November 13, 2024

Keith Cayatoc

B.I.T Student

Nelson Marlborough Institute of Technology

WEB701 Assessment 2 Final milestone

Tutor: Todd Cochrane

## Project Overview

**Yen-Yang App** includes a full-stack application with features such as user authentication, product registration, transaction history, and a Progressive Web App (PWA) implementation. Each component is designed to provide a seamless and interactive experience.

Progressive Web Apps (PWAs) represent a modern web technology that bridges the gap between web and native applications. By utilizing service workers and a web app manifest, PWAs provide offline capabilities, faster load times, and the option for users to "install" the app on their devices. According to research (Google Developers, 2024; MDN Web Docs, 2024), PWAs have shown increased engagement due to these features

## Implemented Functionalities

##### User Authentication

Users can register and login using their credentials. For secure user sessions, JWT tokens are generated, and the token is stored in a local storage to persist the session.

##### Profile Management

Users can view and edit their profile details, this information includes the users name, email, and token balance. Upon logging in, users get a notification about their recent transactions or updates related to their account.

##### Product Registration and Listing

Users can register a product or service with details such as product name, description, and price (in tokens). The registered products are displayed in a list format, allowing other users to view available items.

##### Transaction and Token System

Users can acquire products/services that will then deduct the price from their token balance and add it to the sellers account. Transaction history is recorded and displayed for users to track their purchases, with transaction details including type, product name, amount, and date.

##### Progressive Web App (PWA)

PWA functionality was added to the project which enables offline access and app-like behaviour on supported devices. The setup includes a service-worker.js file to cache assets, a manifest.json file for PWA metadata, and icons for different screen sizes.

## Progressive Web Application (PWA) Implementation for Emerging Web Technology

The service worker caches important files for offline functionality. When users access the application offline, cached content is displayed to ensure continuous access. It also handles install, activate and fetch events to manage the cache and serve resources when offline.

The manifest.json contains information like the app name, theme colour, background colour, start URL, and icons. This file provides metadata, enabling the app to be added to a home screen on mobile devices.

A screen shot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

References

**Google Developers**. (n.d.). *Progressive Web Apps*. Retrieved from <https://developers.google.com/web/progressive-web-apps>

Mozilla Developer Network (MDN). (n.d.). Service Workers: An Introduction. Retrieved from <https://developer.mozilla.org/en-US/docs/Web/API/Service_Worker_API/Using_Service_Workers>

Google Developers. (n.d.). The Web App Manifest. Retrieved from <https://developers.google.com/web/fundamentals/web-app-manifest/>

Mozilla Developer Network (MDN). (n.d.). Using the Fetch API. Retrieved from <https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API/Using_Fetch>

**NPM Documentation**. (n.d.). *JSON Web Token (JWT)*. Retrieved from <https://www.npmjs.com/package/jsonwebtoken>

HubSpot. (2023, June 19). Web development trends to watch in 2023. HubSpot Blog. <https://blog.hubspot.com/website/web-development-trends>