Experiment-10

1. Aim

To study and implement deployment of Ecommerce PWA to GitHub Pages(or other equivalent service, have used Render)

2. Basic Description

🔧 Deployment Steps (Summary):

1. **Create a GitHub Repository:**

   ○ Push all project files (index.html, manifest.json, style.css, script.js, sw.js, icons, etc.) to GitHub.

2. **Create a Static Site on Render:**

   ○ Go to https://render.com and sign in.
   ○ Choose "New Static Site".
   ○ Connect your GitHub account and select your repository.

3. **Set Build and Publish Settings:**

   ○ **Build Command**: (leave empty for static site)
   ○ **Publish Directory**: . (dot, for root)
   ○ Click "Create Static Site" and Render will deploy it automatically.

4. **Automatic Deployment:**

   ○ Every time you push to GitHub, Render re-deploys your site with the latest changes.
   ○ You get a custom Render URL like: https://shop-ease-pwa.onrender.com
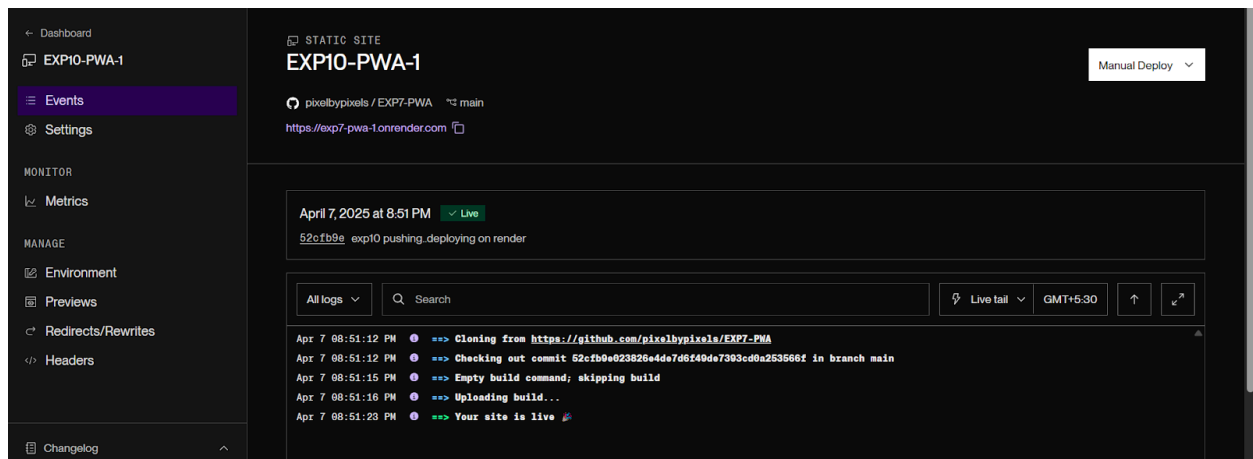
🔍 Things to Ensure:

● All file paths must be **relative** (./style.css, ./script.js, etc.)

- manifest.json and sw.js must be in the **root** directory.
  Icons must be correctly linked in the manifest file and accessible by the browser.
- Make sure the serviceWorker.register() and other scripts are correctly placed and working.

https://exp7-pwa-1.onrender.com

3. Output:

4. Conclusion

We successfully deployed our E-commerce PWA using Render, making it publicly accessible with full PWA functionality like offline support and push notifications. Render simplified the process with GitHub integration and automatic builds. This experiment helped us understand deployment steps, proper file structuring, and the benefits of using modern cloud platforms for hosting web apps.