Sprawozdanie z tematu IV – Hostowanie aplikacji

Przedmiot: Systemy urządzeń mobilnych

Autor: Jakub Opiełka

Kierunek: Informatyka dla inżynierów  
Semestr: I

1. Wstęp

Progressive Web App (PWA) to nowoczesne podejście do tworzenia aplikacji webowych, które łączy cechy tradycyjnych stron internetowych z funkcjonalnościami natywnych aplikacji mobilnych. Dzięki takim technologiom jak Service Worker czy IndexedDB, możliwe jest m.in. działanie offline, szybkie ładowanie czy instalacja na urządzeniu użytkownika.

Celem niniejszego projektu było stworzenie pełnoprawnej aplikacji PWA w czystym JavaScript (bez użycia frameworków), spełniającej określone wymagania funkcjonalne i techniczne. Projekt zakładał implementację własnego service workera, konfigurację manifestu, obsługę trybu offline oraz wdrożenie aplikacji na darmowym serwerze z obsługą HTTPS.

Efektem pracy jest aplikacja WeatherNotes, która umożliwia użytkownikowi sprawdzenie aktualnej pogody w wybranym mieście oraz tworzenie i przeglądanie notatek zapisywanych lokalnie — także w trybie offline.

2. Opis funkcjonalności aplikacji.

Aplikacja WeatherNotes łączy dwie praktyczne funkcje: sprawdzanie aktualnej pogody w wybranym mieście oraz zapisywanie własnych notatek, np. dotyczących planów na dany dzień. Użytkownik może:

* wyszukać aktualną pogodę na podstawie wpisanego miasta (integracja z API OpenWeatherMap),
* dodać notatkę w prostym formularzu (zapis lokalny w IndexedDB),
* przeglądać i usuwać zapisane notatki,
* korzystać z aplikacji offline – ostatnio pobrana pogoda i zapisane notatki są dostępne bez internetu.

3. Opis techniczny krok po kroku.

3.1 Tworzymy strukturę projektu w edytorze.

A screen shot of a computer

AI-generated content may be incorrect.

3.2 Logujemy się do serwisu z darmowym API pogodowym – OpenWeather i generujemy klucz do API

A screenshot of a computer

AI-generated content may be incorrect.

3.2 Tworzymy pierwszy panel w pliku index.html oraz łączymy go z logiką w pliku main.js, która uderza od razu do tego API.

A screenshot of a computer program

AI-generated content may be incorrect.A screenshot of a computer program

AI-generated content may be incorrect.

Mamy pierwszy suckes, dane zostały zwrócone:  
  
A screenshot of a computer

AI-generated content may be incorrect.

3.3 Implementujemy formularz a więc pliki add-note.html oraz add-note.js, warto zauważyć, że wprowadzamy tutaj indexedDB

A screenshot of a computer screen

AI-generated content may be incorrect.

Mamy kolejny sukces, po wprowadzeniu notatki dostajemy komunikat o zapisie:  
A screenshot of a computer

AI-generated content may be incorrect.

Po przejściu do Application – IndexedDB, widzimy zapisane dane:  
A screenshot of a computer

AI-generated content may be incorrect.

3.4 Implementujemy panel note, co istotne, dane pobieramy już z naszej indexedDB, te dane które były wcześniej przedstawione na obrazku w narzędziach przeglądarki.

A screenshot of a computer screen

AI-generated content may be incorrect.

Panel rzeczywiście wyświetla naszą notatkę, działa również routing do formualarza dodawania:  
  
A screenshot of a computer

AI-generated content may be incorrect.

3.5 Implementujemy manifest.json oraz dodajemy ikony. Plik manifest.json zawiera metadane aplikacji PWA, które pozwalają przeglądarce rozpoznać ją jako aplikację „instalowalną”. Dzięki temu użytkownik może dodać aplikację do ekranu głównego tak jak natywną aplikację mobilną. W pliku zdefiniowano m.in. nazwę, kolory, ikonki w wymaganych rozdzielczościach oraz stronę startową. Manifest jest podłączony do każdej strony aplikacji za pomocą znacznika <link rel="manifest"> w sekcji <head>.

A screenshot of a computer program

AI-generated content may be incorrect.

3.6 Pora na service worker, a więc mózg naszej aplikacji PWA. W service-worker.js zastosowano strategię cache-first dla plików statycznych (HTML, CSS, JS, ikony), aby aplikacja mogła działać offline. Dla danych z zewnętrznego API (OpenWeather) użyto strategii network-first, by zawsze próbować pobrać najnowsze dane. Pliki są cache’owane podczas instalacji, a stare cache'y usuwane przy aktywacji, co zapobiega błędom przy aktualizacji aplikacji.  
  
Dodatkowo notatki użytkownika są przechowywane lokalnie w przeglądarce za pomocą IndexedDB i nie są przesyłane do żadnego serwera, co zapewnia prywatność oraz dostępność w trybie offline.

A screenshot of a computer program

AI-generated content may be incorrect.

Niestety tym razem nie mamy sukecsu, service worker nie jest wykrywany:  
A screenshot of a computer

AI-generated content may be incorrect.

Po dodaniu skrptu rejestrującego dodatkwo manualnie service worker, niestety nadal bez sukcesu:  
  
A screen shot of a computer code

AI-generated content may be incorrect.

Być może to kwestia uruchamiania przez live server, dlatego przejdziemy do hostowania aplikacji na netlify, być może tam service worker zadziala.

3.7 Deployment na netlify.

Pakujemy pliki naszego projektu w plik .zip i wrzucamy do Netlify:

A screenshot of a computer

AI-generated content may be incorrect.

Wygląda na to, że udało się zhostować:

A screenshot of a computer

AI-generated content may be incorrect.

Deployment pomógł, service worker jest teraz widziany:

A screenshot of a computer

AI-generated content may be incorrect.

3.8 Aplikacja ma już zaimplementowane większość rzeczy, ale aby aplikacja spełniała jak najlepiej standardy PWA, dodajemy style, mając na uwadze również responsywność.   
  
A screen shot of a computer program

AI-generated content may be incorrect.

W rezultacie, aplikacja zaczęła wyglądać o wiele lepiej:  
  
Sprawdzanie pogody:  
  
A screenshot of a computer

AI-generated content may be incorrect.

Lista notatek z ekranu telefonu:

A screenshot of a computer

AI-generated content may be incorrect.

Widok formularza nowej notatki:  
  
A screenshot of a computer

AI-generated content may be incorrect.

Aby aplikacja miała możliwość nawigacji między panelem pogodowym i notatkowym, dodajemy dodatkową nawigację:  
  
Dodajemy sekcję nawigacji na górze każdego widoku, czyli na początku sekcji body:  
  
A screenshot of a computer program

AI-generated content may be incorrect.

W efekcie mamy działającą nawigację: A screenshot of a computer

AI-generated content may be incorrect.

4. Trudności i ich rozwiązania

4.1. Service Worker nie rejestrował się lokalnie

Początkowo aplikacja nie rejestrowała poprawnie pliku service-worker.js podczas pracy z lokalnym serwerem Live Server. W konsoli przeglądarki nie pojawiały się żadne logi rejestracji, a testowa wizyta pod adresem /service-worker.js zwracała komunikat Cannot GET. Problem wynikał z faktu, że Live Server uruchamiał aplikację z podkatalogu (/weather-notes/), a ścieżki w Service Workerze były zdefiniowane względem root (/). Dodatkowo, Live Server nie obsługuje Service Workerów w sposób w pełni zgodny z produkcyjnym środowiskiem. Problem został rozwiązany poprzez wdrożenie aplikacji na Netlify, które oferuje HTTPS i pełne wsparcie dla PWA. Nauczyło mnie to, że testowanie PWA wymaga odpowiedniego środowiska, najlepiej zbliżonego do rzeczywistego hostingu.

4.2. Brak widoczności komunikatów z console.log

Podczas testowania rejestracji Service Workera długo nie pojawiały się żadne komunikaty w konsoli przeglądarki. Początkowo wydawało się, że kod navigator.serviceWorker.register(...) się nie wykonuje. Po dokładnej analizie okazało się, że plik sw-register.js, który zawierał kod rejestracji, nie został poprawnie załadowany w dokumencie HTML — brakowało znacznika <script src="js/sw-register.js"> lub wskazywał on na błędną ścieżkę. Po poprawieniu ścieżki do pliku wszystko zaczęło działać prawidłowo. Nauczyło mnie to, że nawet drobna literówka w ścieżce może całkowicie zablokować wykonanie istotnej logiki aplikacji.

4.3. IndexedDB nie inicjalizował się poprawnie

Podczas tworzenia formularza do zapisu notatek pojawiał się problem braku połączenia z bazą danych IndexedDB. Po analizie okazało się, że kod uruchamiający transakcję do zapisu był wykonywany zanim baza danych została zainicjalizowana przez indexedDB.open(...). Problem został rozwiązany poprzez przeniesienie logiki zapisu wewnątrz request.onsuccess, co zagwarantowało, że baza danych była w pełni gotowa przed wykonaniem dalszych operacji. Nauczyło mnie to, że praca z IndexedDB wymaga asynchronicznego podejścia i cierpliwego czekania na inicjalizację.

5. Podsumowanie:

Aplikacja działa zgodnie z założeniami — poprawnie wyświetla dane pogodowe z zewnętrznego API, umożliwia dodawanie i przeglądanie notatek offline dzięki IndexedDB oraz spełnia wymagania PWA: posiada manifest, ikony, service workera i działa po zainstalowaniu jak natywna aplikacja.

Wymagałaby dopracowania pod kątem UX, np. lepszych komunikatów dla użytkownika czy informacji o stanie offline/online. Dodatkowo można byłoby rozważyć integrację z backendem lub opcję eksportu notatek.

W trakcie pracy nauczyłem się praktycznego wykorzystania Service Workera i IndexedDB, zrozumiałem ograniczenia środowisk developerskich takich jak Live Server oraz dowiedziałem się, jak prawidłowo przygotować i wdrożyć aplikację PWA na produkcyjny hosting.

6. Załączniki  
  
LINK do netlify: <https://6827b3214e15d29565454e07--velvety-blancmange-4811f7.netlify.app/>

Zrzuty online:   
  
A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

Zrzuty offline:

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.