*import* { precacheAndRoute } *from* "workbox-precaching"**;**

*import* { registerRoute } *from* "workbox-routing"**;**

*import* {

  NetworkFirst**,**

  StaleWhileRevalidate**,**

  CacheFirst**,**

} *from* "workbox-strategies"**;**

*import* { ExpirationPlugin } *from* "workbox-expiration"**;**

*import* { BackgroundSyncPlugin } *from* "workbox-background-sync"**;**

*import* { CacheableResponsePlugin } *from* "workbox-cacheable-response"**;**

const CACHE\_VERSION **=** "v6"**;**

const API\_CACHE **=** `food-api-${CACHE\_VERSION}`**;**

const STATIC\_CACHE **=** `static-assets-${CACHE\_VERSION}`**;**

const IMAGE\_CACHE **=** `images-${CACHE\_VERSION}`**;**

*// 1. Precaching Configuration*

precacheAndRoute(self**.***\_\_WB\_MANIFEST*)**;**

*// 2. API Caching with Background Sync*

registerRoute(

  ({ **url** }) **=>** url**.***href***.**includes("/api/food/get")**,**

**new** NetworkFirst({

    cacheName**:** API\_CACHE**,**

    plugins**:** [

**new** BackgroundSyncPlugin("food-queue"**,** {

        maxRetentionTime**:** 24 **\*** 60**,** *// 24 hours*

      })**,**

**new** ExpirationPlugin({

        maxEntries**:** 50**,**

        maxAgeSeconds**:** 86400**,** *// 24 hours*

      })**,**

**new** CacheableResponsePlugin({

        statuses**:** [0**,** 200]**,**

      })**,**

    ]**,**

  })**,**

  "GET"

)**;**

*// 3. Static Assets Caching*

registerRoute(

  ({ **request** }) **=>** ["script"**,** "style"**,** "font"]**.**includes(request**.***destination*)**,**

**new** StaleWhileRevalidate({

    cacheName**:** STATIC\_CACHE**,**

    plugins**:** [

**new** ExpirationPlugin({

        maxEntries**:** 200**,**

        maxAgeSeconds**:** 60 **\*** 60 **\*** 24 **\*** 30**,** *// 30 days*

      })**,**

    ]**,**

  })

)**;**

*// 4. Image Caching*

registerRoute(

  ({ **request** }) **=>** request**.***destination* **===** "image"**,**

**new** CacheFirst({

    cacheName**:** IMAGE\_CACHE**,**

    plugins**:** [

**new** ExpirationPlugin({

        maxEntries**:** 100**,**

        maxAgeSeconds**:** 60 **\*** 60 **\*** 24 **\*** 30**,** *// 30 days*

      })**,**

    ]**,**

  })

)**;**

*// 5. Navigation Fallback*

registerRoute(

  ({ **request** }) **=>** request**.***mode* **===** "navigate"**,**

**new** NetworkFirst({

    cacheName**:** "pages-cache"**,**

    networkTimeoutSeconds**:** 3**,**

    plugins**:** [

**new** ExpirationPlugin({

        maxEntries**:** 50**,**

        maxAgeSeconds**:** 60 **\*** 60 **\*** 24 **\*** 7**,** *// 7 days*

      })**,**

    ]**,**

  })

)**;**

*// Service Worker Lifecycle Events (Keep only these)*

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

  console**.**log("[SW] Install event")**;**

  event**.**waitUntil(

    caches

**.**open(IMAGE\_CACHE)

**.**then((**cache**) **=>** {

        console**.**log("[SW] Pre-caching essential assets")**;**

**return** cache**.**addAll([

          "/offline"**,**

          "/icons/icon-192x192.png"**,**

          "/icons/icon-512x512.png"**,**

        ])**;**

      })

**.**catch((**error**) **=>** {

        console**.**error("[SW] Install failed:"**,** error)**;**

      })

  )**;**

})**;**

self**.**addEventListener("activate"**,** (**event**) **=>** {

  event**.**waitUntil(self**.***clients***.**claim())**;**

  self**.**skipWaiting()**;** *// Force activate new SW*

})**;**

self**.**addEventListener("activate"**,** (**event**) **=>** {

  console**.**log("[SW] Activate event")**;**

  event**.**waitUntil(

    caches

**.**keys()

**.**then((**cacheNames**) **=>** {

**return** Promise**.**all(

          cacheNames**.**map((**cacheName**) **=>** {

**if** (**!**cacheName**.**includes(CACHE\_VERSION)) {

              console**.**log("[SW] Deleting old cache:"**,** cacheName)**;**

**return** caches**.**delete(cacheName)**;**

            }

          })

        )**;**

      })

**.**then(() **=>** self**.***clients***.**claim())

  )**;**

})**;**