

MAD Assignment - II

DATE:

Define a PWA and explain its significance in modern web app development. ~~Discuss~~ Discuss key characteristics that differentiate PWA's from mobile app

A progressive ~~web~~ web app is a web dev capabilities to deliver an app-like experience to users. It represents a fundamental shift in how we approach web development by combining the best of web and native applications. Key characteristics of PWA is Traditional mobile app -

Installation & distribution:

PWA's: can be access easily through browser ~~and~~ Native. Require downloads from App stores

Updates:

PWA's: Update automatically when new content is available.

Traditional App: Require manual updating through app stores

Cross-platform compatibility:

PWA's: Work across platforms through web browsers. on platform specific OS

Native: Work ~~cross~~ across platforms through

PWA's: Work across platforms through web browsers.

- (b) Key features of PWAs:
- (i) Progressive Enhancement: Works for all browsers regardless of browser choice.
 - (ii) Responsive: Adapts to various screen sizes and orientations.
 - (iii) Connectivity Independent: Functions offline with ~~bandwidth~~ network.
 - (iv) App-like interface
 - (v) Secure: Served via HTTPS.

(2) Responsive Web Design In PWA's:

Ans: Responsive Web Design (RWD) is an approach that makes web pages render well across different devices & window / screen sizes.

Importance

- Ensures consistent user experience across all devices.
- Improves accessibility and usability.
- Reduces maintenance overhead.
- Better ~~marketing~~ SEO performance.
- Essential for PWA ~~and~~ adoption.

Comparison of design approaches

• a) Responsive design:

- Container {
 - width: 100%
 - max-width: 1200px

@media ~~max~~ (max-width: 768px) {
 • container {
 padding: 0 20px; }
 • uses fluid grids
 • flexible images
 • media queries
 • single layout that adapts.

Fluid Design:
 • container {
 width: 90%;
 margin: 0 auto; }
 • column {
 width: 33.33%;
 float: left; }

Uses percentage-based width.
 Continuously adapts
 No breakpoints
 Smoother transitions

Adaptive designs:
 Uses distinct layouts
 Fixed width at breakpoints
 Device specific layouts
 Less flexible, but more controlled.

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

- Adding / Updating records.
- Querying data using indexes.
- Deleting records
- Using transactions for data consistency

Common Use

Common use cases

- Offline data storage
- Caching application data
- Storing user-generated content
- Client-side database for webapps.
- Temporary storage for pending uploads

