MPL Assignment 2

Q1] Define Progressive Web App (PWA) and explain its significance in modern web development. Discuss the key charachteristics that differentiate PWAs from traditional mobile apps.

→ Solution:

A progressive Web App (PWA) is a web application that leverages modern web technologies to deliver an app-like experience to users. It combines the best features of web and mobile apps, Offering fast performance, Offline capabilities, and cross platform support without requiring installing from an app store.

SIGNIFICANCE IN MODERN WEB DEVELOPMENT:

a) Improved User Experience:
PWAs provide a smooth, app-like interface with responsive design.

b) Offline Functionality:
Service workers enable caching, allowing PWAs to work without an internet connection.

c) Cross Platform Compatibility:
A single codebase runs on different devices (desktop, mobile, tablet)

d) Cost-Effective Development
Eliminates the need to develop seperate apps for Android and iOS.

e) No App Store Dependency:
Users can install PWAs directly from Browser, reducing installation barriers.

Key charachteristics Differentiating PWA from Traditional Mobile Apps.

Feature	PWA	Traditional Mobile App
• Installation	Added to the home screen via browser	Installed via app store
• Performance	Fast and lightweight	May consume more storage and memory
• Offline Access	Works offline with cached data	Fully functional offline
• Updates	Auto-updated from the web	Requires manual updates via store.
• Platform	Runs on browsers	Platform specific.
• Development cost	Lower	Higher

Q2] 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.

→ Solution:

Responsive Web Design (RWD) is a web development approach that ensures websites automatically adjust their layout, content, and elements based on the user's device screen size, resolution, and orientation. It uses flexible grids, media queries and scalable images to create a seamless experience across desktops, tablets and smartphones

Importance of RWD in PWA:

a) Enhanced User Experience

PWA aims to work smoothly across devices, and RWD ensures consistent UI/UX.

b) Improved Accessibility

Users can access PWA on any screen size without usability issues.

c) Faster Load Time

Optimized layout and images improve performance, which is critical for PWAs.

d) SEO Benefits

Google prioritizes mobile friendly sites, boosting search rankings.

e) Cost Effective

A single responsive design eliminates the need for seperate mobile and desktop versions.

Comparison of Responsive, Fluid and Adaptive Web Design.

Feature	Responsive	Fluid	Adaptive
Approach	Uses CSS media queries to adjust layout based on screen size.	Uses percentage-based layout for flexible scaling	Uses predefined layout for specific screen size.
Flexibility	Highly flexible	Fully Flexible	Less Flexible.
Performance	Optimized for all devices	can be inconsistent on very large screen	May load unnecessary elements for non-matching screen size
Ease of Implementation	Moderate	Simple	Complex
Best For	Modern website	Simple website.	Apps/ website.

Q37 Describe the lifecycle of service workers, including registration, installation, and activation phases.

→ Solution:

Lifecycle of Service Workers:

It has three main phases:

(a) Registration

- The service worker is registered in the JavaScript code using navigator.serviceWorker.register().
- This tells the browser where to find the service worker script and initiates its lifecycle.

```
if ('serviceWorker' in navigator) {
   navigator.serviceWorker.register('/service-worker.js')
   .then(reg => console.log('Service Worker Registered', reg))
   .catch(err => console.log('Service Worker Registration Failed', err));
}
```

(b) Installation

- After registration, the browser tries to install the service worker.
- The install event is triggered where resources can be proloaded.
- If successful, the service worker moves to the activation phase; otherwise, it retries installation.

```
self.addEventListner('install', event => {
    event.waitUntil(
        caches.open('v1').then(cache => {
            return cache.addAll(['/index.html', '/styles.css',
                '/script.js']);
        })
    );
});
```

(c) Activation

- Once installed, the activate event is triggered.
- The old caches or outdated service workers are cleaned up in this phase.
- The service worker takes control of clients and starts handling fetch events

```
self.addEventListner('activate', event => {
  event.waitUntil(
    caches.keys().then(keys => {
      return Promise.all(keys.map(key => {
        if (key !== 'v1') {
          return caches.delete(key);
        }
      }));
    })
  );
});
```

Q4] Explain the use of IndexedDB in the service worker for data storage.

→ Solution:

IndexedDB is a client-side, NOSQL database that allows web applications to store structured data in the browser. When combined with a service worker, it helps enable offline functionality and efficient caching of data.

Use of IndexedDB in a service worker

(a) Persistent Data Storage

Unlike cache API, which mainly stores request/response objects, IndexedDB allows storing structured data like JSON objects, user preferences and application states.

(b) Offline Support
Enables users to access and modify data even when offline. The data can later sync with the server when online.

(c) Efficient caching
Stores large amounts of data that don't need to be re-fetched from the network, reducing load times.

(d) Background sync
Works with Background sync API to sync data once the internet is available.