifications to users, increasing user engagement and retention.
- Installable: Users can add PWAs to their device's homescreen for easy access, similar to native apps.

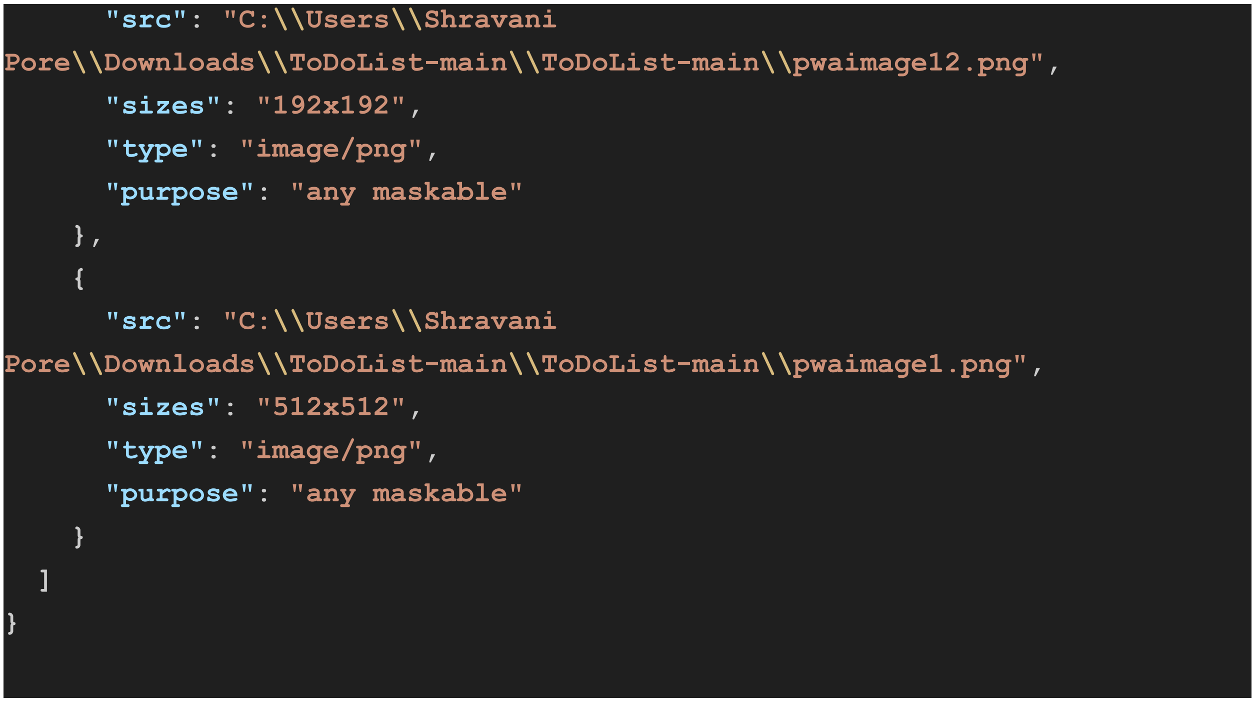
Overall, PWAs combine the best of web and app experiences, offering a lightweight, fast, and engaging way for users to interact with web content on various devices.

Code:
Serviceworker.js

```
var staticCacheName = "pwa";

self.addEventListener("install", function (e) {
  e.waitUntil(
    caches.open(staticCacheName).then(function (cache) {
      return cache.addAll(["/"]);
    })
  );
});

self.addEventListener("fetch", function (event) {
  console.log(event.request.url);

  event.respondWith(
    caches.match(event.request).then(function (response) {
      return response || fetch(event.request);
    })
  );
});
```

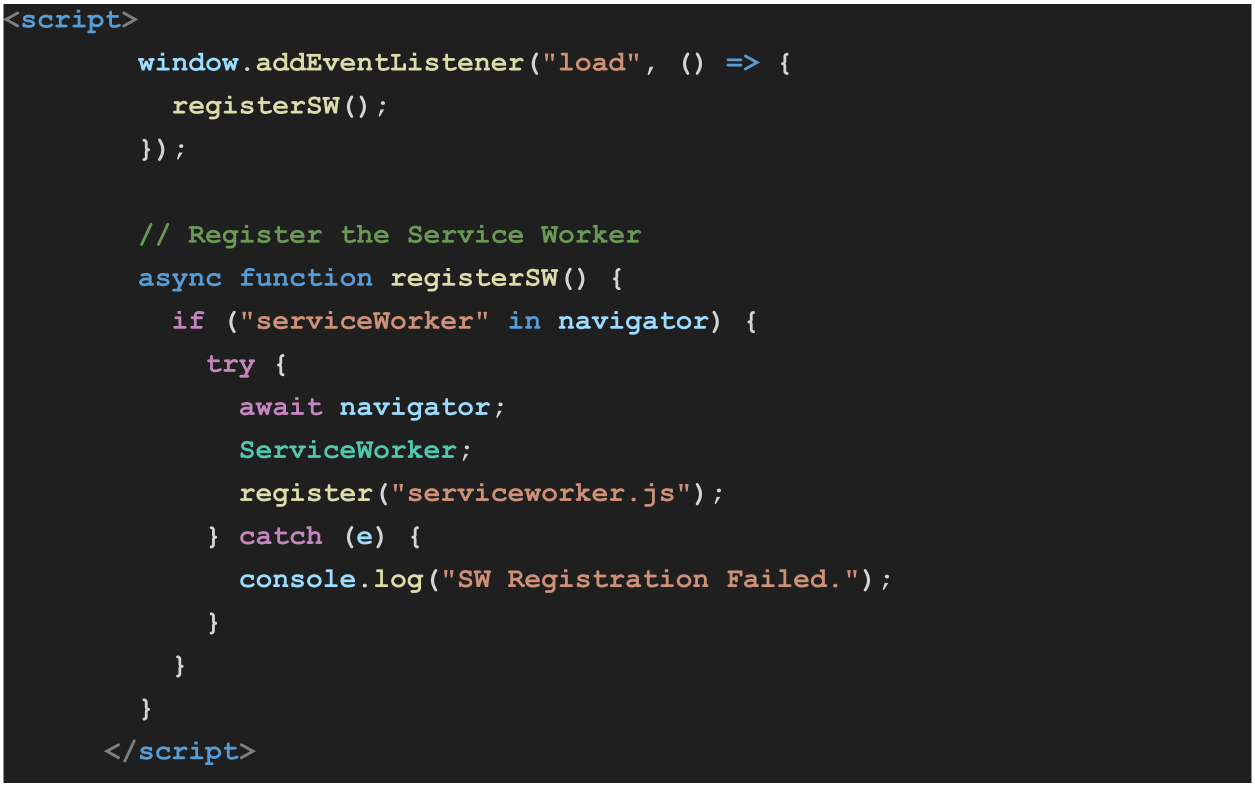
Manifest.json

```
{
  "name": "To-Do List",
  "short_name": "PWA",
  "start_url": "index.html",
  "display": "standalone",
  "background_color": "#5900b3",
  "theme_color": "black",
  "scope": "./",
  "description": "This is a PWA tutorial.",
  "icons": [
    {
```



```
      "src": "C:\\Users\\Shravani
Pore\\Downloads\\ToDoList-main\\ToDoList-main\\pwaimage12.png",
      "sizes": "192x192",
      "type": "image/png",
      "purpose": "any maskable"
    },
    {
      "src": "C:\\Users\\Shravani
Pore\\Downloads\\ToDoList-main\\ToDoList-main\\pwaimage1.png",
      "sizes": "512x512",
      "type": "image/png",
      "purpose": "any maskable"
    }
  ]
}
```

Add following in script of applicatio:

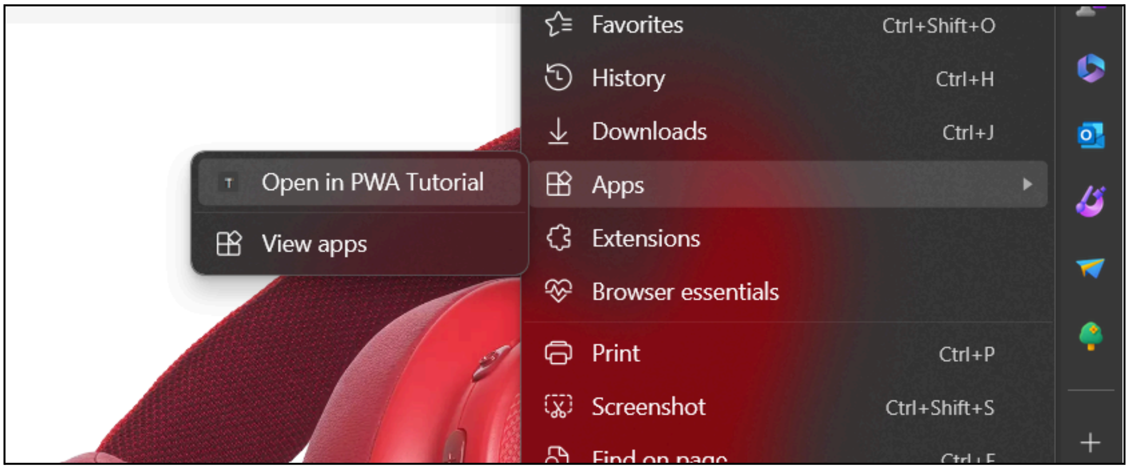


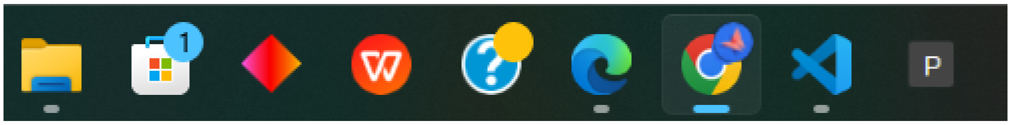
```
<script>
        window.addEventListener("load", () => {
          registerSW();
        });

        // Register the Service Worker
        async function registerSW() {
          if ("serviceWorker" in navigator) {
            try {
              await navigator;
              ServiceWorker;
              register("serviceworker.js");
            } catch (e) {
              console.log("SW Registration Failed.");
            }
          }
        }
      </script>
```

Now run the app on the browser and on the right hand side three dots will appear. Click on those dots and go to more options -> developer tools -> applications

Then go to apps section in the three dots menu and install the app:

Output:





Conclusion:
Successfully added meta data to my Ecommerce PWA in a Web app manifest file to enable “add to homescreen feature”.