

Project One: Progressive Web App

Grade: 100 points

Objective:

Design and develop a dynamic, data-driven progressive web app through an iterative process.

Overview:

Create “three” pages for an app that shows the top 10 places to ____ in Rochester.

Premise:

Conceive of a progressive web app that allows users to discover different activities in multiple locations. Start your development from a point where a user has either searched for or navigated to Rochester as a location and was looking for your chosen activity.

Requirements:

- **Main page**—the listing of the locations
- **Details view**—navigation from each of the individual listings that presents additional data
- **Favorites list**—ability to favorite locations and view them

Process:

Create a JSON file that contains an assortment of data for each location that will be displayed in the main listing and detail view. This file will be “fetched” as if it were an API.

Navigation structure of the app is up to you. You can create the appearance of other features for the app, but you only need to build the three primary interactions of viewing a detail from the main listing, favoriting a location, and viewing this self-created “list”.

Use a design process! Start with sketching to generate as many ideas as you can and move the most promising into wireframes to help determine the structure and organization. Prototype it out further if you want to test your navigation structure and interactions.

Using HTML, CSS, and JavaScript, develop the ability to fetch your JSON data in a ServiceWorker and notify the PWA to display it into some type of listing structure. This should happen in the background using the Background Synchronization API and stored in an IndexedDB. This could be displayed as just a list, or cards, or a grid for example—up to you. You can either use the `postMessage` or Broadcast Channel API to communicate between the Service Worker and the PWA.

Design the app to be responsive including images but should look good on a mobile-sized viewport.

You'll can use either `localStorage` or `IndexedDB` to keep the favorited items as persistent. This can be loaded from and saved to from either the main PWA or in the ServiceWorker – your choice.

Submission:

Upload development files onto Solace
Provide the link.
Submit documentation

Evaluation:

- 15 points - Documentation of your process is beneficial. Provide a single document that presents this process in a clear, and concise way. Someone viewing it should easily make the connection of where you started, what you arrived on and how you got there.
- 50 points - Basic usability is expected including registering the ServiceWorker. Build your functionality and install it on the server. Use coding standards such as comments, indentation, and appropriate function and variable naming. Make sure all files validate.
- 10 points – Valid manifest file with icons so the PWA will be installable.
- 25 points - Visual presentation creates a more enjoyable experience. Be particular with the content's layout, your choice and use of type, and the application of a color theme.