

## MPL Assignment - 2

Q1. 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 (PWA) 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 of PWA in modern web development

- Cross platform compatibility
- Offline support
- Fast performance
- No app store required
- Lower development cost

Key difference between PWA & traditional mobile Apps

### Features

### PWA

### traditional Mobile Apps

#### Installation

Direct from browser

Download from App store

#### Internet required

works offline with  
caching

usually requires  
internet

#### Performance

Fast with service  
workers

Faster but needs  
installations

#### Updates

Automatic no app  
store approval

manual update  
needed.

#### Development cost

Lower

Higher

Q2

Define responsiveness web design and explain its importance in the context of progressive web App! compare and contrast responsiveness fluid and adaptive web design approaches

→ Responsive web design (RWD) is 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 website.

Importance of responsive web design in PWA,

- Better user experience - PWA's 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 of web design approaches

Approaches	How it works	Pros	Cons
------------	--------------	------	------

Responsiveness	uses flexible grids and CSS media queries to adjust layout	works on all devices	can be complex to design
----------------	--	----------------------	--------------------------

Fluid	uses percent based width instead of fixed pixels, so elements resize smoothly	works well on different screen sizes	less control over layout on large screens
-------	---	--------------------------------------	---

Adaptive	uses fixed layout that changes at specific breakpoints	optimized for known screen sizes	more effort required to design for each screen size
----------	--	----------------------------------	---

## key differences

- Responsive adapt dynamically to all screens
- Fluid resizes smoothly, but must be fully prioritized
- Adaptive loads different layouts based on device types.

Q3. Describe the lifecycle of service workers including the registration, installation and activation phase

### → Lifecycle of service workers

A service worker is a script that runs in the background and helps a web app work offline faster and send push notifications.  
Lifecycle has three main phases.

#### 1. Registration Phase

The browser registers the source worker using Javascript code Ex:-

```
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 source worker downloads necessary files (HTML, CSS, JS) and stores them in cache.

If successful it moves to the activation phase.

Code Ex:-

```
self.addEventListener('install', event => {
  event.waitUntil(
```

```
  caches.open(appCache).then(cache => {
    return cache.addAll(['index.html', 'styles.css']);
  })
});
```

This ensures the app loads even without the internet.

### 3. Activation phase

- The old service worker is replaced with the new one
  - Unused cache files from the previous version are deleted
- Final step fetch & sync

Once activated the service worker intercepts network requests, serves cached files and syncs data when the internet is available. This lifecycle makes PWA's faster, more reliable and capable of working offline.

Q4. Explain the use of Indexed DB in the system service workers for data storage.

→ Use of IndexedDB in service workers for data storage. IndexedDB is a browser database that stores large amounts of structured data like JSON objects. It helps PWA's work offline by saving and retrieving data efficiently.

Why use IndexedDB in service workers?

- Offline support - stores data when offline and syncs it later
- Efficient storage - saves structured data like user settings, cart items & form inputs.
- Fast Access - retrieves data quickly without needing a network request.
- Persistent data - Data remains saved even after the browser is closed.

How service workers use IndexedDB?

Opening the database

let db =

```
let request = indexedDB.open('My database');
request.onsuccess = function(event) {
  db = event.target.result;
```

3.

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: 28});

};

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('get user detail');

};

✓