Ria Chaudhari D15A07
Eesha Chavan D15A08
Shraeyaa Dhaigude D15A15

PWA EXP7

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

Theory:

Regular Web App

A regular web app is a website that is designed to be accessible on all mobile devices such that
the content gets fit as per the device screen. It is designed using a web technology stack
(HTML, CSS, JavaScript, Ruby, etc.) and operates via a browser. They offer various native-device features and functionalities. However, it entirely depends on the browser the user
is using. In other words, it might be possible that you can access a native-device feature on
Chrome but not on Safari or Mozilla Firefox because the browsers are incompatible with that
feature.

Progressive Web App

Progressive Web App (PWA) is a regular web app, but some extras enable it to deliver an
excellent user experience. It is a perfect blend of desktop and mobile application experience to
give both platforms to the end-users.

Difference between PWAs vs. Regular Web Apps:

Progressive Web is different and better than a Regular Web app with features like:

1. Native Experience

Though a PWA runs on web technologies (HTML, CSS, JavaScript) like a Regular web app, it gives user experience like a native mobile application. It can use most native device features, including push notifications, without relying on the browser or any other entity. It offers a seamless and integrated user experience that it is quite tough for one to differentiate between a PWA and a Native application by considering its look and feel.

2. Ease of Access

Unlike other mobile apps, PWAs do not demand longer download time and make memory space available for installing the applications. The PWAs can be shared and installed by a link, which cuts down the number of steps to install and use. These applications can easily keep an app icon on the user's home screen, making the app easily accessible to the users and helps the brands remain in the users' minds, and improving the chances of interaction.

3. Faster Services
PWAs can cache the data and serve the user with text stylesheets, images, and other web content even before the page loads completely. This lowers the waiting time for the end-users and helps the brands improve the user engagement and retention rate, which eventually adds value to their business.

Features of the Progressive Web App
The main features are:
● Progressive
They work for every user, regardless of the browser chosen because they are built at the
base with progressive improvement principles.
● Responsive
They adapt to the various screen sizes: desktop, mobile, tablet, or dimensions that can
later become available.
● App-like
They behave with the user as if they were native apps, in terms of interaction and navigation.
● Updated
Information is always up-to-date thanks to the data update process offered by service
workers.
● Secure
Exposed over HTTPS protocol to prevent the connection from displaying information or
altering the contents.
● Searchable
They are identified as “applications” and are indexed by search engines.
● Reactivable
Make it easy to reactivate the application thanks to capabilities such as web notifications.
● Installable
They allow the user to “save” the apps that he considers most useful with the corresponding icon on the screen of his mobile terminal (home screen) without having to
face all the steps and problems related to the use of the app store.
● Linkable
Easily shared via URL without complex installations.

Code:

//manifest.json

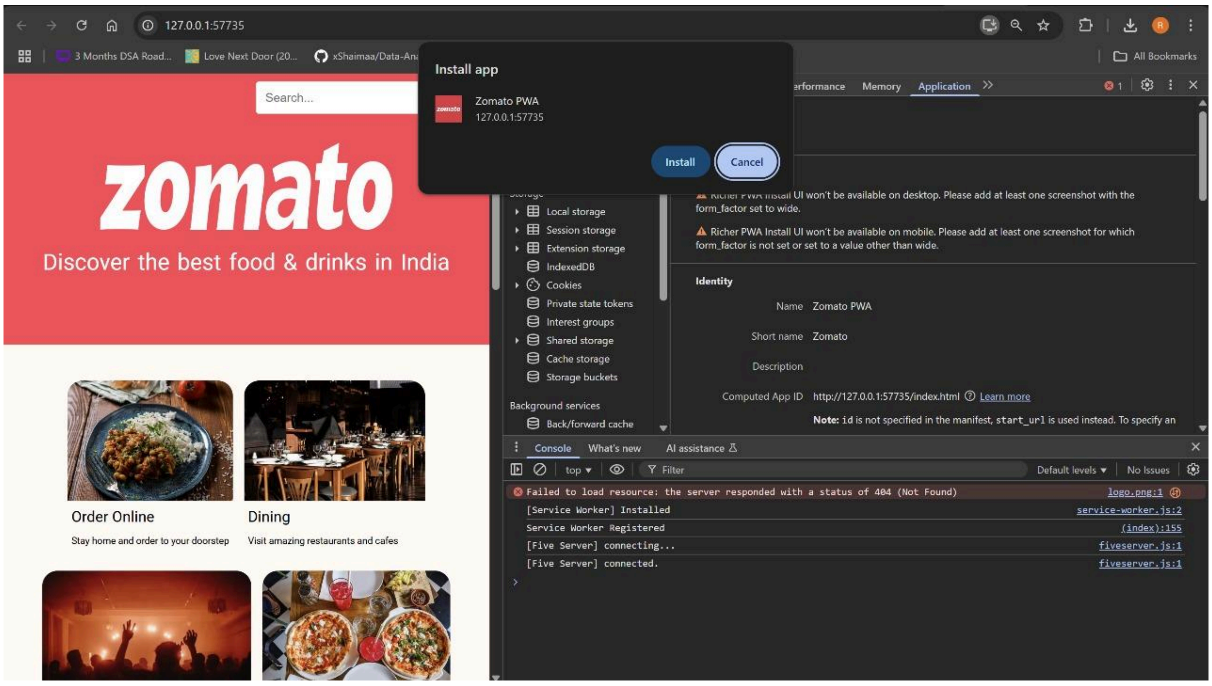
```
{
    "name": "Zomato PWA",
    "short_name": "Zomato",
    "start_url": "./index.html",
    "display": "standalone",
    "background_color": "#ffffff",
    "theme_color": "#ff5a5f",
    "orientation": "portrait",
    "scope": "./",
    "icons": [
      {
        "src": "images/icon-192x192.png",
        "sizes": "192x192",
        "type": "image/png"
      },
      {
        "src": "images/icon-512x512.png",
        "sizes": "512x512",
        "type": "image/png"
      }
    ]
  }
```

//service-worker.js

```
self.addEventListener('install', function(e) {
    console.log('[Service Worker] Installed');
  });

  self.addEventListener('fetch', function(e) {
    console.log('[Service Worker] Fetch intercepted:', e.request.url);
  });
```

Output:



Install app

Zomato PWA
127.0.0.1:57735

Install Cancel

Richer PWA Install UI won't be available on desktop. Please add at least one screenshot with the form_factor set to wide.

Richer PWA Install UI won't be available on mobile. Please add at least one screenshot for which form_factor is not set or set to a value other than wide.

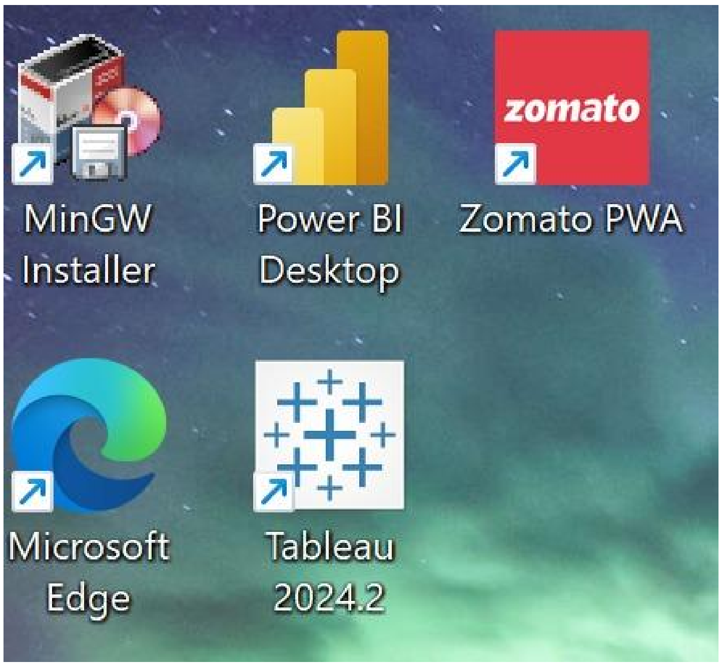
Identity

Name	Zomato PWA
Short name	Zomato
Description	
Computed App ID	http://127.0.0.1:57735/index.html Learn more

Note: id is not specified in the manifest, start_url is used instead. To specify an

```
Failed to load resource: the server responded with a status of 404 (Not Found)   logo.png:1
[Service Worker] Installed                                                service-worker.js:2
Service Worker Registered                                                        (index):155
[Five Server] connecting...                                                    fiveserver.js:1
[Five Server] connected.                                                       fiveserver.js:1
```

Shortcut added to home screen



Conclusion:

Hence, we learnt how to write a metadata of our E-commerce website PWA in a Web App Manifest File to enable add to homescreen feature.