

(9/5) ✓

Name: SHREYAS NAIK

Roll no: 38

Div: DISB

MPL Assignment (2)

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

→ A progressive web app (PWA) is a type of web application that leverages modern web technologies to provide an app-like experience on web. PWAs combine the best features of traditional web apps and native mobile applications, offering improved performance, offline capabilities, enhanced user engagement.

Significance in Modern Web Development

- ~~cross-platform compatibility~~ **Cross-platform compatibility**: Service workers allow caching enabling PWAs to function offline or in low-network conditions.
- **Improved Performance**: PWAs utilize caching strategies to load faster than traditional web pages.
- **SEO friendly**: Unlike native apps, PWAs are indexable by search engines.

Key Characteristics of PWA vs Traditional Mobile Apps.

Feature	PWA	Traditional Mobile App
Installation	No app store required add to home screen	Installed via App Store Store
Offline Support	Yes, using service workers	Yes via local storage and background sync
Performance	Fast due to caching	Optimised but may need more resources
Updates	Automatic updates	Manual update via app store

Q.2) Define responsive web design and explain its importance in the context of PWAs. Compare and contrast responsive fluid, and adaptive web design approaches.

Soln Responsive Web Design is a design approach that ensures a website's layout adjusts dynamically to different screen sizes and resolutions.

Importance in PWAs

- PWAs need to function seamlessly across multiple devices (mobile, tablet, desktop)
- Ensure a consistent user experience regardless of device

sizes and resolution

Enhances accessibility and usability

Improves SEO rankings as mobile friendly websites are favoured by search engines.

Comparison of Responsive, Fluid and Adaptive Web Design

Design Type	Description	Pros
Responsive	Uses flexible grids & media queries to adjust layout dynamically	Flexible, consistent user experience
Fluid	Uses percentage-based widths to scale elements proportionally	Smooth scalability
Adaptive	Uses predefined layouts for specific screen sizes	Optimised for different devices

Describe the lifecycle of Service Works, including registration, installation and activation phases.

Service Works are background scripts that enable offline support, caching and background sync in PWAs. Their lifecycle consists of three main phases

1) Registration:

- The service worker is registered in the browser via navigator.serviceWorker.register()
- This process occurs in the main JavaScript thread of application

2) Installation:

- Run when a new service worker is detected
- Can cache assets for offline use
- If the installation fails, the service worker is discarded.

3) Activation:

- After installation, it moves to the active phase
- Can clear old caches and manage background sync.
- Becomes fully functional after activation

if ('serviceWorker' in navigator) {

navigator.serviceWorker.register('sw.js')

- then (reg ⇒ console.log('service worker registered'))
- catch (err ⇒ console.log('service worker registration failed', err))

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

IndexedDB is client-side NoSQL database used for storing structured data in browsers. It allows PWAs to store

large amounts of data persistently

How IndexedDB works with service workers
stores data locally, enabling offline access
supports complex querying unlike local storage
Asynchronous operations prevent UI blocking
Data persistence even when the browser is closed

eg.

```
let db;
```

```
let request = indexedDB.open("PWA-DB", 1);
```

```
request.onsuccess = function (event) {
```

```
  db = event.target.result;
```

```
  let store = db.createObjectStore("users", {keyPath: "id"});
```

```
};
```

```
request.onsuccess = function (event) {
```

```
  db = event.target.result;
```

```
};
```

```
function addUser(user) {
```

```
  let transaction = db.transaction(["users"], "readwrite");
```

```
  let store = transaction.objectStore("users");
```

```
  store.add(user);
```

```
}
```

```
self.addEventListener("fetch", event => {
```

```
  event.respondWith(
```

```
    caches.match(event.request).then(response => {
```

```
      return response || fetch(event.request);
```

```
    });
```

```
});
```

Benefits of using Indexed DB on PWAs

- enables efficient offline storage
 - Improves app performance
 - Allows background syncing when the connection is restored.
- 