

Service-worker.js

```
const CACHE_NAME = 'horizon-cache-v1';
const urlsToCache = [
  '/',
  '/index.html',
  '/about.html',
  '/styles.css',
  '/script.js',
  '/icons/icon-192x192.png',
  '/icons/icon-512x512.png'
];

// Install event - cache assets
self.addEventListener('install', event => {
  event.waitUntil(
    caches.open(CACHE_NAME)
      .then(cache => {
        console.log('Opened cache');
        return cache.addAll(urlsToCache);
      })
  );
});

// Activate event - clean up old caches
self.addEventListener('activate', event => {
  const cacheWhitelist = [CACHE_NAME];
  event.waitUntil(
    caches.keys().then(cacheNames => {
      return Promise.all(
        cacheNames.map(cacheName => {
          if (cacheWhitelist.indexOf(cacheName) === -1) {
            return caches.delete(cacheName);
          }
        })
      );
    })
  );
});

// Fetch event - serve from cache or network
self.addEventListener('fetch', event => {
  event.respondWith(
```

```

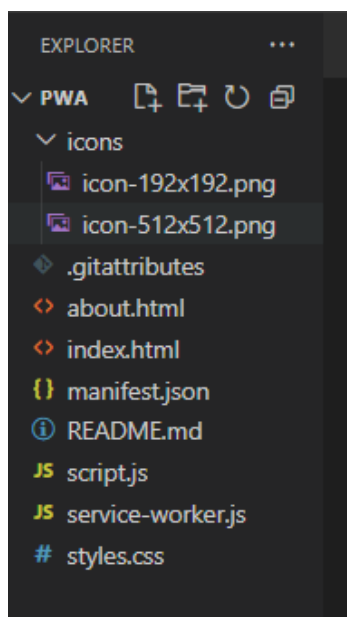
caches.match(event.request)
  .then(response => {
    // Cache hit - return response
    if (response) {
      return response;
    }
    return fetch(event.request).then(
      response => {
        // Check if we received a valid response
        if (!response || response.status !== 200 || response.type !== 'basic') {
          return response;
        }

        // Clone the response
        const responseToCache = response.clone();

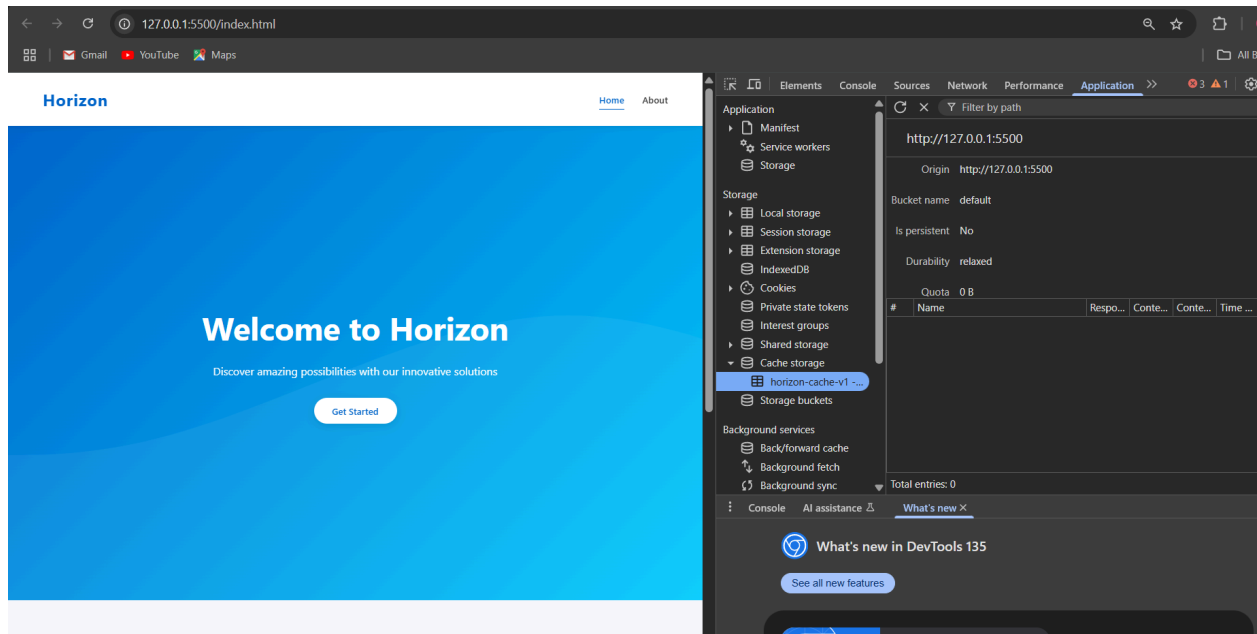
        caches.open(CACHE_NAME)
          .then(cache => {
            cache.put(event.request, responseToCache);
          });

        return response;
      }
    );
  })
);
});

```



Cache storage



Service workers

