**# Unit 19 PWA Homework: Text Editor**

**## Your Task**

As you have progressed through this course, you have put together a number of impressive projects that you can show off to potential employers. This project is no exception; in fact, it features some of the most impressive expressions of the concepts you have learned so far.

Your task is to build a text editor that runs in the browser. The app will be a single-page application that meets the PWA criteria. Additionally, it will feature a number of data persistence techniques that serve as redundancy in case one of the options is not supported by the browser. The application will also function offline.

To build this text editor, you will start with an existing application and implement methods for getting and storing data to an IndexedDB database. You will use a package called `idb`, which is a lightweight wrapper around the IndexedDB API. It features a number of methods that are useful for storing and retrieving data, and is used by companies like Google and Mozilla.

You will deploy this full-stack application to Heroku using the [Heroku Deployment Guide on The Full-Stack Blog](https://coding-boot-camp.github.io/full-stack/heroku/heroku-deployment-guide).

**## User Story**

```md

AS A developer

I WANT to create notes or code snippets with or without an internet connection

SO THAT I can reliably retrieve them for later use

```

**## Acceptance Criteria**

```md

GIVEN a text editor web application

WHEN I open my application in my editor

THEN I should see a client server folder structure

WHEN I run `npm run start` from the root directory

THEN I find that my application should start up the backend and serve the client

WHEN I run the text editor application from my terminal

THEN I find that my JavaScript files have been bundled using webpack

WHEN I run my webpack plugins

THEN I find that I have a generated HTML file, service worker, and a manifest file

WHEN I use next-gen JavaScript in my application

THEN I find that the text editor still functions in the browser without errors

WHEN I open the text editor

THEN I find that IndexedDB has immediately created a database storage

WHEN I enter content and subsequently click off of the DOM window

THEN I find that the content in the text editor has been saved with IndexedDB

WHEN I reopen the text editor after closing it

THEN I find that the content in the text editor has been retrieved from our IndexedDB

WHEN I click on the Install button

THEN I download my web application as an icon on my desktop

WHEN I load my web application

THEN I should have a registered service worker using workbox

WHEN I register a service worker

THEN I should have my static assets pre cached upon loading along with subsequent pages and static assets

WHEN I deploy to Heroku

THEN I should have proper build scripts for a webpack application

```

**## Mock-Up**

The following animation demonstrates the application functionality:

![Demonstration of the finished Unit 19 Homework being used in the browser and then installed.](./Assets/00-demo.gif)

The following image shows the application's `manifest.json` file:

![Demonstration of the finished Unit 19 Homework with a manifest file in the browser.](./Assets/01-manifest.png)

The following image shows the application's registered service worker:

![Demonstration of the finished Unit 19 Homework with a registered service worker in the browser.](./Assets/02-service-worker.png)

The following image shows the application's IndexedDB storage:

![Demonstration of the finished Unit 19 Homework with a IndexedDB storage named 'jate' in the browser.](./Assets/03-idb-storage.png)

**## Grading Requirements**

This homework is graded based on the following criteria:

**### Technical Acceptance Criteria: 40%**

\* Satisfies all of the above acceptance criteria plus the following:

\* Uses IndexedDB to create an object store and includes both GET and PUT methods

\* The application works without an internet connection

\* Automatically saves content inside the text editor when the DOM window is unfocused

\* Bundled with webpack

\* Create a service worker with workbox that Caches static assets

\* The application should use babel in order to use async / await

\* Application must have a generated `manifest.json` using the `WebpackPwaManifest` plug-in

\* Can be installed as a Progressive Web Application

**### Deployment: 32%**

\* Application deployed to Heroku at live URL with build scripts

\* Application loads with no errors

\* Application GitHub URL submitted

\* GitHub repo contains application code

**### Application Quality: 15%**

\* Application user experience is intuitive and easy to navigate

\* Application user interface style is clean and polished

\* Application resembles the mock-up functionality provided in the homework instructions

**### Repository Quality: 13%**

\* Repository has a unique name

\* Repository follows best practices for file structure and naming conventions

\* Repository follows best practices for class/id naming conventions, indentation, quality comments, etc.

\* Repository contains multiple descriptive commit messages

\* Repository contains quality README file with description, screenshot, and link to deployed application

**## Review**

You are required to submit the following for review:

\* The URL of the deployed application

\* The URL of the GitHub repository, with a unique name and a README describing the project

- - -

© 2022 Trilogy Education Services, LLC, a 2U, Inc. brand. Confidential and Proprietary. All Rights Reserved.