Experiment No: 08

Aim: To implement service worker registration and complete the install and activation process for a new service worker for an E-commerce Progressive Web App (PWA).

Theory:

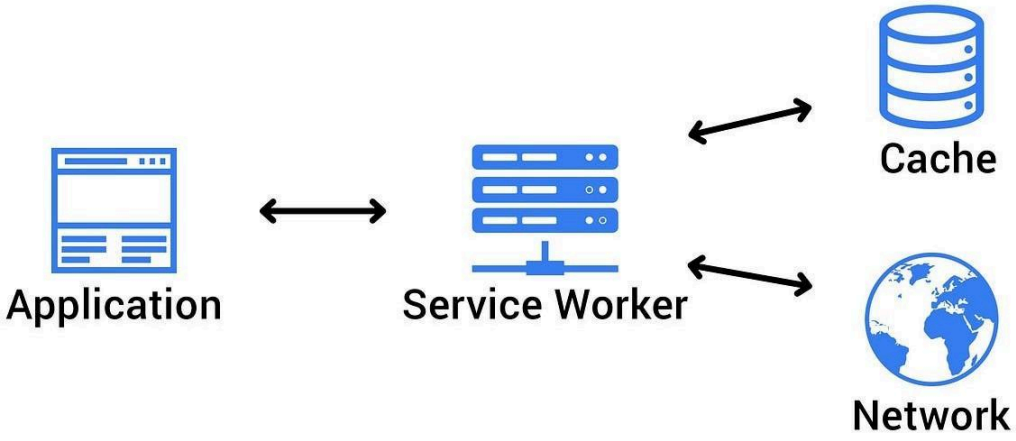
A service worker is a background script in a web browser that functions independently of user interaction. It serves as a customizable network intermediary, enabling control over how network requests from a web page are managed. Service workers facilitate various functionalities like monitoring network traffic, administering push notifications, and building "offline-first" web applications using the Cache API.

Key Points on Service Workers:
- Service workers act as network intermediaries, providing control over network requests and responses.
- They exclusively operate over HTTPS to ensure security measures.
- Service workers undergo a life cycle comprising registration, installation, and activation stages.
- They can cache resources for offline usage, handle push notifications, and execute background sync operations.

What Can be Achieved with Service Workers:
- Network Traffic Control: Manipulate network requests and responses, store resources for offline access, and manage offline content.
- Resource Caching: Utilize the Cache API to locally store resources for offline accessibility.
- Push Notification Management: Handle push notifications and present messages to users.
- Background Sync: Continue operations even when the internet connection is disrupted.

Limitations of Service Workers:
- Window Access Restriction: Service workers lack direct access to the Document Object Model (DOM) but can communicate with the window using postMessage.
- Port 80 Operation Prohibition: Service workers necessitate HTTPS and cannot function on Port 80.

Service Worker Life Cycle:
- Registration: Incorporate service worker registration in the primary JavaScript code to inform the browser of its location, thereby initiating the installation process.
- Installation: During installation, the service worker caches essential resources for offline accessibility and manages any pre-caching tasks.
- Activation: Following installation, the service worker enters the activation phase, where it assumes control over pages within its scope. It also cleans up outdated caches and prepares to handle network requests.



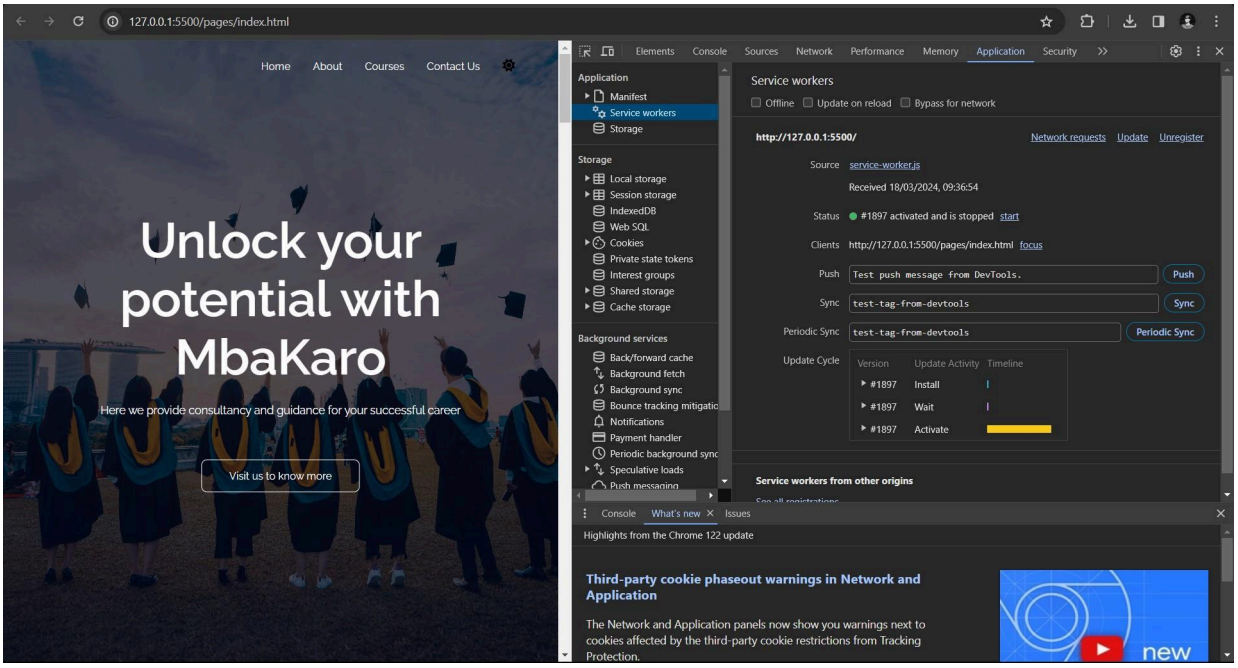
Code Example:

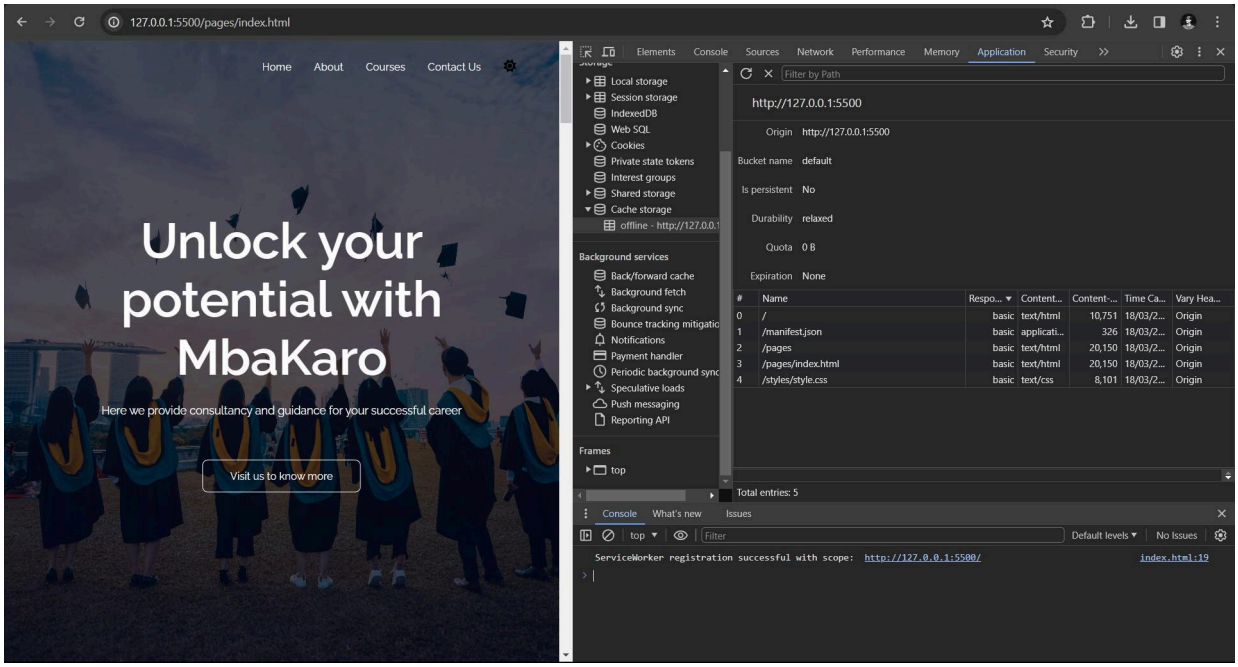
service-worker.js:
javascript

```
self.addEventListener("install", function (event) { event.waitUntil(preLoad());
});

var filesToCache = [ '/',
'/menu', '/contactUs', '/offline.html',
];

var preLoad = function () {
return caches.open("offline").then(function (cache) { return cache.addAll(filesToCache);
});
};

self.addEventListener("fetch", function (event) {
event.respondWith(checkResponse(event.request).catch(function () {
return returnFromCache(event.request);
}));
event.waitUntil(addToCache(event.request));
```

```
});

var checkResponse = function (request) { return new Promise(function (fulfill, reject) {
fetch(request).then(function (response) { if (response.status !== 404) {
fulfill(response);
} else {
reject();
                                                    }
                                                    }, reject);
});
};


var addToCache = function (request) {
return caches.open("offline").then(function (cache) { return fetch(request).then(function (response)
{
return cache.put(request, response);
});
});
};

var returnFromCache = function (request) {
return caches.open("offline").then(function (cache) { return cache.match(request).then(function
(matching) {
if (!matching || matching.status == 404) { return cache.match("offline.html");
} else {
return matching;
                                                    }
                                                    });
});
};
```

Name: Aaqueeb Pinjari Roll No.: 56 Class: D15B





Conclusion:

Service workers are pivotal in elevating web applications by facilitating functionalities like offline access, push notifications, and background sync. Familiarity with their life cycle and capabilities is fundamental for crafting resilient and effective Progressive Web Apps.