Shrusti Chintawar
DISB
09

MPL assignment 2

04/03

Page No.
Date

Q1) Define progressive web App (PWA) & explain its significance in modern web development. Discuss the key characteristics that differentiate PWAs from traditional mobile apps.

Soln:- A progressive Web app (PWA) is a type of web application that combines the best features of both traditional web pages & native mobile apps.
It leverages modern web technologies such as service workers, manifests, & responsive design, to deliver a fast reliable & engaging user experience across different devices

Significance of PWAs in modern web development.

PWAs have gained popularity in modern web development due to their ability to provide app like experiences without requiring installation from an app store.
Their significance includes

- cross-platform compatibility - works seamlessly across desktops tablets & mobile devices
- Improved performance: Caching & efficient loading reduce page load times
- offline functionality - service workers enable offline access, enhancing user engagement

difference in PWA & Traditional Apps

features	PWA	Traditional app
Installation	direct from browser	download from app store
Internet req	work offline with caching	usually requires internet

performance	fast with service workers	fast with need installation
updates	Automatic	Manual
development cost	lower	higher

Q2) define responsive web design and explain its importance in the content of progressive web Apps compare & contrast responsive, fluid, & adaptive web design approaches

Soln:- definition of responsive web design:
Responsive web design (RWD) is a technique that makes web pages adjust automatically to different screen size & devices it ensures good user experience on mobile, tablet) & desktops without needing separate

Importance of Responsive design in PWAs:

i) better user experience
ii) faster load time
iii) SEO benefits
iv cost effective

Comparisons:

Approaches	How it works	pros	cons
Responsive	css media queries & flexible grids	works on all devices	can be complex

Key difference:

1) Responsive adopts dynamically to all screens
2) fluid resizes smoothly but may not be fully optimized
3) adaptive loads different layout based on device type

Q3) describes the lifecycle of service workers, including registration, initialization & activation phases

Soln: lifecycle of service workers: A service worker is a script that runs in the background and helps a web app work offline, load faster & send push notifications its lifecycle has 3 main phase

1) Registration phase: The browser registers the service worker using Javascript

eg

```
if ('service worker' in navigator) {
    navigator.service worker.register
    .then() ⟹ console.log("service registered")
    .catch (error ⟹ console.log("register failed :", error));
```

2) Installation phase

i) The service worker downloads necessary files (HTML, CSS, JS) ~~and~~

ii) If successful, it moves to the activation phase

code eg

self.addEventListener('install' event ⟹ {

```
cache.open('app-cache')
return cache.addAll(['/', 'index.html', 'style.
});
);
});
```

iii) Activation phase
The old services worker is replaced with new one request-serves cached files & types data

Q4) Explain the use of indexed DB in the services worker for data storage

Soln: use of Index DB in service worker for data storage: Indexed DB is a browser database that stores large amount of structured data like JSON objects It helps pwfs work offline by saving & retrieving data efficiency

why use IndexedDB in service workers?

i) offline support - store data when offline & sync it later
ii) efficient storage - saves structure data like user setting cart items or form inputs
iii) faster access - retrieves data quickly without needing a network request
iv) 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('My database', 1);

    request.onsuccess = function (event) {

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

creating a store & 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);
    };
```