DATE:

Aim - To study the requirement for progressive web application for E-commerce using the concept of service workers, web app manifest and frame work tools.

Theory -

- What is PWA ?

→ PWA stands for "progressive web App" and it refers to a type of web application that offers a range of benefits over traditional websites. Here are three key points to know about PWAs:

1. Improved User Experience - PWAs provide a seamless app-like experience for users, with features such as offline access, push notifications and the ability to be installed on a devices home screen.
2. Faster load Times - PWAs use service workers to cache web site assets and content, allowing them to load quickly.
3. PWAs offer many benefits over traditional web and native apps, including increased engagement, better performance and lower development costs.

- Importance of PWA

→ 1. Improved User Experience - PWAs provide a fast and seamless experience to users, with features such as

DATE:

offline access, push notifications and app-like interface, leading to higher user engagement and retention.

2. Increased Reach - PWAs can be accessed through any modern web browser, making them accessible to wider audience without the need for installation through an app store.

3. Lower costs and Development Time - Developing a PWA is generally quicker and more cost-effective than developing a native app for different platforms, as it requires a single codebase that can be updated easily, leading to lower development and maintenance costs.

4. Better Performance - PWAs are designed to load quickly, even on slow connections, and use less data compared to traditional web apps. This results in a better user experience and reduced bounce rates, leading to increased conversions and revenue for businesses.

- Technical components of PWA: Web App manifest, service worker, service worker lifecycle.

DATE:

→ The technical components of PWA include:

1. Web App manifest - A JSON file that describes the PWA, including it's name, icon, color scheme and other information that the browser uses to install and display the PWA. The web App manifest also allows users to add the PWA to their home screen and launch it like a native app.

2. Service workers - A Javascript file that acts as a proxy between the PWA and the network allowing it to work offline and provide other advanced features such as push notifications background syncing and caching. The service worker runs in the background, separate from the PWA, and can intercept network requests and respond with cached content.

3. Service worker Lifecycle - The service worker has a lifecycle consisting of four stages: registration, installation, activation and update. During registration, the service worker is downloaded and installed in the browser. During installation, the service worker caches the necessary resources for offline access. During activation, the service worker takes over control of network requests, allowing the PWA to work offline.

DATE:

• Framework tools: webpack :- webpack-PWA-framework, Lighthouse :- Lighthouse PWA-framework.

→ webpack and Lighthouse are two popular framework tools used in the development of PWA.

1. webpack - webpack is a module bundler that can be used to bundle Javascript, CSS, images, and other resources used in PWAs. It can also be used with plugins such as the webpack-pwa-manifest and webpack-pwa-framework to optimize pwas for performance and offline access. The webpack-pwa-plugin for example, allows developers to easily configure the service workers and precache assets, making it easy to develop PWA.

2. Lighthouse - Lighthouse is an open-source tool developed by google that can be used to audit and test the performance, accessibility and best practices of PWAs. It can be used in the browser or as a command-line tool to generate detailed reports on issues that affect the user experience of PWAs. The Light house pwa-framework is a pre-configured version of Lighthouse that provides a set of checks specifically for PWAs, helping developers identify and fix issues.