

Experiment 7

Aim: To write meta data of your PWA in a Web app manifest file to enable “add to homescreen feature”

Theory:

Progressive Web Apps (PWAs)

Progressive Web Apps are web applications designed to deliver an app-like experience using modern web technologies. They offer improved accessibility, performance, and can be installed directly onto a user's device without using an app store.

Key Features of PWAs

- **Installable on Home Screen**

PWAs can be added to a device's home screen through a web browser using a Web App Manifest (manifest.json). This feature is supported across various platforms including desktop browsers (Chrome, Edge) and mobile devices (Android, iOS).

- **Offline Support via Service Workers**

Service Workers enable PWAs to cache assets such as HTML, CSS, JS, and images, making them functional even without internet access. They also support background synchronization for delayed data updates.

- **Optimized Performance**

By leveraging pre-caching strategies, PWAs load quickly on repeat visits, even under slow network conditions. They utilize the PRPL pattern (Push, Render, Pre-cache, Lazy-load) to enhance speed and responsiveness.

- **App-Like Behavior**

With support for fullscreen or standalone modes, splash screens, and push notifications, PWAs closely mimic native app experiences, thanks to configurations provided in the Web App Manifest.

- **Cross-Platform Usability**

PWAs run seamlessly on all device types—desktop, mobile, or tablet—using a single codebase, eliminating the need for separate development for Android or iOS.

- **Secure by Default (HTTPS)**

PWAs must be served over HTTPS to ensure secure connections, protect against data breaches, and maintain user trust through integrity and encryption.

Understanding the Web App Manifest

The Web App Manifest (manifest.json) is a crucial component of a PWA. It provides the browser with essential metadata, allowing the app to be installable and to control its appearance and behavior when launched.

Primary Uses of the Manifest:

- Triggers the “Add to Home Screen” prompt
- Defines launch behavior (e.g., fullscreen or standalone)
- Sets icons, colors, and splash screen appearance

Important Manifest Properties

Property	Purpose	Example Value
name	Full application name	"My PWA App"
short_name	Display name on home screen	"PWA"
start_url	Launch entry point	"/index.html"
display	UI mode on launch	"standalone"
icons	App icon list (various sizes/types)	[{"src": "icon-192x192.png", ...}]
background_color	Splash screen background color	"#ffffff"
theme_color	App theme (toolbar) color	"#4285f4"
scope	Navigation scope	"/app/"

Browser Requirements for Installation

For a browser to show the “Add to Home Screen” prompt, several criteria must be satisfied:

- Secure HTTPS Connection – Mandatory for all PWAs
- Linked Manifest File – Included in HTML via `<link rel="manifest" href="/manifest.json">`
- Registered Service Worker – Ensures offline support and caching
- User Engagement – Browsers like Chrome require users to visit the site multiple times (e.g., twice in 5 minutes)

PWA Installation Process

1. Browser verifies the existence of a valid manifest and service worker.
2. If conditions are met, Chrome or Edge displays an install prompt (on both desktop and mobile).
3. On iOS (Safari), installation is done manually through the “Add to Home Screen” option in the share menu.
4. Once installed, the app launches in standalone mode without browser UI.

Code:

```
{
  "name": "taskverse",
  "short_name": "taskverse",
  "start_url": ".",
  "display": "standalone",
  "background_color": "#0175C2",
  "theme_color": "#0175C2",
  "description": "A new Flutter project.",
  "orientation": "portrait-primary",
  "prefer_related_applications": false,
  "icons": [
    {
      "src": "icons/Icon-192.png",
      "sizes": "192x192",
      "type": "image/png"
    },
    {
      "src": "icons/Icon-512.png",
      "sizes": "512x512",
      "type": "image/png"
    },
    {
      "src": "icons/Icon-maskable-192.png",
      "sizes": "192x192",
      "type": "image/png",
      "purpose": "maskable"
    },
    {
      "src": "icons/Icon-maskable-512.png",
      "sizes": "512x512",
      "type": "image/png",
      "purpose": "maskable"
    }
  ]
}
```

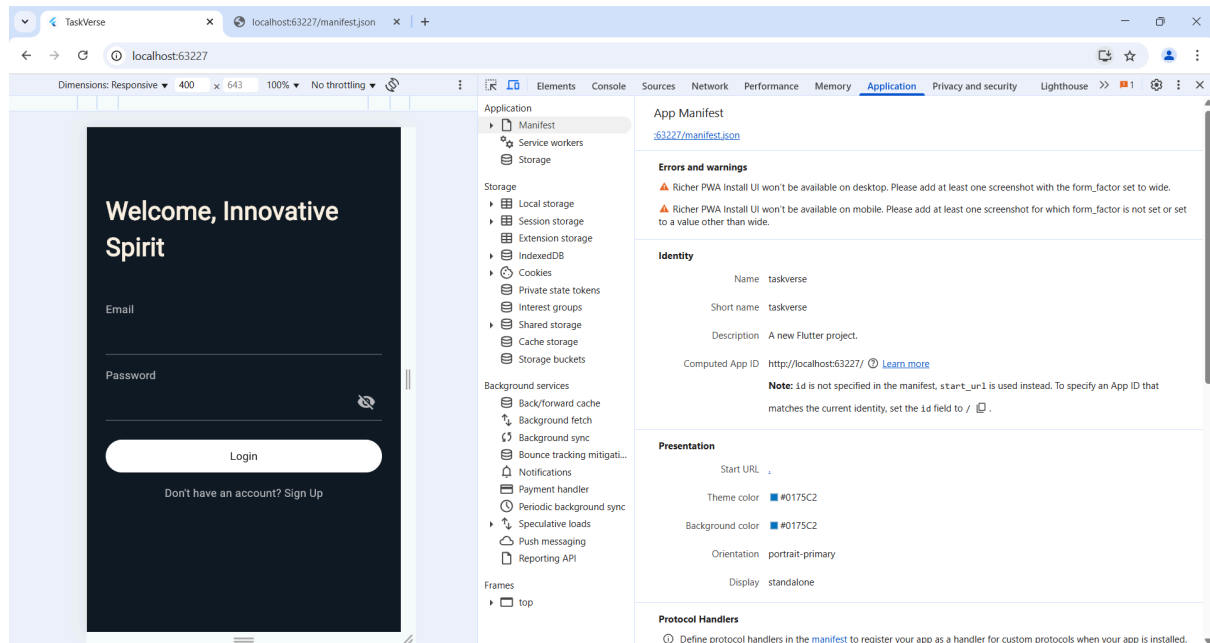
Add this in index.html inside the head tag:

```
<link rel="manifest" href="manifest.json">
```

Run the website on the live server.

Inspect the webpage, go to application

Inside applications, verify if manifest.json has been loaded successfully or not.



GitHub Link: https://github.com/BKCODE2003/Flutter_PWA

Conclusion:

This experiment illustrated how to activate the "Add to Home Screen" functionality in a Progressive Web App (PWA) by correctly setting up the Web App Manifest ('manifest.json'). Key attributes such as the app's name, icons, display mode, and theme color were defined within the manifest. By linking it in the HTML file and confirming its successful loading through browser developer tools, we ensured that the app met the necessary installation requirements. This allows users to install the PWA directly onto their devices, providing a native-like experience that boosts accessibility, user engagement, and offline capability—without relying on traditional app stores.