**EXPERIMENT No. 7**

## 

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

### **Theory:**

### **Making a Vite React App a Progressive Web App (PWA)**

A Progressive Web App (PWA) is a type of web application that provides a reliable, fast, and engaging user experience similar to a native mobile app. It leverages modern web technologies, including service workers, caching strategies, and a web app manifest, to enable offline functionality, push notifications, and an installable experience.

In the context of a Vite React application, converting the app into a PWA involves integrating these key components:

1. Service Workers:  
   * Service workers act as a proxy between the browser and the network, enabling background processes such as caching and offline capabilities.
   * They allow the app to load even when there is no internet connection by storing static assets locally.
2. Web App Manifest:  
   * The manifest.json file provides metadata about the app, including its name, icons, theme color, and display mode.
   * This file helps the browser recognize the app as installable and controls its behavior when launched.
3. Vite PWA Plugin:  
   * Since Vite does not have built-in PWA support, a plugin like vite-plugin-pwa is required to handle service worker registration and manifest integration automatically.
   * This plugin simplifies the process of enabling caching, auto-updates, and background synchronization.

### Key Features of a PWA:

* Offline Availability: Allows users to access the app without an internet connection.
* Fast Performance: Uses caching strategies to reduce loading time.
* App-Like Interface: Can be installed on devices and launched in a standalone mode.
* Secure & Reliable: Served over HTTPS to prevent data tampering.
* Background Sync & Push Notifications: Keeps users engaged with real-time updates.

**Steps to Make Your Vite React App a PWA**

### **1️ Install Vite PWA Plugin**

1. Open your terminal inside the project folder.
2. Run the command to install the PWA plugin.
3. This plugin helps to add service workers and caching automatically.

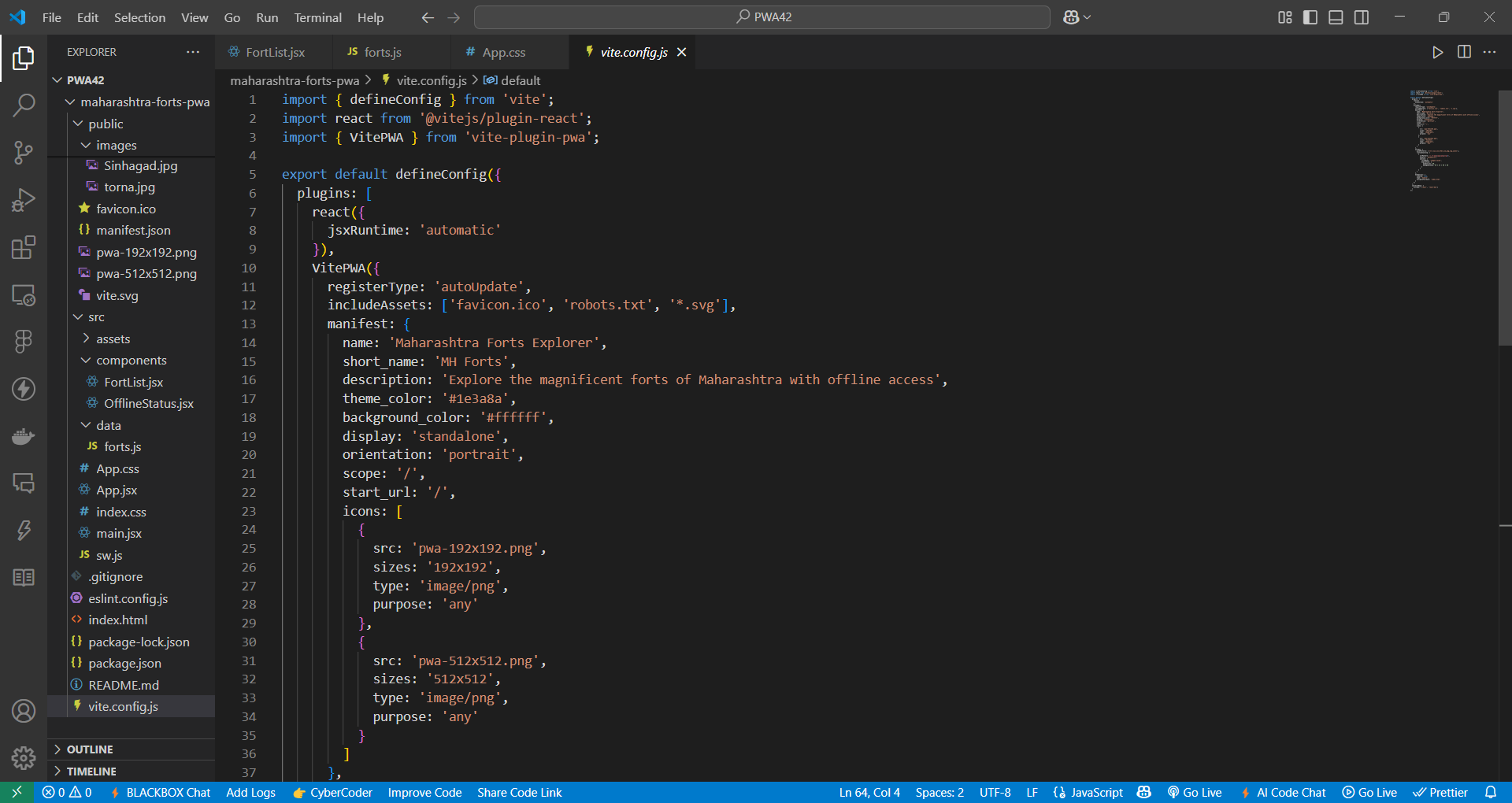
### **2️ Configure Vite for PWA**

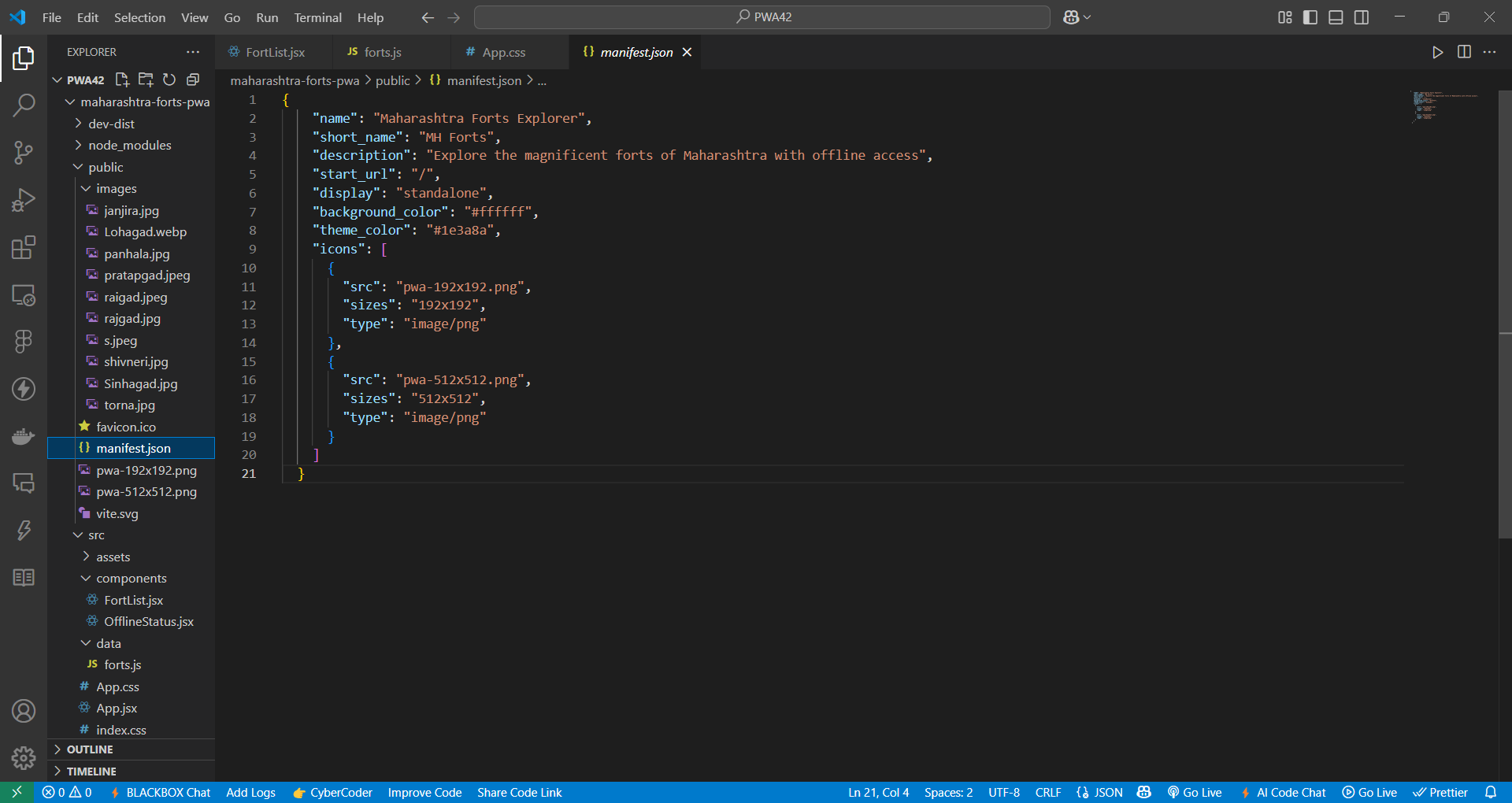
1. Open the vite.config.js file in your project.
2. Import and add the PWA plugin inside the plugins section.
3. Set up options like auto-updates, caching, and offline support.
4. Save the file and restart the Vite server.

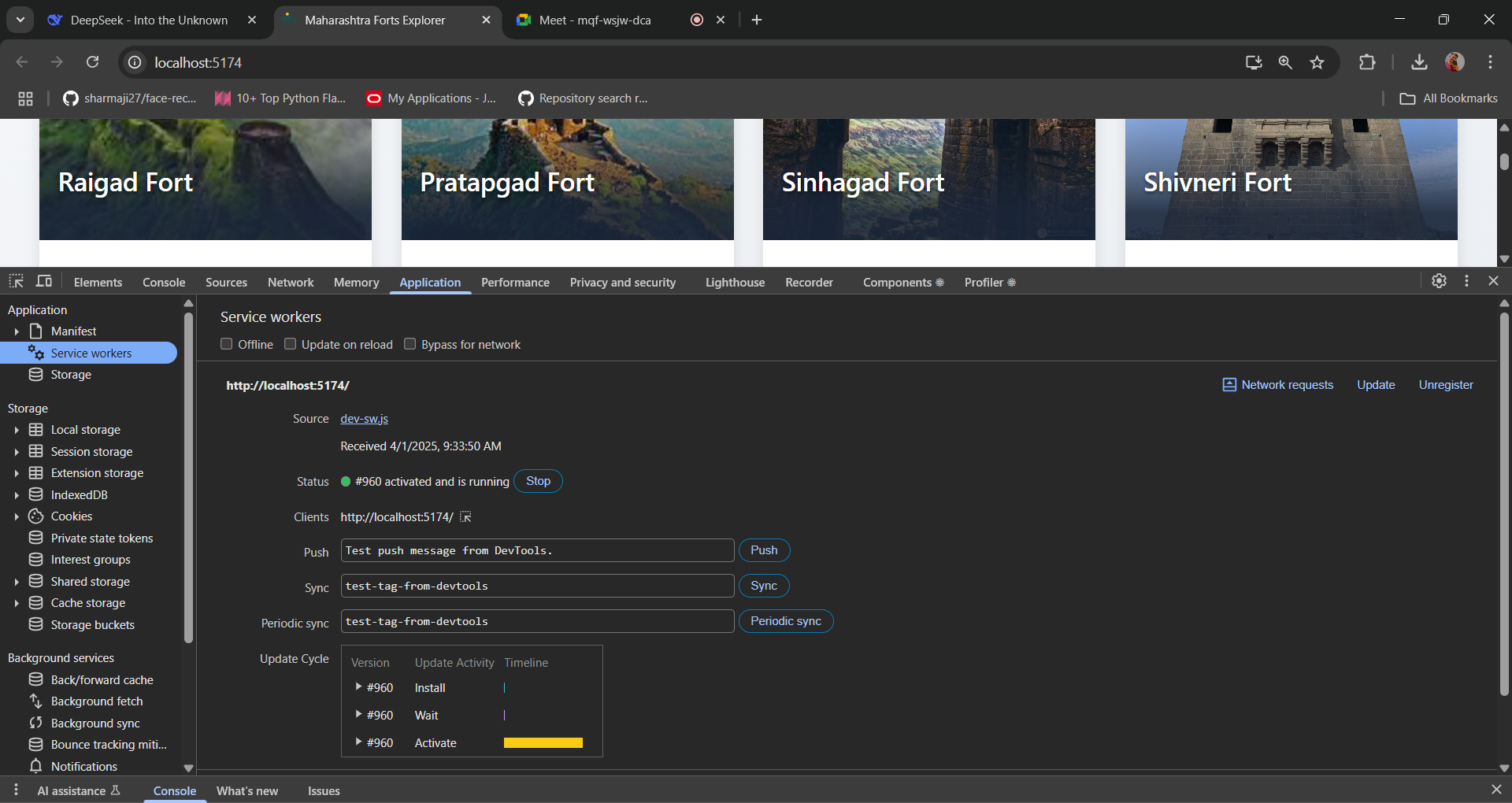
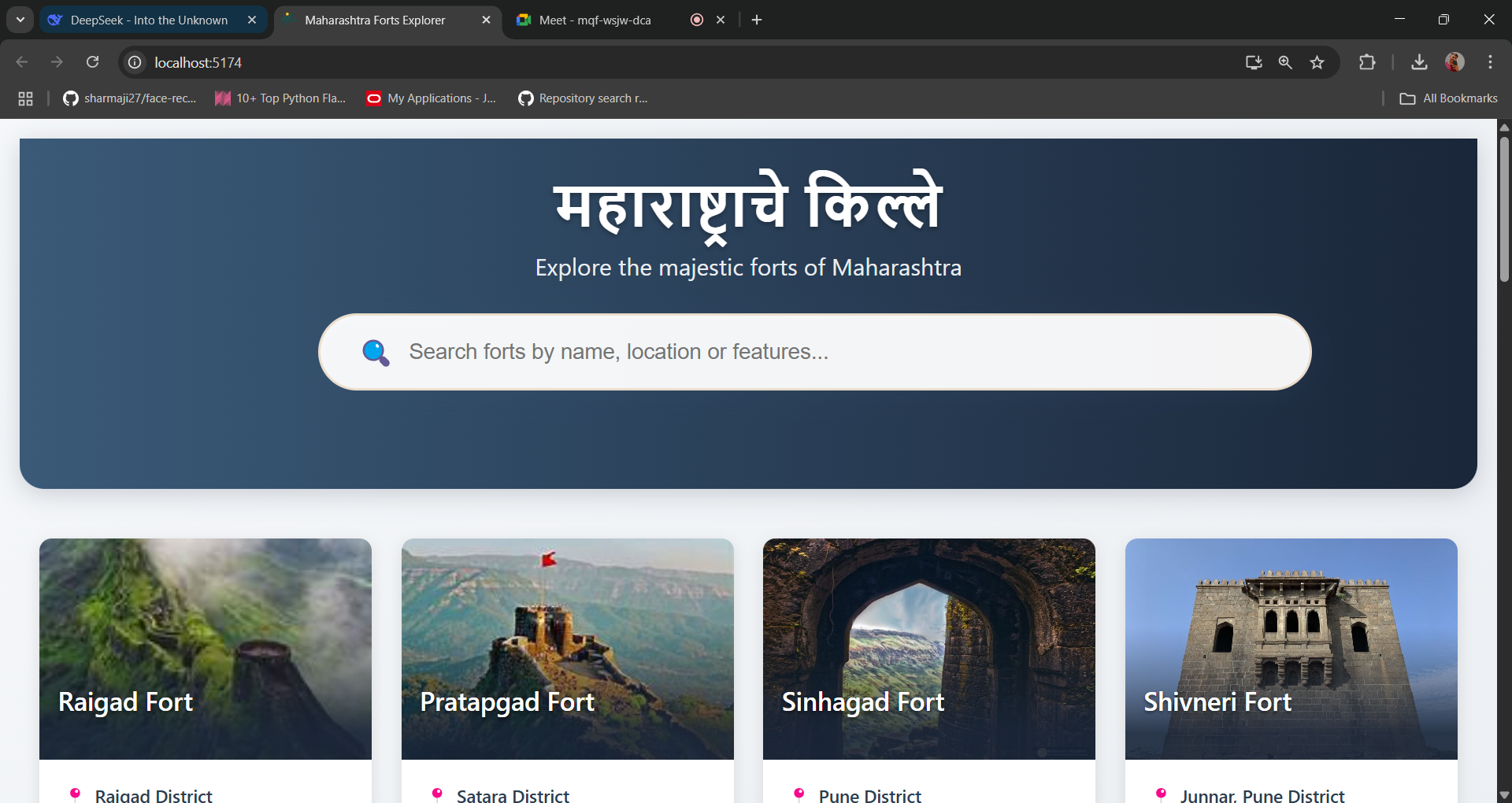
### **3️ Add manifest.json File**

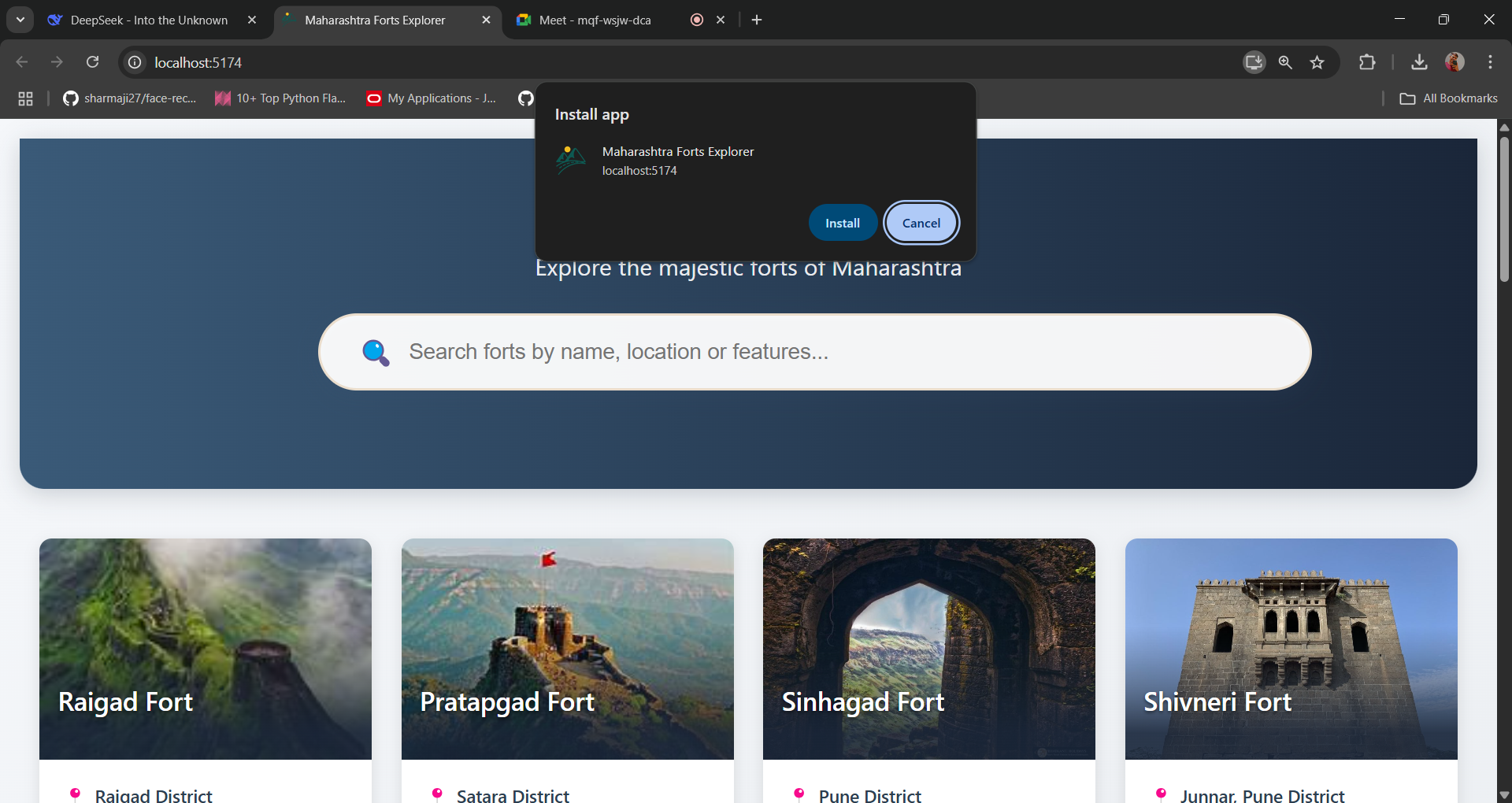
1. Go to the public folder in your project.
2. Create a new file named manifest.json.
3. Add details like the app name, icons, theme color, and display mode.
4. This file helps browsers recognize your app as installable**.**

**OUTPUT :**

****

****



**Conclusion:**

By following these steps, we successfully converted our Vite React application into a Progressive Web App. This enables offline accessibility, caching for better performance, and an installable app experience on mobile and desktop devices. Implementing PWA features enhances user engagement and makes the app function seamlessly even in limited network conditions.