

Name - Teetu Mamtoru

Div - D15B-30

Sub - MPL.

MPL Assignment-2

Ch/05

1

Q1) Define Progressive web app (PWA) & explain its Significance in Modern web development. Discuss the key characteristics that differentiate PWAs from traditional Mobile apps.

→ A Progressive web APP (PWA) is a type of web Application that work, like a Mobile App but runs in a browser. It can be installed on a device, work Offline & provide a fast and smooth user experience.

Significance of PWA in Modern web development:

1] Cross Platform Compatibility - works on both Mobile & desktop with a single codebase.

2] Offline Support - can support without the internet using cached data.

3] Fast Performance - loads quickly, even on slow networks

4] No app store required - ~~at~~ users can install it directly from the browser.

5] Lower development cost - One pwa, can replace separate android and ios apps

key difference between PWA and Traditional Mobile APP-

Features	PWA	Traditio
Installation	Direct from browser	Download from App Store
Internet Required	work offline with caching	usually required Internet.

Performance	Fast with Service workers	Faster but need Installation
updates	Automatic, no App Store Approval	Manual updates needed.
Development Cost	Lower (one codebase for all)	Higher (Separate app for each platform)

PWA's combine the best of web & MobileApp, Making them efficiency & user friendly.

Q2) Define responsive web design, and ~~web~~ 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 size and devices. It ensures a good user experience on mobile tablet and desktops without needs separate version of a website.

Importance of responsive Design in PWA's

- 1] Better user Experience - PWA's work smoothly on any device
- 2] Task load time - Optimized design improves speed.
- 3] SEO Benefits - google ranks responsive sites higher.
- 4] Cost efficient - No need to build multiple version for different screens.

•) Comparison of web design Approaches -

Approach	How it works	Pros	Cons
Responsive	use flexible grids & CSS Media queries to adjust layout	works on all devices improves SEO	can be complex design
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
Adaptive	Uses fixed layouts that changes at specific breakpoints	optimized for known screen sizes	More efficient to design for each screen size.

Q5) Describe the lifecycle of Service workers, including registration installation & activation for phases.

→ lifecycle of Service workers -

A Service workers is a script that runs in the background & helps a web App work offline, load faster & send push notification. Its lifecycle has these Main phases

1] Registration phase -

The registers the service workers using JavaScript

Code Example -

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

2) Installation Phase -

The service worker downloads necessary files (HTML, CSS, JS) & stores them in cache.

If successful, it moves to the Activation phase.

Code Example -

```
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 workers with the new one. Unused cache file from the previous version are detected.

code example -

```

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

```

4) final step: fetch & Sync

Once activated, the ~~servic~~ service worker intercepts network requests serves cached files & syncs data when the internet is available. This lifecycle makes PWA's faster, more reliable & capable for working for Offline.

Q4) Explain the use of indexed DB in the service worker for data storage.

→ Indexed DB is a browser database that stores large amount of structured data like JSON Objects. It helps PWAs work Offline by saving & retrieving data efficiently.

Why use Indexed DB in Services workers?

1) Offline Support - Stores data when ~~at~~ offline & syncs it later -

- 2) efficient storage - Saves structured data like user settings, cart items or form inputs
- 3) Faster Access - Retrieves data quickly without needed a network request.
- 4) Persistent Data - Data remains saved even after the browser is closed

Opening the database -

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

creating a store & adding Data -

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

Fetching data in service worker.

```
let transaction = db.transaction('users', 'readonly');  
let store = transaction.objectStore('users');  
let getUser = store.get(1);  
getUser.onsuccess = function() {  
    console.log(getUser.result);  
};
```