**Experiment 7:-**

Implementation of Add to Home Screen Feature in Progressive Web Apps (PWA)

**Theory:-**

### What is a Progressive Web App (PWA)?

A Progressive Web App (PWA) is a type of web application that provides a native app-like experience while running in a browser. It can work offline, load fast, and be installed on a user's device just like a mobile app.

### Key Features of a PWA:

1. Responsive Design – Works on different screen sizes (mobile, tablet, desktop).
2. Service Worker – A background script that enables offline functionality and caching.
3. Web App Manifest – A JSON file that defines app metadata like name, icon, and theme color.
4. Secure (HTTPS) – Ensures a secure connection for improved user trust.
5. Add to Home Screen (A2HS) – Allows users to install the app on their home screen.

### Add to Home Screen (A2HS) Functionality:

The A2HS feature enables users to install a PWA on their home screen without using an app store. This enhances accessibility and user engagement. It requires:

* A valid web app manifest (manifest.json)
* A service worker (serviceworker.js)
* HTTPS hosting

**Index.html**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<meta name="theme-color" content="#ffffff">

<title>Simple E-commerce PWA</title>

<link rel="manifest" href="/manifest.json">

<link rel="icon" href="https://via.placeholder.com/192" type="image/png">

<style>

body {

font-family: Arial, sans-serif;

margin: 0;

padding: 20px;

background: #f0f0f0;

}

.product-grid {

display: grid;

grid-template-columns: repeat(auto-fill, minmax(200px, 1fr));

gap: 20px;

max-width: 1200px;

margin: 0 auto;

}

.product-card {

background: white;

padding: 15px;

border-radius: 8px;

box-shadow: 0 2px 4px rgba(0,0,0,0.1);

text-align: center;

}

.product-card img {

max-width: 100%;

height: 150px;

object-fit: cover;

}

button {

background: #007bff;

color: white;

border: none;

padding: 10px 20px;

border-radius: 5px;

cursor: pointer;

}

button:hover {

background: #0056b3;

}

#notification {

position: fixed;

bottom: 20px;

right: 20px;

background: #28a745;

color: white;

padding: 10px 20px;

border-radius: 5px;

display: none;

}

</style>

</head>

<body>

<h1>Simple E-commerce PWA</h1>

<div class="product-grid" id="products"></div>

<div id="notification">Item added to cart!</div>

<script>

// Sample product data

const products = [

{ id: 1, name: "T-Shirt", price: 19.99, image: "https://via.placeholder.com/200" },

{ id: 2, name: "Jeans", price: 39.99, image: "https://via.placeholder.com/200" },

{ id: 3, name: "Shoes", price: 59.99, image: "https://via.placeholder.com/200" }

];

// Render products

function renderProducts() {

const grid = document.getElementById('products');

grid.innerHTML = products.map(product => `

<div class="product-card">

<img src="${product.image}" alt="${product.name}">

<h3>${product.name}</h3>

<p>$${product.price}</p>

<button onclick="addToCart(${product.id})">Add to Cart</button>

</div>

`).join('');

}

// Add to cart simulation

function addToCart(productId) {

const notification = document.getElementById('notification');

notification.style.display = 'block';

setTimeout(() => notification.style.display = 'none', 2000);

}

// Register service worker

if ('serviceWorker' in navigator) {

window.addEventListener('load', () => {

navigator.serviceWorker.register('/sw.js')

.then(reg => {

console.log('Service Worker registered', reg);

})

.catch(err => {

console.log('Service Worker registration failed:', err);

});

});

}

renderProducts();

</script>

</body>

</html>

**manifest.json**

{

"name": "Simple E-commerce PWA",

"short\_name": "E-commerce",

"start\_url": "/index.html",

"display": "standalone",

"background\_color": "#ffffff",

"theme\_color": "#ffffff",

"icons": [

{

"src": "https://via.placeholder.com/192",

"sizes": "192x192",

"type": "image/png"

},

{

"src": "https://via.placeholder.com/512",

"sizes": "512x512",

"type": "image/png"

}

]

}

**Serviceworker.js**

**const CACHE\_NAME = 'ecommerce-pwa-v1';**

**const urlsToCache = [**

**'/',**

**'/index.html',**

**'/manifest.json'**

**];**

**// Install event**

**self.addEventListener('install', event => {**

**event.waitUntil(**

**caches.open(CACHE\_NAME)**

**.then(cache => {**

**return cache.addAll(urlsToCache);**

**})**

**);**

**});**

**// Activate event**

**self.addEventListener('activate', event => {**

**event.waitUntil(**

**caches.keys().then(cacheNames => {**

**return Promise.all(**

**cacheNames.filter(name => name !== CACHE\_NAME)**

**.map(name => caches.delete(name))**

**);**

**})**

**);**

**});**

**// Fetch event**

**self.addEventListener('fetch', event => {**

**event.respondWith(**

**caches.match(event.request)**

**.then(response => {**

**return response || fetch(event.request);**

**})**

**);**

**});**

**// Sync event**

**self.addEventListener('sync', event => {**

**if (event.tag === 'sync-cart') {**

**event.waitUntil(**

**// Add your sync logic here**

**console.log('Background sync triggered')**

**);**

**}**

**});**

**// Push event**

**self.addEventListener('push', event => {**

**const data = event.data.json();**

**const options = {**

**body: data.body,**

**icon: 'https://via.placeholder.com/192'**

**};**

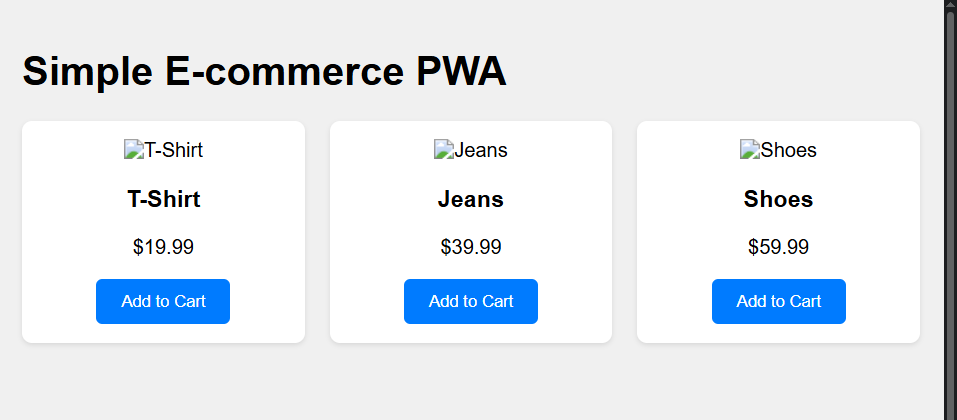
**event.waitUntil(**

**self.registration.showNotification(data.title, options)**

**);**

**});**

**Output:-**

****

**Conclusion:-**

In this experiment, we successfully developed a Progressive Web App (PWA) that includes the Add to Home Screen (A2HS) feature. The application can be installed on a user's device, providing an enhanced user experience similar to a native app. This demonstrates how PWAs can bridge the gap between web and mobile applications while remaining lightweight and accessible.