Name: YOGESHCHANDRA PARANJPE Div: D15B RollNo.52

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

Theory:

PWA stands for Progressive Web App. It's a type of application software delivered through the web, built using common web technologies such as HTML, CSS, and JavaScript. PWAs are intended to work on any platform that uses a standards-compliant browser. They have several characteristics that distinguish them from traditional web applications:

Features:

1. Progressive Enhancement : PWAs are designed to work for every user, regardless of the browser they are using, by employing progressive enhancement principles.

2. Responsive: They are responsive and adapt to any device size, whether it's a desktop, tablet, or mobile phone.

3. Connectivity Independent: PWAs can work even when the device is offline or has a poor internet connection, thanks to service workers and caching strategies.

4. App-like Interface: PWAs offer an app-like experience with features such as push notifications and full-screen mode.

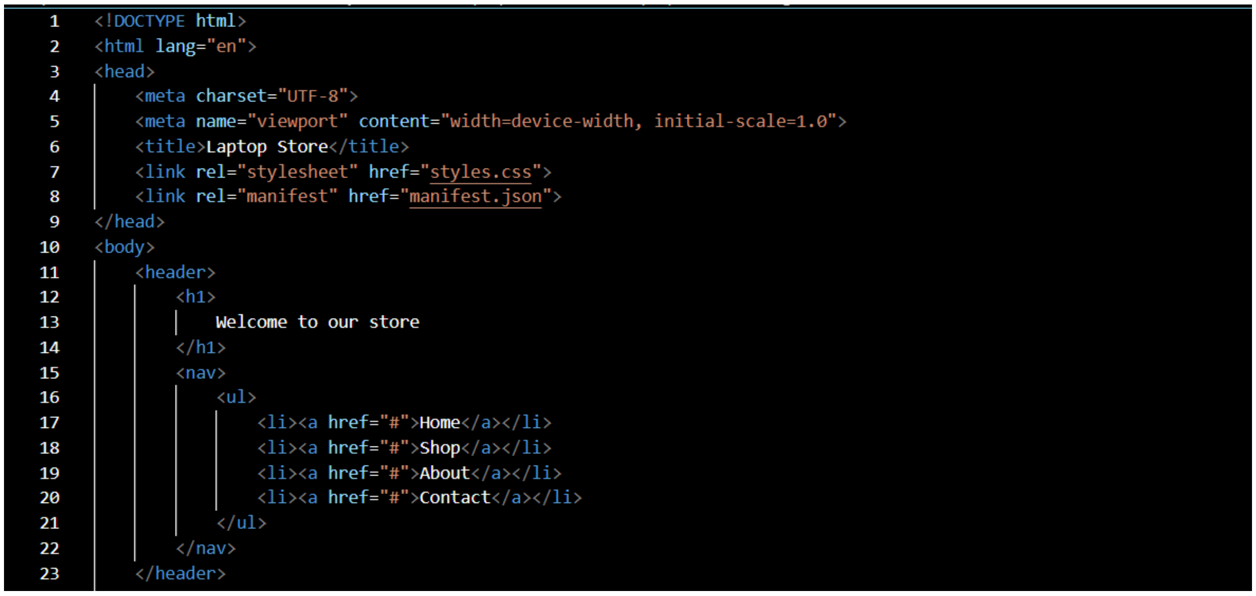
5. Discoverable: PWAs are discoverable by search engines and can be indexed like traditional websites.

PWA vs Traditional web app:
Progressive Web Apps (PWAs) differ from traditional native apps in their platform independence, installation process, distribution model, and update mechanism. Unlike native apps, PWAs can run on any device with a modern web browser and are installed directly from the browser without the need for app stores. They are distributed through URLs and receive automatic updates, ensuring users always have the latest version. PWAs also offer offline functionality and leverage web storage mechanisms. While they may not have access to the full range of native device capabilities, PWAs provide a cost-effective and efficient way to deliver app-like experiences across different platforms and devices directly from the web.

The below steps have to be followed to create a progressive web application:

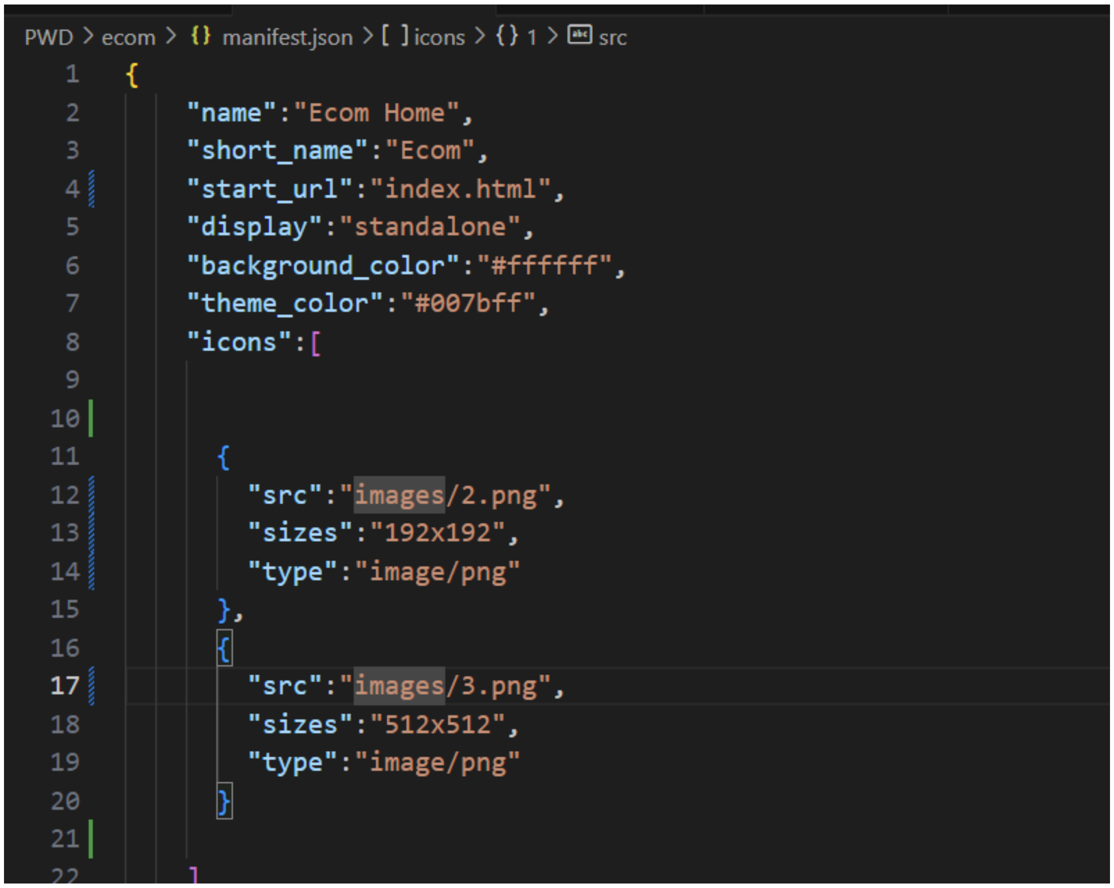
Step 1: Create an HTML page that would be the starting point of the application. This HTML will contain a link to the file named manifest.json. This is an important file that would be created in the next step.

Name: YOGESHCHANDRA PARANJPE Div: D15B RollNo.52



```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Laptop Store</title>
    <link rel="stylesheet" href="styles.css">
    <link rel="manifest" href="manifest.json">
</head>
<body>
    <header>
        <h1>
            Welcome to our store
        </h1>
        <nav>
            <ul>
                <li><a href="#">Home</a></li>
                <li><a href="#">Shop</a></li>
                <li><a href="#">About</a></li>
                <li><a href="#">Contact</a></li>
            </ul>
        </nav>
    </header>
```

Step 2: Create a manifest.json file in the same directory. This file basically contains information about the web application. Some basic information includes the application name, starting URL, theme color, and icons. All the information required is specified in the JSON format. The source and size of the icons are also defined in this file.



PWD > ecom > {} manifest.json > [ ] icons > {} 1 > src

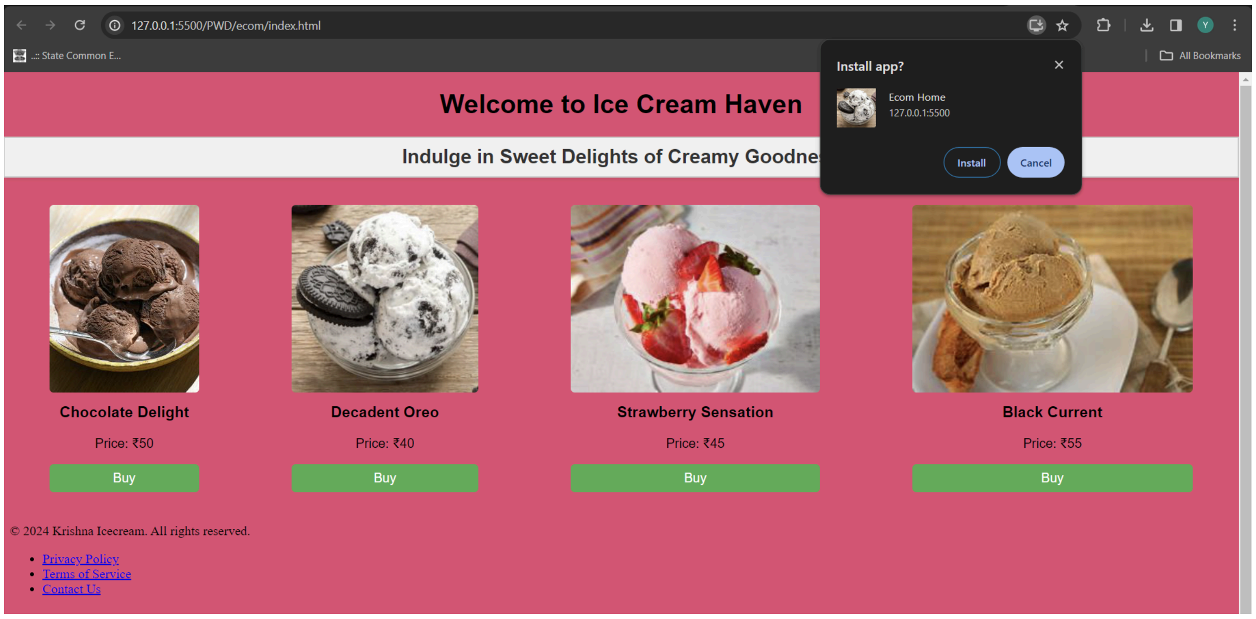
```
{
    "name":"Ecom Home",
    "short_name":"Ecom",
    "start_url":"index.html",
    "display":"standalone",
    "background_color":"#ffffff",
    "theme_color":"#007bff",
    "icons":[


        {
            "src":"images/2.png",
            "sizes":"192x192",
            "type":"image/png"
        },
        {
            "src":"images/3.png",
            "sizes":"512x512",
            "type":"image/png"
        }

    ]
```

Name: YOGESHCHANDRA PARANJPE Div: D15B RollNo.52

Step 3: Create a new folder named images and place all the icons related to the application in that folder. It is recommended to have the dimensions of the icons at least 192 by 192 pixels and 512 by 512 pixels. The image name and dimensions should match that of the manifest file.

Step 4: Serve the directory using a live server so that all files are accessible.

Step 5: Open the index.html file in Chrome navigate to the Application Section in the Chrome Developer Tools. Open the manifest column from the list.



Welcome to Ice Cream Haven

Indulge in Sweet Delights of Creamy Goodne

Chocolate Delight
Price: ₹50
Buy

Decadent Oreo
Price: ₹40
Buy

Strawberry Sensation
Price: ₹45
Buy

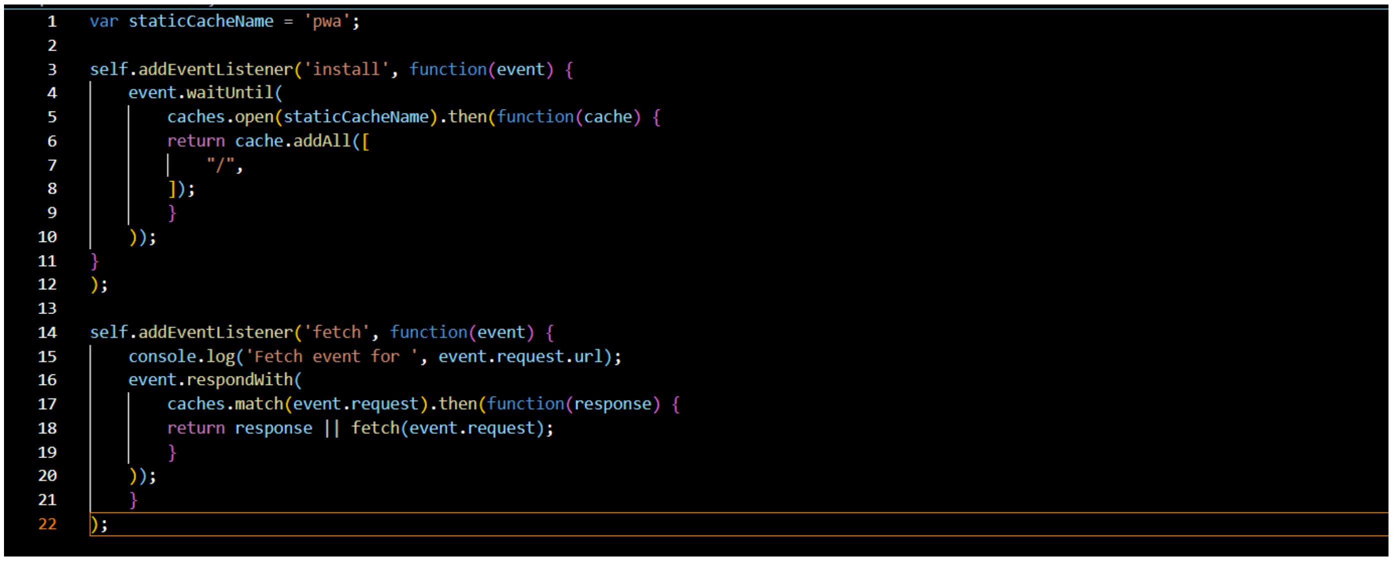
Black Current
Price: ₹55
Buy

© 2024 Krishna Icecream. All rights reserved.

- Privacy Policy
- Terms of Service
- Contact Us

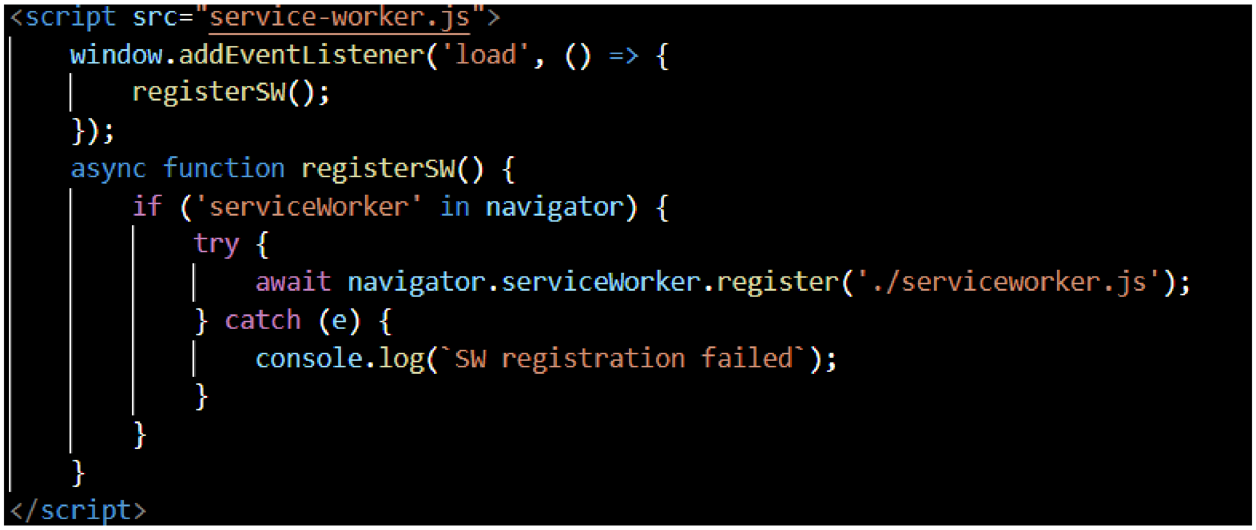
Step 6: Under the installability tab, it would show that no service worker is detected. We will need to create another file for the PWA, that is, serviceworker.js in the same directory. This file handles the configuration of a service worker that will manage the working of the application.

Name: YOGESHCHANDRA PARANJPE Div: D15B RollNo.52



```
var staticCacheName = 'pwa';

self.addEventListener('install', function(event) {
    event.waitUntil(
        caches.open(staticCacheName).then(function(cache) {
        return cache.addAll([
            "/",
        ]);
        }
    ));
}
);

self.addEventListener('fetch', function(event) {
    console.log('Fetch event for ', event.request.url);
    event.respondWith(
        caches.match(event.request).then(function(response) {
        return response || fetch(event.request);
        }
    ));
    }
);
```

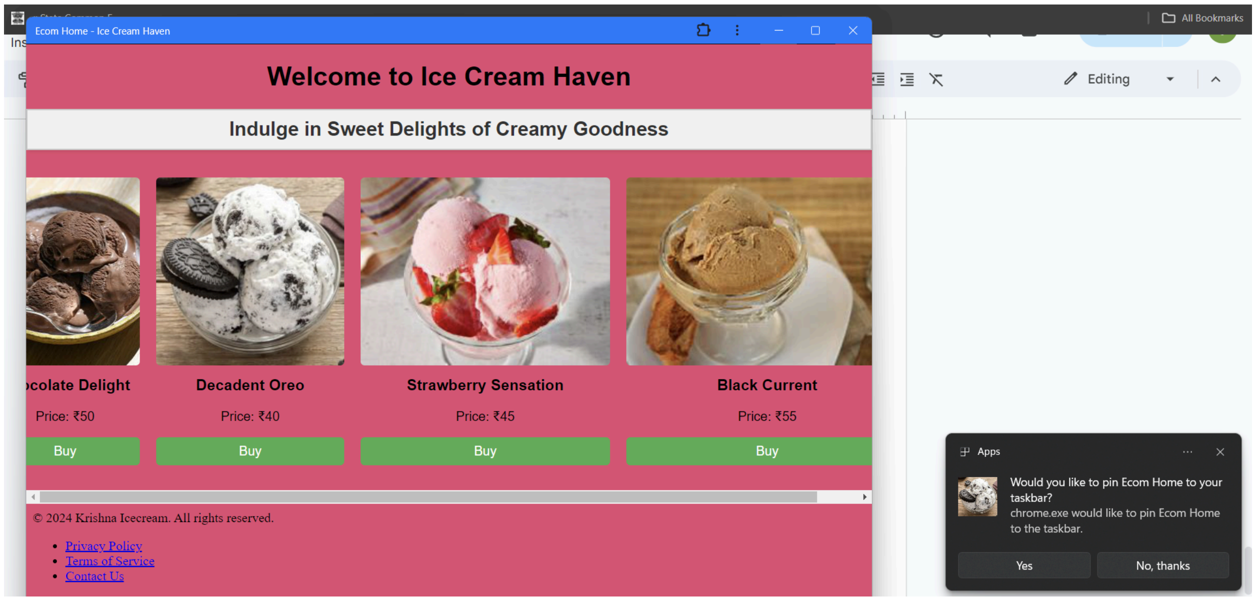
Step 7: The last step is to link the service worker file to index.html. This is done by adding a short JavaScript script to the index.html created in the above steps. Add the below code inside the script tag in index.html.
Installing the application:
Navigating to the Service Worker tab, we see that the service worker is registered successfully and now an install option will be displayed that will allow us to install our app.
Click on the install button to install the application. The application would then be installed, and it would be visible on the desktop.



```
<script src="service-worker.js">
    window.addEventListener('load', () => {
        registerSW();
    });
    async function registerSW() {
        if ('serviceWorker' in navigator) {
            try {
                await navigator.serviceWorker.register('./serviceworker.js');
            } catch (e) {
                console.log(`SW registration failed`);
            }
        }
    }
</script>
```

For installing the application on a mobile device, the Add to Home screen option in the mobile browser can be used. This will install the application on the device.

Name: YOGESHCHANDRA PARANJPE Div: D15B RollNo.52



Ecom Home - Ice Cream Haven

Welcome to Ice Cream Haven

Indulge in Sweet Delights of Creamy Goodness

colate Delight
Price: ₹50
Buy

Decadent Oreo
Price: ₹40
Buy

Strawberry Sensation
Price: ₹45
Buy

Black Current
Price: ₹55
Buy

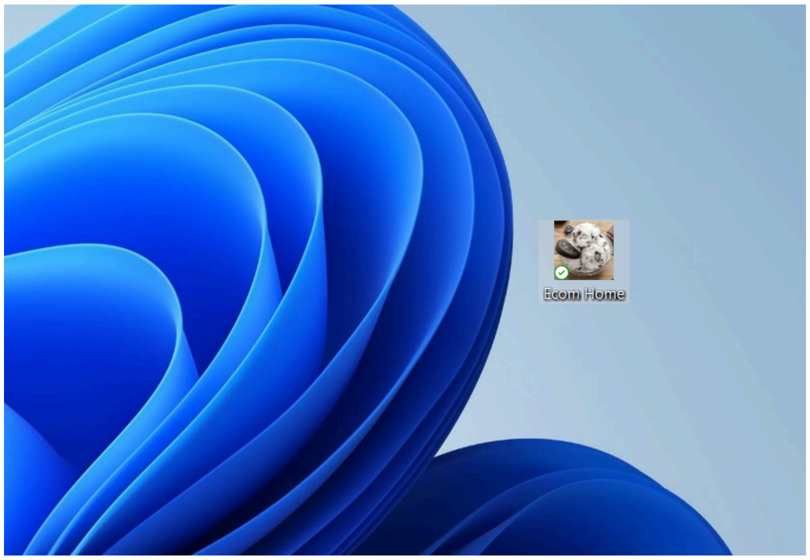
© 2024 Krishna Icecream. All rights reserved.

- Privacy Policy
- Terms of Service
- Contact Us

Apps

Would you like to pin Ecom Home to your taskbar?
chrome.exe would like to pin Ecom Home to the taskbar.

Yes | No, thanks



Conclusion: Thus writing metadata for the PWA, especially for an eCommerce application, is crucial for enabling features like the "add to homescreen" functionality. By crafting a well-structured manifest.json file with accurate metadata properties such as name, description, icons, and colors, developers can enhance the accessibility and user experience of their PWAs.