

Name: Kshitij Hundre  
Div: D15C  
Roll no: 18

DOP:  
DOS:

## Experiment-7

### 1. Aim

To write metadata of your Ecommerce PWA in a Web App Manifest file to enable the “**Add to Homescreen**” feature.

### 2. Basic Description

In this experiment, we focus on enhancing our **Ecommerce Progressive Web App (PWA)** by integrating a manifest.json file. This file contains metadata such as the app name, icons, theme color, display mode, and start URL. When properly configured and served with a secure HTTPS connection, this setup allows users to **install the web app on their home screen**, providing a native-app-like experience.

This is particularly important for eCommerce apps, where:

- **User engagement and retention** matter.
- A quick-launch option from the homescreen improves accessibility.
- It supports an **offline-capable** shopping experience when combined with service workers.

```
{
  "name": "ShopEase - Your Ecommerce Store",
  "short_name": "ShopEase",
  "start_url": "/index.html",
  "display": "standalone",
  "background_color": "#ffffff",
  "theme_color": "#4CAF50",
  "icons": [
    {
      "src": "icons/icon-192x192.png",
      "sizes": "192x192",
      "type": "image/png"
    },
    {
      "src": "icons/icon-512x512.png",
      "sizes": "512x512",
      "type": "image/png"
    }
  ]
}
```

Index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>ShopEase - E-Commerce</title>

  <!-- Link to manifest -->
  <link rel="manifest" href="manifest.json">

  <!-- Link to external CSS -->
  <link rel="stylesheet" href="style.css">

  <!-- Meta tags for theme -->
  <meta name="theme-color" content="#4CAF50" />
</head>
<body>
  <div class="container">
    <h1>Welcome to <span class="highlight">ShopEase</span>!</h1>
    <p>Your favorite online store as a PWA.</p>
    
    <p class="tip"> Try adding this app to your home screen!</p>
  </div>
  <script src="script.js"></script>
</body>
</html>
```

sw.js

```
// INSTALL: Cache app shell
self.addEventListener('install', (event) => {
  console.log('[Service Worker] Installed');
  event.waitUntil(
    caches.open('v2').then((cache) => {
      return cache.addAll([
        '/',
        '/index.html',
        '/style.css',
        '/script.js',
        '/manifest.json',
        '/icons/icon-192.png'
      ])
    })
  )
})
```

```
    });  
  })  
);  
});
```

```
// ACTIVATE
```

```
self.addEventListener('activate', (event) => {  
  console.log('[Service Worker] Activated');  
  // Optional: Clean old caches  
  event.waitUntil(  
    caches.keys().then((keys) =>  
      Promise.all(  
        keys.map((key) => {  
          if (key !== 'v2') {  
            console.log('[Service Worker] Removing old cache:', key);  
            return caches.delete(key);  
          }  
        })  
      )  
    )  
  );  
});
```

```
// FETCH: Serve from cache or network
```

```
self.addEventListener('fetch', (event) => {  
  console.log('[Service Worker] Fetching:', event.request.url);  
  event.respondWith(  
    caches.match(event.request).then((res) => {  
      return res || fetch(event.request);  
    })  
  );  
});
```

```
// SYNC: Background sync example
```

```
self.addEventListener('sync', (event) => {  
  if (event.tag === 'sync-products') {  
    console.log('[Service Worker] Background Sync - Products');  
    event.waitUntil(syncProductData());  
  }  
});
```

```
function syncProductData() {
```

```

return new Promise((resolve) => {
  // Simulated background sync task
  setTimeout(() => {
    console.log('✅ Product data synced in background!');
    resolve();
  }, 2000);
});
}

// PUSH: Show notification when push received
self.addEventListener('push', (event) => {
  console.log('[Service Worker] Push Received');
  let data = {};

  try {
    data = event.data.json();
  } catch (e) {
    console.warn('Push data is not JSON:', event.data.text());
    data = { title: 'ShopEase Notification', body: event.data.text() };
  }

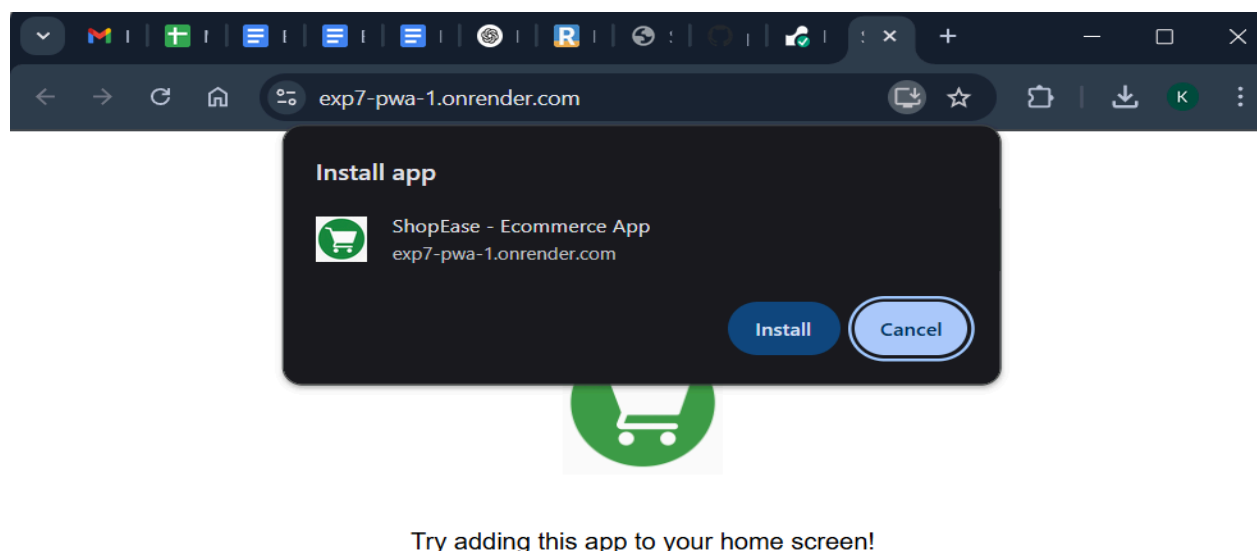
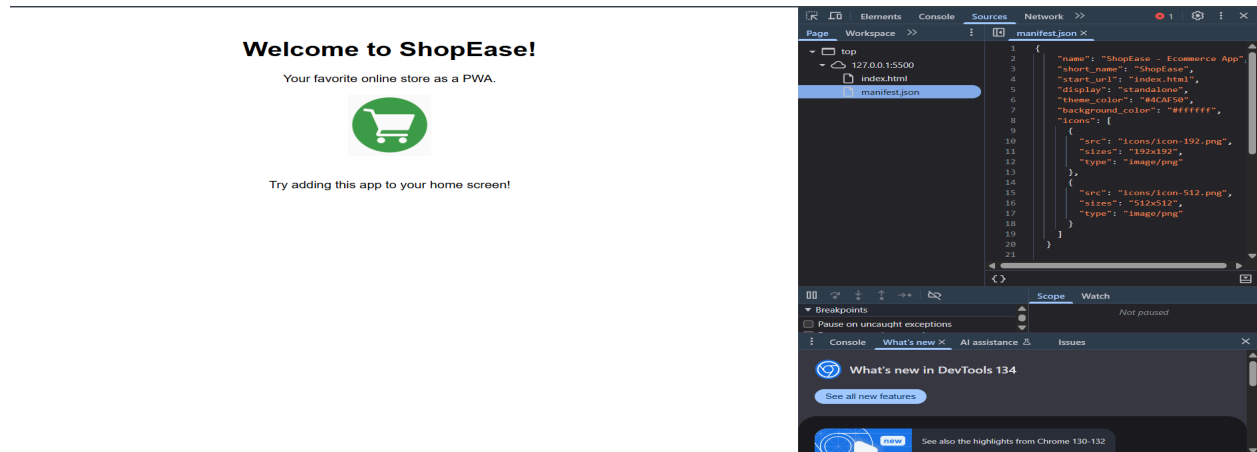
  const title = data.title || 'ShopEase';
  const options = {
    body: data.body || 'You have a new update!',
    icon: '/icons/icon-192.png',
  };

  event.waitUntil(
    self.registration.showNotification(title, options)
  );
});

```

**Github:** <https://github.com/pixelbypixels/EXP7-PWA.git>

### 3. Output:



## 4. Conclusion

By successfully adding a Web App Manifest file to our Ecommerce PWA:

- We enabled “**Add to Home Screen**”, improving user accessibility.
- We enhanced our app’s appearance and functionality to feel more **native**.
- This makes our PWA **installable, reliable, and engaging**, all while using standard web technologies.