Name:- Harsh Pramod Padyal DISB 40

MPL Assign 02

05/05

Page No.
Date

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

Ans

A progressive web App (PWA) is a type of web application that works like a mobile app but runs in a browser

Significance of PWA

1. Cross-Platform Compatibility.
2. Offline Support.
3. Fast Performance.
4. No App store Required.
5. Lower Development cost.

Key Differences betwn PWA and Traditional mobile Apps :-

Feature	PWA	Traditional Mobile App
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 installation.
Updates	Automatic, no app store approval	Manual updates require

Feature	PWA	Traditional Mobile App.
Development cost	Lower (one codebase for all)	Higher (seperate apps for each platform)

Q.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.

Ans.

Defination of Responsive web Design:
Responsive web Design (RWD) is a technique that makes web pages adjust automatically to different screen sizes and devices.

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 Benifits - Google ranks responsive sites higher.

4. Cost-Effective - No need to build multiple versions for different screens.

Page No.
Date

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 types.

a.3) Describe the lifecycle of services workers, including registration, installation, and activation phases.

Ans

Lifecycle of service Workers

A service workers 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 brower 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));
}
```

2. Installation Phase :-

- The service workers downloads necessary files (HTML, CSS, JS) and stores them in cache

code Example:

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

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(key => {
      return Promise.all(keys.map(key => {
        if (key !== 'app-cache'){
          return caches.delete(key);
        }
      }));
    })
  );
});
```

Page No.
Date

Final step: Fetch and Sync

Once activated, the Service Worker intercepts network requests, serves cached files, and Syncs data when the internet available.

Q.4) Explain the use of IndexedDB in the Service worker for data storage.

Ans

Use of IndexedDB in Service Worker for Data Storage :-

IndexedDB is a browser database that stores large amounts of structured data like JSON objects.

Why Use IndexedDB in Service Workers? :-

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.

How Service Workers Use IndexedDB?

Opening the Database :-

```
let db;
let request = indexedDB.open('myDatabase', 1);

request.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: '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);
};
```