service-worker.js

```
// LiveStrike Service Worker
const CACHE_NAME = 'livestrike-cache-v3';

// Core files (static only, not dynamic PHP)
const ASSETS_TO_CACHE = [
  './',
  './index.php',
  './landing-page.php',
  './dashboard.php',
  './offline.html', //  Custom offline fallback page
  './assets/images/logo-192.png',
  './assets/images/logo-512.png',
];

// INSTALL: Pre-cache static assets + offline page
self.addEventListener('install', (event) => {
  event.waitUntil(
    caches.open(CACHE_NAME).then((cache) => cache.addAll(ASSETS_TO_CACHE))
  );
  self.skipWaiting();
});

// ACTIVATE: Clean old caches
self.addEventListener('activate', (event) => {
  event.waitUntil(
    caches.keys().then((cacheNames) => {
      return Promise.all(
        cacheNames.map((cache) => {
          if (cache !== CACHE_NAME) {
            return caches.delete(cache);
          }
        })
      );
    })
  );
  self.clients.claim();
});

// FETCH: Handle requests
self.addEventListener('fetch', (event) => {
  const request = event.request;
  const url = new URL(request.url);

  //      Only handle GET requests for caching
  if (request.method !== 'GET') {
    // Still allow network fetches for POST/PUT/DELETE etc.
    return;
  }

```

```
  // 1 Dynamic PHP pages (scoreboard, manage-matches, etc.)
  if (url.pathname.endsWith('.php')) {
    event.respondWith(
      fetch(request)
        .then((response) => {
          const clone = response.clone();
          caches.open(CACHE_NAME).then((cache) => cache.put(request, clone));
          return response;
        })
        .catch(() =>
          caches.match(request).then((cached) => cached || caches.match('./offline.html'))
        )
    );
    return;
  }

  // 2 API calls (network-first, fallback to cache)
  if (url.pathname.startsWith('/api/')) {
    event.respondWith(
      fetch(request)
        .then((response) => {
          const clone = response.clone();
          caches.open(CACHE_NAME).then((cache) => cache.put(request, clone));
          return response;
        })
        .catch(() => caches.match(request))
    );
    return;
  }

  // 3 Static assets (cache-first)
  event.respondWith(
    caches.match(request).then((cachedResponse) => {
      return (
        cachedResponse ||
        fetch(request).catch(() => caches.match('./offline.html'))
      );
    })
  );
});

```