**Project Setup**

**Step 1: Install Node.js and Create a React App**

First, install **Node.js** (if not already installed). Then, create a new React app.

bash

npx create-react-app restaurant-survey

cd restaurant-survey

npm start

**Step 2: Install PWA Support in React**

React does not enable PWA features by default, so enable them:

bash

npm install --save workbox-core workbox-precaching workbox-routing workbox-strategies

**2. Build the Survey Form**

**Step 3: Create a Survey Form in React**

Modify src/App.js to include a simple survey form:

javascript

import React, { useState } from "react";

function App() {

const [rating, setRating] = useState(null);

const [submitted, setSubmitted] = useState(false);

const handleSubmit = () => {

localStorage.setItem("surveyRating", rating);

setSubmitted(true);

};

return (

<div style={{ textAlign: "center", padding: "20px" }}>

{!submitted ? (

<>

<h2>How was your meal?</h2>

<button onClick={() => setRating("Excellent")}>Excellent</button>

<button onClick={() => setRating("Good")}>Good</button>

<button onClick={() => setRating("Average")}>Average</button>

<br />

<button onClick={handleSubmit} disabled={!rating}>Submit</button>

</>

) : (

<h2>Thank you! Enjoy a free dessert on us! 🍰</h2>

)}

</div>

);

}

export default App;

💡 **This simple form stores the rating in localStorage and displays a thank-you message.**

**3. Convert React App into a PWA**

**Step 4: Enable PWA Service Worker**

* Open src/index.js and modify:

javascript

import React from "react";

import ReactDOM from "react-dom";

import App from "./App";

import \* as serviceWorkerRegistration from "./serviceWorkerRegistration"; // Import the service worker

ReactDOM.render(<App />, document.getElementById("root"));

serviceWorkerRegistration.register(); // Register PWA

**Step 5: Add service-worker.js**

Create a public/service-worker.js file:

javascript

self.addEventListener("install", (event) => {

event.waitUntil(

caches.open("restaurant-survey-cache").then((cache) => {

return cache.addAll([

"/",

"/index.html",

"/static/js/bundle.js",

]);

})

);

});

self.addEventListener("fetch", (event) => {

event.respondWith(

caches.match(event.request).then((response) => {

return response || fetch(event.request);

})

);

});

🔹 **This allows offline functionality by caching essential assets.**

**Step 6: Modify public/manifest.json**

This file makes the app installable on iPads.

json

{

"short\_name": "Survey",

"name": "Restaurant Survey App",

"icons": [

{

"src": "logo192.png",

"type": "image/png",

"sizes": "192x192"

}

],

"start\_url": "/",

"display": "standalone",

"background\_color": "#ffffff",

"theme\_color": "#000000"

}

💡 **This makes the app launch in full-screen mode on an iPad.**

**4. Deploy Your PWA**

**Step 7: Build and Deploy**

Run:

bash

npm run build

This generates a build/ folder.

**Step 8: Deploy to Free Hosting (Netlify)**

* Install Netlify CLI:

bash

npm install -g netlify-cli

* Deploy:

bash

netlify deploy --prod

* **Copy the given URL** (e.g., https://restaurant-survey.netlify.app).

**5. Install & Use PWA on iPad**

**Step 9: Install on iPad**

1. **Open Safari** and visit your app URL (https://restaurant-survey.netlify.app).
2. **Tap the "Share" button** at the bottom of the screen.
3. **Select "Add to Home Screen"**.
4. The app will now appear **like a native iPad app** on the home screen!

**6. Extra Features (Optional)**

**✅ Store Data in Firebase Instead of LocalStorage**

* Install Firebase:

bash

npm install firebase

* Modify handleSubmit to store in Firestore:

javascript

CopyEdit

import { getFirestore, collection, addDoc } from "firebase/firestore";

import { initializeApp } from "firebase/app";

const firebaseConfig = { apiKey: "your-api-key", authDomain: "your-app.firebaseapp.com", projectId: "your-project-id" };

const app = initializeApp(firebaseConfig);

const db = getFirestore(app);

const handleSubmit = async () => {

await addDoc(collection(db, "surveys"), { rating });

setSubmitted(true);

};

🎉 **Now the data is stored in Firebase instead of locally!**

**7. Summary**

| **Step** | **Task** | **Tools** |
| --- | --- | --- |
| 1️⃣ | Create React App | React.js |
| 2️⃣ | Build Survey Form | React.js |
| 3️⃣ | Add PWA Features | Service Worker, manifest.json |
| 4️⃣ | Deploy | Netlify (Free) |
| 5️⃣ | Install on iPad | Safari “Add to Home Screen” |
| 6️⃣ | (Optional) Store Data Online | Firebase Firestore |