

MPL Assignment 2.

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.

~~Answer:~~ A progressive Web App (PWA) in modern Web Development

1. Cross-Platform compatibility → works on both mobile and desktop with a single codebase.
2. Offline Support - Can function without internet using cached data.
3. Fast performance - Loads quickly, even on slow networks.
4. No App Store Required → Users can install it directly from the browser
5. Lower Development Cost → One PWA can replace separate Android and iOS apps.

Difference between PWA and Traditional Apps.

Feature	PWA	Traditional mobile App
① "Install"	Direct from browser	Download from app store.
② Internet required	works offline with caching	Usually requires internet
③ Performance	Fast with service workers	Faster but needs installation
④ Updates	Automatic, no app store approval	manual updates needed.

⑤ Development cost	lower (one codebase for all)	higher (separate for each platform)
--------------------	------------------------------	-------------------------------------

PWAs combine the best of web and mobile apps, making them efficient and user-friendly.

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.

Definition of Responsive Web Design:

Responsive Web Design (RWD) 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 website.

Importance of Responsive Design in PWAs

- 1. Better User experience - PWAs work smoothly on any device
- 2. Faster load time - Optimized design improves speed.
- 3. SEO Benefits - Google ranks responsive sites higher.
- 4. Cost-Effective - No need to build multiple versions for different screens.

Comparison of Web Design approaches:

Approach	How it works	Pros	Cons
Responsive	Uses flexible grids and CSS media queries to adjust layout	works on all devices improves SEO.	can be complex to 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 change at specific breakpoints	optimized for known screen sizes	more effort required to design for each screen size.

Key Differences

- Responsive adapts dynamically to all screens.
- Fluid resizes smoothly but may not be fully optimized.
- Adaptive loads different layouts based on device type.

Responsive design is best for PWAs because it ensures a seamless experience on all devices.

Q.3. Describe the lifecycle of service workers , including registration , installation and activation phases.

Answer 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 notifications , its lifecycle has three main phases

1. Registration phase

- The browser registers the service worker using JavaScript

Code Example :

```
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 the activation phase

Code Example :

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

Code Example:

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

- The service worker is now fully active and controls network requests.

Final step : Fetch and Sync.

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

Q.4.

Explain the use of indexedDB in 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 and retrieving data efficiently.

Usage

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 IndexedDB by Service Workers.

Opening the Database

let db;

let request = indexedDB.open('my Database', 1);

req.onsuccess = function(event) {

db = event.target.result;

};

Creating a Store and Adding Data

request.onupgradeneeded = function(event) {

let db = event.target.result;

let store = db.createObjectStore('Users', {keyPath : 'id'});

store.add({id: 1, name: 'Shawari', age: 20});

};

FOR EDUCATIONAL USE

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);
};
```

Q.