

To-Do List PWA (Progressive Web App)

Description: Create a simple yet functional To-Do List application using JavaScript that works seamlessly both online and offline. This PWA will allow users to manage their daily tasks efficiently by adding, editing, and deleting tasks. It will save the tasks locally on the user's device, ensuring that they remain accessible even when offline.

Features:

- **Add, Edit, and Delete Tasks:** Users can easily manage their tasks with intuitive controls.
- **Offline Capability:** Tasks are stored in the browser's local storage or IndexedDB, enabling full offline functionality.
- **Service Worker:** A Service Worker is implemented to cache essential files, making the app accessible even when the internet connection is unavailable.
- **Responsive Design:** The app is fully responsive, ensuring a smooth experience across various devices (desktop, tablet, and mobile).
- **Installation as PWA:** The app can be installed on a mobile device directly from the browser, similar to a native app.

Technologies Used:

- HTML, CSS, JavaScript
- Service Worker for caching
- IndexedDB or localStorage for offline data storage
- Web App Manifest for PWA installation

Relevant course:

<https://youtube.com/playlist?list=PL4cUxeGkcC9gTxqJBcDmoi5Q2pzDusSL7&si=3XIDTQaJ9ZP5QBv2>

2. Spotify-like PWA with JavaScript

Description: Develop a music streaming application inspired by Spotify using JavaScript as a Progressive Web App (PWA). This app will enable users to browse through music, create playlists, and listen to songs. It will have a robust caching system that stores previously played songs and playlists offline, allowing the app to function when the user is not connected to the internet. The design will closely mimic popular music streaming apps, with a focus on user experience and performance.

Features:

- **Browse and Play Songs:** Users can search for songs and play them within the app.
- **Create Playlists:** Users can organize their favorite tracks into playlists.
- **Offline Playback:** Songs and playlists are cached locally for playback even without an internet connection.
- **Push Notifications:** Users receive notifications about new songs or albums (optional).
- **Installable PWA:** The app can be installed on mobile devices for a more native-like experience.

Technologies Used:

- HTML, CSS, JavaScript
- Service Worker for caching and offline playback
- Web Storage (localStorage/IndexedDB) for storing songs and playlists
- Web App Manifest for enabling PWA installation

These descriptions follow a similar structure as your previous project files, giving clear information about functionality, technologies, and features.

Similar project:

https://youtube.com/playlist?list=PLHeRUSseYHRmdUqrWc3Rju6F1AKIMPdjCc&si=fzJsFrwu-3af_b0