Nidhi · krishna · Pednekar
D15B / 43

05/05/24

MPL Assignment - 2.

DATE:

Q1. Define Progressive Web App (PWA) and explain its significance in modern web development. Discuss the key characteristics that differentiate PWA's 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 :-

1] Cross Platform Compatibility
2] Offline Support
3] Fast Performance
4] No App Store Required
5] Lower Development Cost

Differences between PWA and traditional Mobile Apps

Feature	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 installation
Updates	Automatic. No app store required	Manual updates needed.

DATE:

PWA's combine the best of web and mobile apps making them efficient and user friendly.

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.

--> Responsive Web Design (RWD) is a technique that makes web pages adjust automatically to different screens size and device. It ensures a good user experience on mobiles, tablets and desktops without needing separate versions of website.

Importance of Responsive Design in PWA :-

1] Better User Experience :- PWA's work smoothly on any device.
2] Faster load Time : Optimized design improves speed.
3] SEO benefits :- Google rank 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	Can be complex to design.

DATE:

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

Differences :-

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

Responsive design is best for PWA's because it ensures a seamless experience on all devices.

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

→ 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 3 main steps/phases :-

i] Registration Phase :- The browser registers the service worker using JavaScript.

DATE:

Code Example :-

```
if ('Service Worker' 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 messages 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 ('appCache').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.

Final step : Fetch and 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 IndexedDB in the service worker for data storage.

→ Use of IndexedDB in the Service worker 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 :-

1] Offline Support :- Stores data when offline and syncs it later.
2] Efficient Storage :- Saves structured data like user setting, 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.

DATE:

How Servie Workers use Indexed DB:-

- Opening the Database :

```
let db;
let request = indexedDB.open('My Database', id);
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: 'Joh Nidhi Pednikar', age: 25})
};
```

- Fetching Data in Servie Worker :-

```
let transaction = db.transaction('users', 'readonly');
let store = transaction.objectStore('users');
let getUser = store.get(1);

getUser.onSuccess = function() {
    console.log(getUser.result);
};
```