Process:

- The moment user request a page , the fetch will intercept the request

- Fetch will return a promise .

- We ll open the dynamiccaches and put the - caches.open

- We ll put the request into the caches - caches.put with the response clone returned from the fetch promise

- Return the response

- If promise got rejected fetch will go to caches (pre-cache) to match the request URL request

11. Installation Steps / Performance Steps and Results –

Q1	Write a code for dynamic caching in service worker file and include it in ecommerce website.

Source code:

```javascript
const dynamicCacheName = 'site-dynamic-v1';
// install event
self.addEventListener('install', evt => {
  console.log('service worker installed');
});

// activate event
self.addEventListener('activate', evt => {
  console.log('service worker activated');
});

// fetch event
self.addEventListener('fetch', evt => {
  //console.log('fetch event', evt);
  evt.respondWith(
    caches.match(evt.request).then(cacheRes => {
      return cacheRes || fetch(evt.request).then(fetchRes => {
        return caches.open(dynamicCacheName).then(cache => {
          cache.put(evt.request.url, fetchRes.clone());
          return fetchRes;
        })
      });
    })
  );
});
```

Output: