# Web Progressive App

Ridj Bissessur

ridj.uk

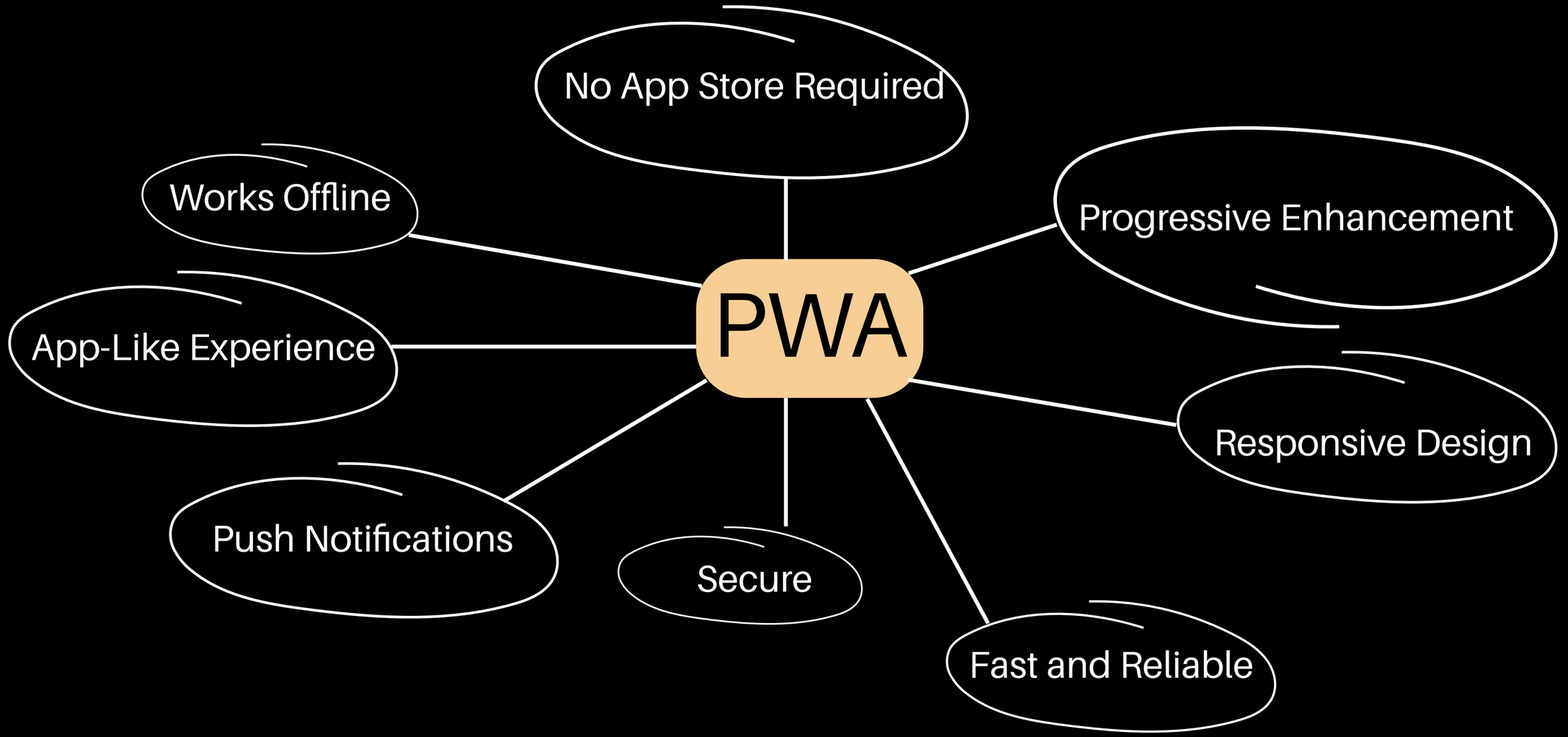
# What is PWA?

1

A **Progressive Web App** (PWA) is a type of web application that combines the best features of both **websites and native mobile apps**. It is designed to provide a **fast, reliable, and engaging user experience** across different devices and platforms.

# Features of PWA

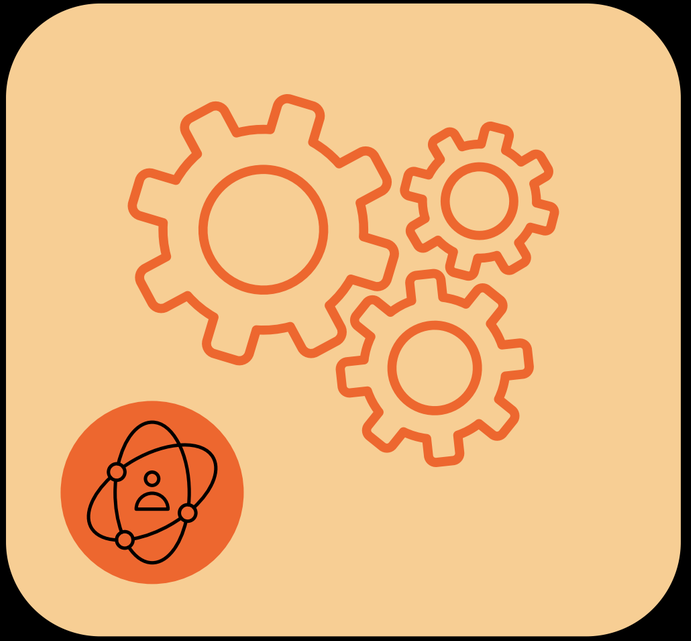
2



# From Websites to PWA

3

1. Runs on HTTPs
2. Has a manifest.json file
3. Has a service worker
4. Is responsive and smooth



# Manifest.json

4

**The manifest.json is a JSON file that provides metadata about a Progressive Web App (PWA). It tells the browser how the app should behave when installed on a user's device, helping create a native app-like experience.**

**Why is manifest.json Important?**

**Allows users to install the PWA on their home screen.**
**Defines the app's appearance when launched (icons, colors, display mode).**
**Enhances the user experience by making the app feel more like a native app.**

# Example of a manifest.json

5

```
{
 "name": "My PWA",
 "short_name": "PWA",
 "description": "An awesome Progressive Web App",
 "start_url": "/index.html",
 "display": "standalone",
 "background_color": "#ffffff",
 "theme_color": "#000000",
 "icons": [
  {
   "src": "/icons/icon-192x192.png",
   "sizes": "192x192",
   "type": "image/png"
  },
  {
   "src": "/icons/icon-512x512.png",
   "sizes": "512x512",
   "type": "image/png"
  }
 ]
}
```

# Service Worker

**A Service Worker is a background script that runs separately from the main web page, enabling features like offline support, caching, push notifications, and background synchronization in a Progressive Web App (PWA).**

# Key Features of a Service Worker

7

- **Offline Functionality** – Caches files so the app works without an internet connection.
- **Improved Performance** – Loads pages faster by serving cached content.
- **Push Notifications** – Receives and handles push messages from the server.
- **Background Sync** – Saves data when offline and syncs it when the internet is available.
- **Intercepts Network Requests** – Controls how requests are handled, improving efficiency.

# Service Worker script

8

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

**CACHE_NAME:** This defines the name of the cache storage. Every time we update files, we should change this name (e.g., 'my-pwa-cache-v2').

**urlsToCache:** This is an array of files that will be stored in the cache for offline use.

# Service Worker script



```
self.addEventListener('install', event => {
  event.waitUntil(
    caches.open(CACHE_NAME)
      .then(cache => {
        console.log('Caching app shell');
        return cache.addAll(urlsToCache);
      })
  );
});
```

**install event:** Triggers when the service worker is installed.

**event.waitUntil():** Ensures that caching completes before the service worker finishes installation.

**caches.open(CACHE_NAME):** Opens a cache storage with the given name.

**cache.addAll(urlsToCache):** Saves all the specified files into the cache.

# Service Worker script



```
self.addEventListener('activate', event => {
  event.waitUntil(
    caches.keys().then(cacheNames => {
      return Promise.all(
        cacheNames.map(cache => {
          if (cache !== CACHE_NAME) {
            console.log('Deleting old cache:', cache);
            return caches.delete(cache);
          }
        })
      );
    })
  );
});
;
```

**activate event:** Fires when a new service worker takes control.

**caches.keys():** Retrieves all cache names stored in the browser.

**Loop through cache names:** If a cache name does not match CACHE_NAME, it gets deleted.

**Why?** This prevents old versions of files from being stored and ensures the app stays updated..

# Service Worker script



```
self.addEventListener('fetch', event => {
  event.respondWith(
    caches.match(event.request).then(response
=> {
      return response || fetch(event.request);
    })
  );
});
```

**fetch event:** Listens to all network requests made by the app.

**caches.match(event.request):**
Checks if the requested file is in the cache.
**return response || fetch**

**(event.request):**
If the file is found in the cache, it is returned.
If not, the request is fetched from the internet.

**Why?** This allows offline functionality while ensuring the latest files are used when online.

# Best Practice for future Web Designer 12

**OBJECTIVE**

- Mobile first approach
- Use Light room to test your website
- Ensure properly sized images

# Thank you for your attention



*learn more at:*

**MDN**
*https://developer.mozilla.org/en-US/docs/Web/Progressive_web_apps/Guides/Making_PWAs_installable*

**YouTube**
*https://www.youtube.com/watch?v=1WWweyBaWZk*