

Unten findest du ein **komplettes, kostenloses Starter-Projekt** für eine offlinefähige **PWA CO₂-Rechner-Web-App**, optimiert für **Vercel**. Enthalten sind:

- Client-seitige CO₂-Berechnung (keine Serverkosten)
- Teilen per Web Share API, WhatsApp, E-Mail
- Push-Benachrichtigungen via **Firebase Cloud Messaging (FCM)** (Free Tier), inkl. Platzhalter
- Offline-Support via Service Worker + Cache Storage
- **PWA** (manifest.json, Icons) + **Vercel-Konfiguration**

Hinweis: Setze deine eigenen Werte (Firebase Keys, App-Namen, Icons) an den markierten Stellen ein.

Projektstruktur

```
co2-pwa/
├── public/
│   ├── icons/
│   │   ├── icon-192.png
│   │   └── icon-512.png
│   ├── manifest.webmanifest
│   ├── sw.js
│   ├── firebase-messaging-sw.js
│   └── favicon.ico
├── index.html
├── styles.css
└── app.js
└── push.js
└── vercel.json
└── README.md
```

index.html

```
<!doctype html>
<html lang="de">
<head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <title>CO2 Rechner PWA</title>
    <link rel="manifest" href="/public/manifest.webmanifest">
    <link rel="icon" href="/public/favicon.ico">
    <meta name="theme-color" content="#0ea5e9" />
    <link rel="stylesheet" href="/styles.css" />
</head>
<body>
    <header>
```

```

<h1>CO2-Rechner</h1>
<button id="btn-install" hidden>App installieren</button>
</header>

<main>
  <section class="card">
    <h2>Berechnung</h2>
    <form id="form-co2">
      <label>Strecke (km)
        <input type="number" min="0" step="0.1" id="distance" required />
      </label>
      <label>Verkehrsmittel
        <select id="mode" required>
          <option value="car">Auto</option>
          <option value="bus">Bus</option>
          <option value="train">Bahn</option>
          <option value="plane">Flugzeug (Kurzstrecke)</option>
        </select>
      </label>
      <button type="submit">Berechnen</button>
    </form>
    <div id="result" class="result" aria-live="polite"></div>
    <div class="share">
      <button id="btn-share" disabled>Teilen</button>
      <a id="btn-wa" class="link" target="_blank" rel="noopener">WhatsApp</
a>
      <a id="btn-mail" class="link" href="mailto:">>E-Mail</a>
    </div>
  </section>

  <section class="card">
    <h2>Push-Benachrichtigungen</h2>
    <p>Optional: Aktiviere Push, um Erinnerungen oder Tipps zu erhalten.</
p>
    <button id="btn-push">Push aktivieren</button>
    <pre id="push-status"></pre>
  </section>
</main>

<footer>
  <small>Offline-fähig • Datenschutz: Alle Berechnungen im Browser</small>
</footer>

<script src="/app.js" type="module"></script>
<script src="/push.js" type="module"></script>
</body>
</html>

```

styles.css

```
:root { --bg:#0b1020; --fg:#eaf2ff; --card:#121a2f; --accent:#0ea5e9; }
*{box-sizing:border-box} body{margin:0;font-family:system-ui,Segoe
UI,Roboto,Arial;background:var(--bg);color:var(--fg)}
header,footer{padding:1rem 1.25rem;display:flex;gap:.75rem;align-
items:center;justify-content:space-between;border-bottom:1px solid #223}
footer{border-top:1px solid #223;border-bottom:none}
h1{margin:0;font-size:1.35rem} h2{margin:.25rem 0 .75rem}
main{padding:1.25rem;display:grid;gap:1rem;grid-template-columns:repeat(auto-
fit,minmax(320px,1fr))}
.card{background:var(--card);border:1px solid #223;border-radius:
16px;padding:1rem;box-shadow:0 6px 24px rgba(0,0,0,.25)}
label{display:flex;flex-direction:column;gap:.35rem;margin:.5rem 0}
input,select,button,.link{padding:.7rem .85rem;border-radius:10px;border:1px
solid #2a395a;background:#0f1629;color:var(--fg)}
button{cursor:pointer} button:disabled{opacity:.5;cursor:not-allowed}
.result{margin-top:.75rem;padding:.75rem;border-radius:
10px;background:#0f1b34;border:1px solid #243450}
.share{display:flex;gap:.5rem;margin-top:.5rem}
#btn-install{background:var(--accent);border:0;color:#001226}
.link{display:inline-flex;align-items:center;text-decoration:none}
```

app.js

```
// --- PWA Installation prompt handling ---
let deferredPrompt;
const installBtn = document.getElementById('btn-install');
window.addEventListener('beforeinstallprompt', (e) => {
  e.preventDefault();
  deferredPrompt = e;
  installBtn.hidden = false;
});
installBtn?.addEventListener('click', async () => {
  installBtn.hidden = true;
  await deferredPrompt?.prompt();
  deferredPrompt = null;
});

// --- Service Worker Registration ---
if ('serviceWorker' in navigator) {
  navigator.serviceWorker.register('/public/sw.js');
}

// --- CO2 calculation ---
const EF = {
  car: 171, // g CO2 / km (Durchschnitt, vereinfacht)
```

```

bus: 104,    // g CO2 / km
train: 41,   // g CO2 / km (Fernverkehr ca.)
plane: 255,  // g CO2 / km (Kurzstrecke, stark vereinfacht)
};

const form = document.getElementById('form-co2');
const resultEl = document.getElementById('result');
const btnShare = document.getElementById('btn-share');
const btnWA = document.getElementById('btn-wa');
const btnMail = document.getElementById('btn-mail');

let lastText = '';

form.addEventListener('submit', (e) => {
  e.preventDefault();
  const km = parseFloat(document.getElementById('distance').value);
  const mode = document.getElementById('mode').value;
  if (!Number.isFinite(km) || km <= 0) return;
  const grams = km * (EF[mode] ?? 0);
  const kg = grams / 1000;
  lastText = `Für ${km.toFixed(1)} km per ${mode} fallen ca. $ ${kg.toFixed(2)} kg CO2 an.`;
  resultEl.textContent = lastText + ' (vereinfachte Schätzung)';
  btnShare.disabled = false;
  // Set share links
  btnWA.href = `https://wa.me/?text=${encodeURIComponent(lastText)}`;
  btnMail.href = `mailto:?subject=${encodeURIComponent('Mein CO2-Ergebnis')}&body=${encodeURIComponent(lastText)}`;
});

// --- Web Share API (native share sheet) ---
btnShare.addEventListener('click', async () => {
  try {
    if (navigator.share) {
      await navigator.share({ title: 'CO2-Ergebnis', text: lastText, url: location.href });
    } else {
      alert('Teilen wird von diesem Browser nicht nativ unterstützt. Nutze WhatsApp/E-Mail.');
    }
  } catch (e) {
    console.warn('Share canceled/failed', e);
  }
});

```

push.js

```
// Push via Firebase Cloud Messaging (kostenloses Free Tier)
// 1) Firebase Web-App anlegen -> Config hier einfügen
const firebaseConfig = {
  apiKey: 'YOUR_API_KEY',
  authDomain: 'YOUR_PROJECT.firebaseio.com',
  projectId: 'YOUR_PROJECT_ID',
  storageBucket: 'YOUR_PROJECT.appspot.com',
  messagingSenderId: 'YOUR_SENDER_ID',
  appId: 'YOUR_APP_ID',
  measurementId: 'G-XXXXXXX'
};
const vapidKey = 'YOUR_PUBLIC_VAPID_KEY'; // Firebase Messaging -> Web Push
Zertifikate

const btnPush = document.getElementById('btn-push');
const out = document.getElementById('push-status');

async function loadFirebase() {
  // dynamisch laden, keine Bundler nötig
  const [{ initializeApp }, { getMessaging, getToken, onMessage }] = await
Promise.all([
  import('https://www.gstatic.com/firebasejs/10.12.4.firebaseio-app.js'),
  import('https://www.gstatic.com/firebasejs/10.12.4/firebase-
messaging.js'),
]);
  const app = initializeApp(firebaseConfig);
  const messaging = getMessaging(app);
  return { getToken, onMessage, messaging };
}

btnPush?.addEventListener('click', async () => {
  try {
    if (!('Notification' in window)) throw new Error('Browser unterstützt
Notifications nicht');
    const perm = await Notification.requestPermission();
    if (perm !== 'granted') throw new Error('Benachrichtigungen nicht
erlaubt');

    const { getToken, onMessage, messaging } = await loadFirebase();
    // Service Worker für FCM muss unter /public.firebaseio-messaging-sw.js
liegen
    const reg = await navigator.serviceWorker.register('/public.firebaseio-
messaging-sw.js');

    const token = await getToken(messaging, { vapidKey,
serviceWorkerRegistration: reg });
    out.textContent = 'FCM Token:\n' + token + '\n\n(Sende Test-Push über
Firebase Console / Server)';
  }
})
```

```
onMessage(messaging, (payload) => {
  console.log('FG message', payload);
  out.textContent = 'Nachricht empfangen (Foreground):\n' +
JSON.stringify(payload, null, 2);
});
} catch (e) {
  out.textContent = 'Fehler: ' + e.message;
}
});
```

public/manifest.webmanifest

```
{
  "name": "CO2 Rechner",
  "short_name": "CO2",
  "start_url": "/",
  "scope": "/",
  "display": "standalone",
  "background_color": "#0b1020",
  "theme_color": "#0ea5e9",
  "icons": [
    { "src": "/public/icons/icon-192.png", "sizes": "192x192", "type": "image/png" },
    { "src": "/public/icons/icon-512.png", "sizes": "512x512", "type": "image/png" }
  ]
}
```

public/sw.js (Service Worker: Offline-Cache)

```
const CACHE_NAME = 'co2-pwa-v1';
const ASSETS = [
  '/',
  '/index.html',
  '/styles.css',
  '/app.js',
  '/push.js',
  '/public/manifest.webmanifest',
  '/public/icons/icon-192.png',
  '/public/icons/icon-512.png'
];
self.addEventListener('install', (e) => {
  e.waitUntil(caches.open(CACHE_NAME).then((c) => c.addAll(ASSETS)));
});
```

```

    self.addEventListener('activate', (e) => {
      e.waitUntil(
        caches.keys().then((keys) => Promise.all(keys.map((k) => (k !==
CACHE_NAME ? caches.delete(k) : null))))
      );
    });
    self.addEventListener('fetch', (e) => {
      e.respondWith(
        caches.match(e.request).then((cached) => cached || fetch(e.request))
      );
    });
  });

```

public/firebase-messaging-sw.js (FCM Background-Push)

```

// Läuft im Service Worker Kontext für FCM
importScripts('https://www.gstatic.com/firebasejs/10.12.4.firebaseio-app-
compat.js');
importScripts('https://www.gstatic.com/firebasejs/10.12.4/firebase-messaging-
compat.js');

firebase.initializeApp({
  apiKey: 'YOUR_API_KEY',
  authDomain: 'YOUR_PROJECT.firebaseio.com',
  projectId: 'YOUR_PROJECT_ID',
  storageBucket: 'YOUR_PROJECT.appspot.com',
  messagingSenderId: 'YOUR_SENDER_ID',
  appId: 'YOUR_APP_ID'
});

const messaging = firebase.messaging();

messaging.onBackgroundMessage((payload) => {
  const { title = 'Push', body = '', icon = '/public/icons/icon-192.png' } =
payload?.notification || {};
  self.registration.showNotification(title, { body, icon });
});

```

vercel.json

```
{
  "cleanUrls": true,
  "headers": [
    {
      "source": "/public/(.*)",
      "headers": [

```

```

        { "key": "Cache-Control", "value": "public, max-age=31536000,
immutable" }
    ]
}
]
}

```

README.md (Kurz-Anleitung)

```
# CO2 Rechner PWA (Vercel)

## Setup
1. **Firebase** (optional für Push):
   - Projekt erstellen → Web App anlegen → Config + VAPID Key kopieren.
   - In `push.js` und `public.firebaseio-messaging-sw.js` eintragen.
2. (Optional) **Icons** ersetzen in `public/icons/`.

## Lokales Testen
- Einfach mit einem Static Server starten (z. B. `npx serve` oder VSCode Live Server).
- `https` ist für Push/Install auf manchen Plattformen Pflicht (Vercel liefert automatisch https).

## Deployment auf Vercel (kostenlos)
- Repo auf GitHub/GitLab/Bitbucket pushen.
- In Vercel importieren → Build Command: **none**, Output Dir: **/** (Root).
- Alternativ: `vercel` CLI im Projektordner ausführen.

## Nutzung
- App aufrufen → Formular ausfüllen → Ergebnis teilen via Button (Web Share / WhatsApp / E-Mail).
- Über "Push aktivieren" Berechtigung erteilen und FCM Token testen.

## Datenschutz
- Alle Berechnungen finden im Browser statt.
- Es werden ohne Firebase/Push keine Daten an Server gesendet.
```

Nächste Schritte (optional)

- **Genauere Emissionsfaktoren** und Einheiten (z. B. nach UBA/DEFRA) einpflegen.
- **IndexedDB** für Berechnungsverlauf + Export/Import.
- **iOS-Optimierungen** (Add to Home Screen Hinweise, Touch-Icons).
- **Analytics** nur DSGVO-konform oder ganz weglassen.

Wenn du willst, passe ich dir das Template direkt mit deinen Texten/Logos/Emissionsfaktoren an.