

Experiment No 07

Aim: create a simple hello world PWA

Code:

index.html

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>Hello World</title>
  <link rel="manifest" href="/manifest.json">
  <link rel="stylesheet" href="/css/style.css">
  <link rel="icon" href="/favicon.ico" type="image/x-icon" />
  <link rel="apple-touch-icon" href="/images/hello-icon-152.png">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta name="theme-color" content="white"/>
  <meta name="apple-mobile-web-app-capable" content="yes">
  <meta name="apple-mobile-web-app-status-bar-style" content="black">
  <meta name="apple-mobile-web-app-title" content="Hello World">
  <meta name="msapplication-TileImage" content="images/hello-icon-144.png">
  <meta name="msapplication-TileColor" content="#FFFFFF">
</head>
<body class="fullscreen">
  <div class="container">
    <h1 class="title">Hello World!</h1>
  </div>
  <script src="/js/main.js"></script>
</body>
</html>
```

manifest.json

```
{
  "name": "Hello World",
  "short_name": "Hello",
  "icons": [{
    "src": "images/hello-icon-128.png",
    "sizes": "128x128",
    "type": "image/png"
  }, {
    "src": "images/hello-icon-144.png",
    "sizes": "144x144",
    "type": "image/png"
  }, {
    "src": "images/hello-icon-152.png",
    "sizes": "152x152",
    "type": "image/png"
  }, {
```

```

    "src": "images/hello-icon-192.png",
    "sizes": "192x192",
    "type": "image/png"
  }, {
    "src": "images/hello-icon-256.png",
    "sizes": "256x256",
    "type": "image/png"
  }, {
    "src": "images/hello-icon-512.png",
    "sizes": "512x512",
    "type": "image/png"
  }
],
"lang": "en-US",
"start_url": "/index.html",
"display": "standalone",
"background_color": "white",
"theme_color": "white"
}

```

sw.js

```

var cacheName = 'hello-pwa';
var filesToCache = [
  './',
  './index.html',
  './css/style.css',
  './js/main.js'
];

```

```

/* Start the service worker and cache all of the app's content */
self.addEventListener('install', function(e) {
  e.waitUntil(
    caches.open(cacheName).then(function(cache) {
      return cache.addAll(filesToCache);
    })
  );
  self.skipWaiting();
});

```

```

/* Serve cached content when offline */
self.addEventListener('fetch', function(e) {
  e.respondWith(
    caches.match(e.request).then(function(response) {
      return response || fetch(e.request);
    })
  );
});

```

css folder

style.css

```
body {  
  font-family: sans-serif;  
}  
  
/* Make content area fill the entire browser window */  
html,  
.fullscreen {  
  display: flex;  
  height: 100%;  
  margin: 0;  
  padding: 0;  
  width: 100%;  
}  
  
/* Center the content in the browser window */  
.container {  
  margin: auto;  
  text-align: center;  
}  
  
.title {  
  font-size: 3rem;  
}
```

js folder

main.js

```
window.onload = () => {  
  'use strict';  
  
  if ('serviceWorker' in navigator) {  
    navigator.serviceWorker  
      .register('./sw.js');  
  }  
}
```

images folder

Output:

