

PWA Assignment 2

Q1) What is Progressive Web App (PWA) and explain its significance in modern web development. Discuss the key characteristics that differentiate PWAs from traditional mobile apps.

- Ans1)
- A progressive web app (PWA) is a type of web application that uses modern web technologies to provide users with a native app-like experience directly through their web browser. PWAs are designed to be reliable, fast and engaging, offering features such as offline access, push notifications and the ability to install the app.
 - Significance of PWAs in modern web development lies in their ability to bridge the gap between web and native mobile applications.
 - Differentiating factors:
 - 1) Cross platform compatibility: PWAs are built using web technologies like HTML, CSS, JS which makes them compatible with platforms like desktops, smartphones, tablets.
 - 2) Responsiveness: PWAs are designed to adapt seamlessly to different screen sizes.
 - 3) Offline functionality: PWAs can cache resources and content, enabling users to access the app even when they're offline or have poor internet connection.

- 4) Progressive enhancement: They can deliver basic functionality to all users while providing enhanced features to modern devices.
- 5) Installability: PWAs can be installed on a user's device directly on the browser.
- 6) Discoverability: Are discoverable through search engines, social media & links.
- 7) Push notifications: Can send push notifications to users, increasing engagement and enabling real-time communication.

Q2) Define responsive web design and explain its importance in the context of PWA. Compare & contrast responsive, fluid and adaptive web design approaches.

- Ans: 2)
- Responsive web design is an approach to web dev aimed at creating websites that provide an optimal viewing and interaction experience across a wide range of devices & screen sizes.
 - In context of PWA, responsive design is crucial for ensuring that the app functions and looks good on various devices.
- Responsive Web Design
- Uses flexible layouts and fluid grids to adapt the layout and content of a website to different screen sizes.
 - Achieves responsiveness through CSS media queries that adjust styles based on screen

width, height and orientation.

- Fluid web design

- Similar to responsive design, fluid design uses flexible layouts and fluid grids to adapt to different screen sizes.
- Focuses more on fluidity and flexibility.

- Adaptive Web design

- Involves creating multiple layouts or designs tailored to specific screen sizes or device categories.
- User-server side techniques to detect the user's device and deliver the appropriate layout or design.

(Q3) Describe the lifecycle of Service Workers, including registrations, installation and activation phases.

Ans3 1. Registration

- Occurs in the main script of web application, the main.js file.
- Script is fetched from the server & registered with the browser using navigator.serviceWorker.register().

2. Installation

- The browser fires the 'install' event, allowing the Service Worker to cache static assets.

and other resources needed to run the web application offline.

- The service worker can use the `event.waitUntil()` method to extend the installation process.

3. Activation

- Once the Service Worker is successfully installed, browser fires the 'activate' event.
- The activate event is also an opportunity to take control of client pages & intercept network request using event listeners such as 'fetch' and 'message'.

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

Ans4) • IndexedDB is a low-level API provided by modern web browsers for client-side storage of large amounts of structured data, including Service Workers to store & retrieve data in a structured manner.

- Persistent Storage: IndexedDB provides persistent storage, meaning the data stored in the database persists even after the web page or the service worker is closed or refreshed.

2. Asynchronous API : IndexedDB operations are asynchronous which means they don't block the main thread of the web app.
 3. Structured Data Storage : IndexedDB stores data in a structured manner, similar to a NoSQL database.
 4. Large Data Handling : IndexedDB is capable of handling large amounts of data efficiently, making it suitable for applications that require storage of substantial datasets.
 5. Transaction-based : IndexedDB operations are performed within transactions, ensuring data integrity and consistency.
- In context of Service Worker, IndexedDB can be used for various purposes.
 - 1) Caching : Storing resources such as HTML, CSS, JS files, images and other assets for offline access.
 - 2) Data Persistence : Storing application data, such as user preferences, settings or application state to provide a seamless user experience across sessions.

3) Background Synchronization : Storing data received from application background synchronization tasks such as periodic updates or push notifications without interrupting the user's interaction with the web application.