Sanika Menebre
D15B - 34

MPL Assignment - 2

05/05

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.

⇒ 1) A Progressive Web App (PWA) is a type of web application that combines the best features of web and mobile apps, offering a fast, reliable, and engaging user experience.

2) PWAs use modern web capabilities to deliver and app-like experience while being accessible through a web browser.

Significance in Modern Web Development

- Cross-platform compatibility : PWAs work on any device with a modern web browser, reducing the need for separate development for IOS and Android.
- Offline functionality : Thanks to Service Workers, PWAs can work even without an internet connection.
- App-like experience : PWAs provide a full-screen experience, push-notifications and home screen installation, making them

similar to native apps.

- SEO-friendly: Unlike traditional mobile apps, PWAs are indexable by search engines, improving discoverability.

	PWA	Traditional Mobile App
1)	No app store needed; installed via browser	Installed via App store or Play store
2)	Updates are automatic, no user intervention needed.	Updates are required to be manual from the store.
3)	Fast loading with caching	Faster but requires most storage.
4)	Offline access with service workers	Offline access only if designed so.
5)	Development cost is lower due to single codebase	Development cost is higher due to separate development for platforms.

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.

⇒ 1) Responsive Web Design (RWD) ensures that web applications automatically adjust their layout, images, and content based on the screen size and resolution of the user's device.

2) This approach is crucial for PWAs, as they must provide a seamless experience across various devices, from desktops to smartphones.

Importance of Responsive Web Design in PWAs

- ensures consistency in user experience across different screen sizes.
- enhances usability by adapting navigation and content layout.
- Reduces development effort by eliminating the need for multiple versions of the same app.

1.] Responsive:

1) Uses flexible grids, media queries, and scalable images to adjust layout based on screen size.

2) It provides a seamless experience across all devices but can be complex to implement.

2.] Fluid :

1) Uses percentage - based layouts, making elements scale relative to the screen size
2) It has smooth resizing without fixed breakpoints but it may lead to poor readability if not carefully designed.

3.] Adaptive :

1) Uses fixed layouts for specific screen sizes, loading a different design depending on the device.
2) It has optimized experience for each device type but it requires multiple versions, increasing the maintenance effort.

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

⇒ 1) A Service Worker is a script that runs in the background, enabling features like caching and push notifications.
2) Its lifecycle consists of three main phases.

1.] Registration :

The Service Worker is registered using Javascript in the main web page.
The browser checks if a new Service Worker

4) It handles large and complex data efficiently.
5) It supports transactions and indexing for better performance.

```
const dbPromise = idb.open('my-database', 1, upgradeDB => {
  if (!upgradeDB.objectStoreNames.contains('items')) {
    upgradeDB.createObjectStore('items', { keyPath: 'id' });
  }
});

// Storing data
function saveData(data) {
  dbPromise.then(db => {
    const tx = db.transaction('items', 'readwrite');
    const store = tx.objectStore('items');
    store.put(data);
    return tx.complete;
  });
}

// Retrieving data
function getData(id) {
  return dbPromise.then(db => {
    const tx = db.transaction('items', 'readonly');
    const store = tx.objectStore('items');
  });
}
```

This approach allows PWAs to maintain functionality even when offline by caching essential data and retrieving it when needed.

3.] Activation:

The old service Worker is replaced with the new one. cleanup of outdated caches is performed.

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

After activation, the Service Worker starts controlling the app and can handle fetch requests.

Q4. Explain the use of IndexedDB in the Service Worker for data storage.

⇒ 1) IndexedDB is a low-level, NOSQL database that allows web applications to store large amounts of structured data.

2) Unlike localStorage, it provides effecient read/write operations and works asynchronously.

3) It enables offline data storage and retrieval.

file is available.

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

2.] Installation:

The Service Worker is downloaded and installed. This phase is used for caching essential assets.

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