MAD & PWA Lab

Journal

Experiment No.	Assignment-2
Assignment 2 Questions	1. Define Progressive Web App (PWA) and explain its significance in modern web development.Discuss the key characteristics that differentiate PWAs from traditional mobile apps 2. Define responsive web design and explain its importance in the context of Progressive Web Apps.Compare and contrast responsive, fluid, and adaptive web design approaches. 3. Describe the lifecycle of Service Workers, including registration, installation, and activation phases. 4. Explain the use of IndexedDB in the Service Worker for data storage.
Roll No.	20
Name	Sarthak Harad C
Class	D15B
Subject	MAD & PWA Lab
Lab Outcome	LO4:Understand various PWA frameworks and their requirements LO5: Design and Develop a responsive User Interface by applying PWA Design techniques LO6:Develop and Analyze PWA Features and deploy it over app hosting solutions
Grade:	05

Sarthak Harade
D15B
20

05/05

20/3/25

MPL Assignment 2

Q.1. Define Progressive Web App (PWA) and explain its significance in modern web development. Discuss the key characteristics that differentiate PWAs from traditional mobile apps.

→ A progressive Web App is a type of web application that works like a mobile app but runs in a browser. It can be installed on a device, works offline and provides a fast and smooth user experience.

Significance :-

1) Cross Platform Compatibility - Works on both mobile and desktop with a single codebase.

2) Offline Support - Can function without the internet using cached data.

3) Fast Performance - loads quickly, even on slow networks.

Differences :-

PWA	Traditional Mobile App
(1) Installation is direct from browser.	(1) Download from app store
(2) Works offline with caching.	(2) Usually requires internet.
(3) Fast with service workers.	(3) Faster but needs installation.
(4) Updates are automatic, no app store approval.	(4) Manual updates needed.

Q.:- Define responsive Web design and explain its importance in the context of Progressive Web Apps. Compare and contrast responsive, fluid and adaptive web design approaches

→ Responsive Web Design is a technique that makes web pages adjust automatically to different screen sizes and devices. It ensures a good user experience on mobiles, tablets and desktops without needing separate versions of a website.

Importance :-

- Better User Experience - PWAs work smoothly on any device.
- Faster Load Time - Optimized design improves speed.
- SEO Benefits - Google ranks responsive sites higher
- Cost effective - No need to build multiple versions for different screens.

Comparison :-

Approach	How It Works	Pros	Cons
1) Responsive	Uses flexible grids and CSS media queries to adjust layout.	Works on all devices improves SEO.	Can be complex to design.
2) Fluid	Uses percent based widths instead of fixed pixels so elements resize smoothly.	Works well on different screen sizes, easy to implement.	Less control over layout on large screens
3) Adaptive	Uses fixed layouts that change at specific breakpoints	Optimized for known screen sizes	More effort required to design for each screen size.

Q.3 Describe the lifecycle of Service Workers, including registration, installation and activation phase.

—4 Lifecycle of Service Workers

A Service Worker is a script that runs in the background and helps a web app work offline, load faster and send push notification. Its lifecycle has three main phase:

1. Registration Phase

- The browser registers the Service Worker using JS.

Code Eg:-

```
if ('serviceWorker' in navigator){
    navigator.serviceWorker.register('/sw.js')
    .then(() => console.log('Service Worker Registered'))
    .catch(error => console.log('Registration Failed:', error))
}
```

This tells the browser to install and activate the service worker.

2. Installation Phase

- The Service Worker downloads necessary files (HTML, CSS, JS) and stores them in cache.
- If successful, it moves to activation phase.

Code eg:-

```
self.addEventListener('install', event => {
        event.waitUntil(
            caches.open('app-cache').then(cache => {
            return.cache.addAll(['/', '/index.html', '/style.
                css']);
            })
        );
});
```

3) Activation Phase

. The old Service Worker is replaced with the new one.

Code Eg :-

```
self.addEventListener('activate', event => {
  event.waitUntil(
      caches.keys().then(keys => {
          return Promise.all(keys.map(key => {
          if (key !== 'app-cache'){
              return caches.delete(key);
          }
          }));
        })
      );   });
```

Q.4. Explain the use of indexed DB in the Service Worker For data Storage.

→ IndexedDB is a browser database that stores large amounts of structured data like JSON objects. It helps PWAs work offline by saving and retrieving data efficiently.

(1.) Offline Support - Stores data when offline and syncs it later.

(2.) Efficient Storage - Saves structured data like user settings, cart items or form inputs.

(3.) Faster Access - Retrieves data quickly without needing a network request

(4) Persistent Data - Data remains saved even after the browser is closed.

Use of Service Workers Use IndexedDB :-

(1.) Opening the Database

```
let db;
let request = indexedDB.open ('MyDatabase',1);
 request.onsuccess = function (event){
        db = event.target.result;
     };
```

(2) Creating a Store & Adding Data

```
request.onupgradeneeded = function (event){
   let db = event.target.result;
   let store = db.createObjectStore ('Users', {keyPath: 'id'});
   store.add ({id:1, name: 'John Doe', age:25});
   };
```

(3) Fetching Data in Service Worker

```
let transaction = db.transaction ('Users', 'readonly');
  let store = transaction.objectStore ('Users');
   let getUser = store.get (1);
  getUser = store.get (1);

   getUser.onsuccess = function (){
       console.log(getUser.result);
     };
```