COM3504 Intelligent Web - PWA Assignment Documentation

Introduction

The first stage of the PWA Assignment entails a working solution PWA app which can create, and store information related to user accounts, events and stories. The data related to these is cached and stored locally in the browser and on IndexedDB in the case of JSON objects. Future implementations for stage 2 include a working MongoDB database and a search function for users to be able to search for events.

Layout design

The layout of the PWA was designed using StartBootstrap's clean blog template. This template was chosen for its relatively simple layout and blog-like compatibility with the specification of the assignment.

Project Structure

To fulfil the assignment requirements, it was determined that the following pages would be required for the PWA:

- **Home Page**: a user may view a list of events on the home page and click on an event name to view the event
- **User registration**: a user may create a new account by providing their name, desired username and desired password
- **User login**: a user may login to their existing account
- **View event**: a user may view a list of stories associated with a specific event, and click on a link to create a new story of their own to be associated with said event
- **Event creation**: a user may create a new event if it does not already exist
- **Story creation**: a user may create a story including a picture at an event and select the event which the story would belong to (or create a new event if it does not exist)
- **Event search**: a user may search for an event, either through a list or on a map.

Functionalities implemented

A substantial number of the project requirements have managed to be met without the use of the MongoDB database, namely requirements related to creation of stories, events and use of a we worker. Functionalities included in the first stage of the assignment are:

- Storage of user accounts, events and stories as JSON objects and use of said JSON objects for registration and login. Storage is done with IndexedDB to enable implementation of **cache before network** strategy in the final product
- Service worker for PWA for caching page layout templates and the above-mentioned data.
- Registration and login functionalities for user accounts
- Creation and display of events and stories
- Implementation of **WebRTC** on story creation page to enable webcam use and snapshot creation so that a photo can be attached to a story
- Implementation of Leaflet maps to show location of events, including when offline (caching of maps is included in the service worker)

Insertion & Searching of data via forms

- **Challenges:** ensuring correct matching of object attributes so that relevant data is found, as errors thrown may not point this out clearly
- **Solution:** implemented 2 forms for creating user stories/events and user registration/login, to make the structure more modular
- **Requirements**: data is successfully passed from the forms to the JSON objects and IndexedDB. Events, stories and user accounts can be created with their relevant data listed in the project requirements
- **Limitations**: A search form is not yet included in the current version of the PWA app

Caching of the app template with web worker

- **Challenges:** The bootstrap template contained a lot of logo files, and it is required to list every single file for caching. This would have led to a very long list of cache files in the web worker. In addition, SSL certificate generator did not work for the web worker.
- **Solution:** Did not list every single file from bootstrap template for caching as not all were needed. An alternative SSL certificate generator was used for the web worker to enable HTTPS.
- **Requirements**: Web worker successfully caches the needed files for offline use. All pages within the PWA correctly function offline, except for the pages which display the map.
- **Limitations**: Maps currently do not work offline; an initial connection is required before the tiles of the map are cached.

Caching data using IndexedDB

- **Challenges:** Debugging when a JSON object is not correctly added to the IndexedDB, as this can have several causes; the Ajax query, a possible browser issue, etc. Identifying the source of such an error can be tricky.
- **Solution:** Started out with simpler JSON objects with fewer attributes and built up the solution on-the-go. When debugging, “elimination of possible causes” approach was used.
- **Requirements**: IndexedDB successfully stores JSON objects for user accounts, events and stories. This will later enable the caching strategy “cache before network” to be used with MongoDB and for searching locally when a network is unavailable.
- **Limitations**: when the search function is added in the next stage of the project, a local search through IndexedDB may be out of date and data matching may not be as sophisticated.

Future tasks

The remaining future tasks for stage 2 are aimed at addressing the remaining requirements, namely related to event searching and use of extra packages. Objectives which are planned for stage 2 include:

- Implementation of MongoDB databases: 2 database files are planned for use; one will be used for user accounts due to the need to encrypt user passwords in the final product, and the other for the remaining data
- Implementation of the passport package to enable user registrations/logins with MongoDB
- Implementation of the event search function to allow users to search for events, either by list or from a map
- Extension of webcam function to make use of the socket.io package
- Modification of the styling to improve overall appearance of the PWA app
- Implementation of access control for creation of events and stories; a user account currently isn't required to perform any of the mentioned tasks.