CMPS 350 Web Fundamentals
Lab 8 – Client-side JavaScript

Objective

In this Lab, you will **build a Census , a Todo and a Gallery apps that persists data offline.** You will use IndexedDB and Localbase library to get, add, update, and delete records in Indexed Database.

In the Lab, you will practice:

- Create and interact with the IndexedDB database using the Localbase library.
- Create and Delete Collections.
- Perform database CRUD operations
- Filter, Find, and Sort Documents

Overview

In this Lab, you will create two applications.

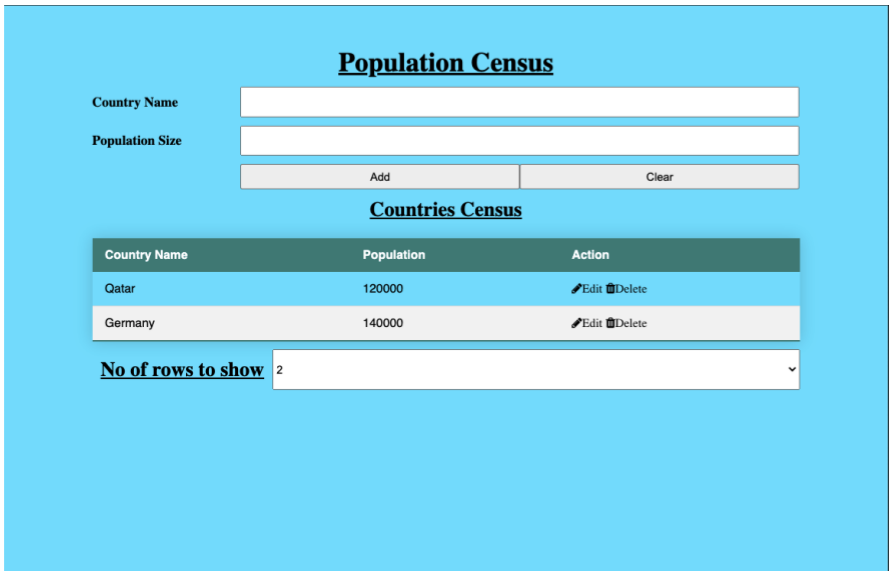
1. Country Census App
2. Todo List App
3. Local Image Gallery App

Project Setup

1. Sync "Lab8-Client JS Data Management" from the Lab GitHub repository and copy it to your repository.
2. The folder contains both the Todo and Census App HTML and CSS files. Your task will be to write the necessary JavaScript code that allows you to persist data.

PART A – Census App

Open the Census App on WebStorm and try to implement the following application.



Population Census

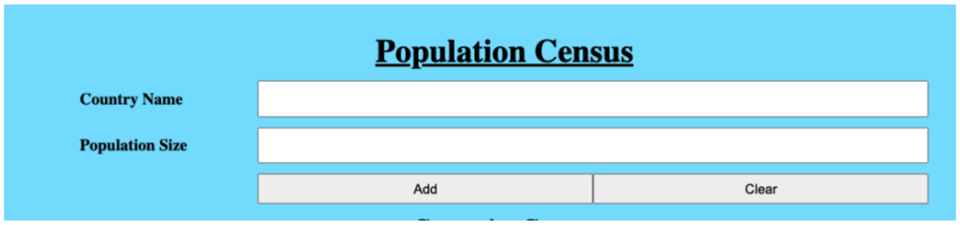
Country Name	
Population Size	

Countries Census

Country Name	Population	Action
Qatar	120000	Edit Delete
Germany	140000	Edit Delete

No of rows to show 2

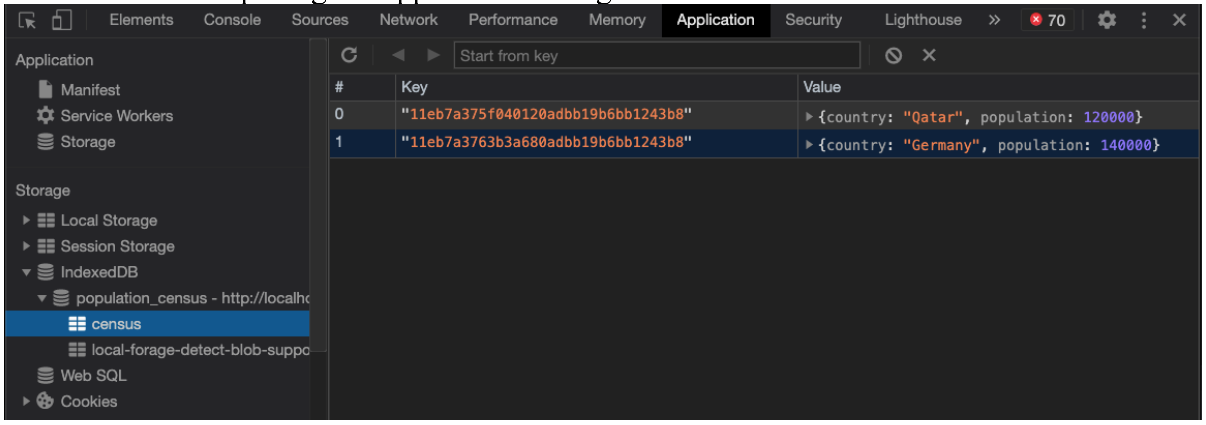
1. Add the following script tag to your code to get the Localbase library
   `<script src="https://unpkg.com/localbase/dist/localbase.min.js"></script>`
2. Create the census IndexedDB database inside the index.js file
   `const db = new Localbase('census.db');`
3. Implement the following functionalities
   a. When the users click on the **add button,** you should
      i. Call a function called `addCensus` that receives the form,
      ii. Convert the form into a JSON object,
      iii. Add the JSON object into a collection called '**census.**'



Population Census

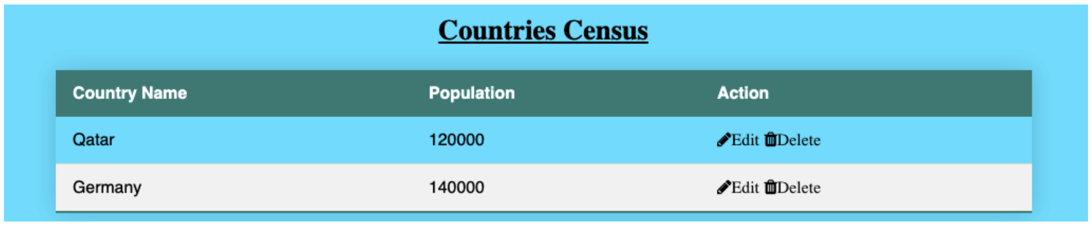
Country Name	
Population Size	

   b. Check if the census information is saved properly by going into your browser dev tools and opening the application/storage tab



#	Key	Value
0	"11eb7a375f040120adbb19b6bb1243b8"	{country: "Qatar", population: 120000}
1	"11eb7a3763b3a680adbb19b6bb1243b8"	{country: "Germany", population: 140000}

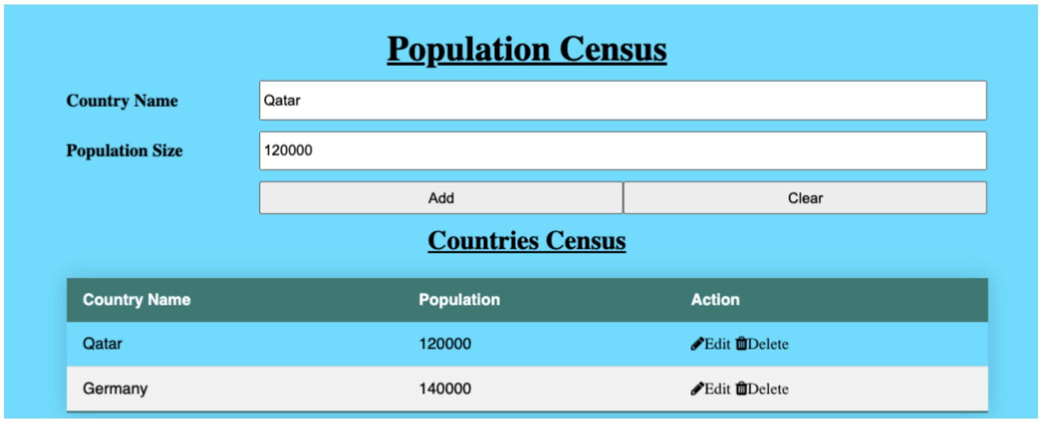
c. If the user clicks the clear button, make sure you clear the content of the two boxes.
d. As soon as the user opens the application, try to load all the census information in a table. You should generate the table data programmatically using the same techniques that you learnt last week.



Countries Census

Country Name	Population	Action
Qatar	120000	Edit Delete
Germany	140000	Edit Delete

e. Implement the edit functionality. When the user clicks on the edit icon, you should
   i. Call a function names `editCensus`
   ii. Load the country name and population into the input boxes
   iii. Update the census document that corresponds to the one that is being edited.



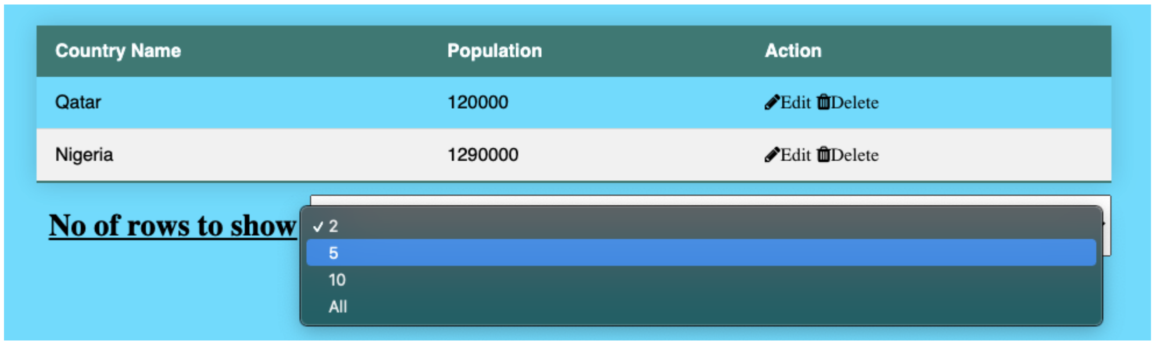
Population Census

Country Name	Qatar
Population Size	120000

Countries Census

Country Name	Population	Action
Qatar	120000	Edit Delete
Germany	140000	Edit Delete

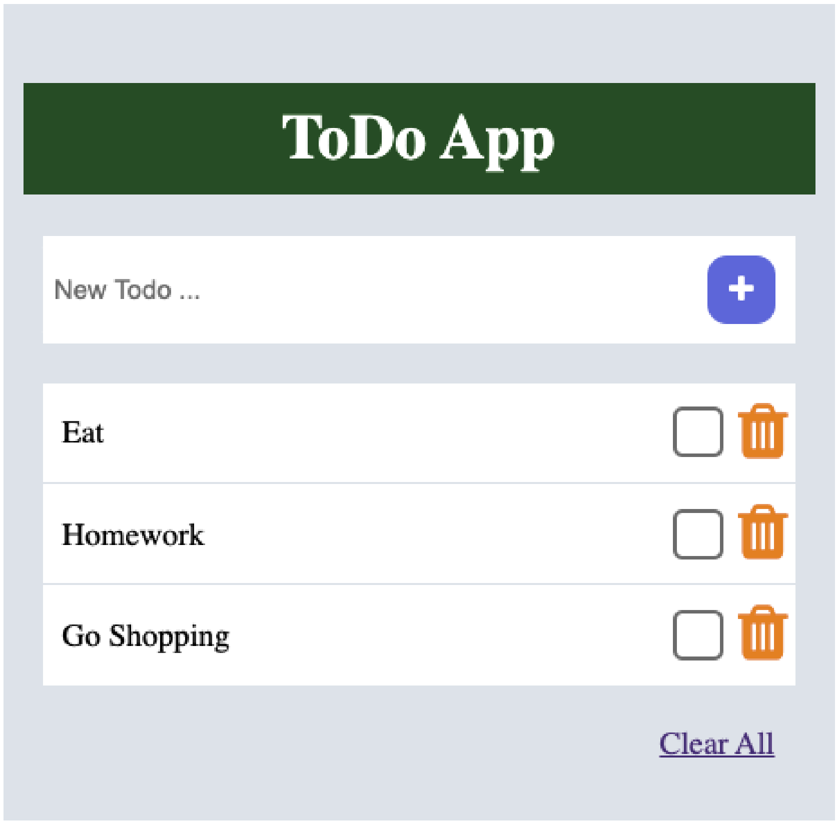
f. When the user clicks on the delete icon, delete the document from the collection and the table.
g. Add a way to filter the total number of records to show at a single time. You should use the limit method to limit the number of records returned from the database.



Country Name	Population	Action
Qatar	120000	Edit Delete
Nigeria	1290000	Edit Delete

PART B – Todo App

Design and develop a Todo web app using HTML, CSS, and JavaScript. Enhance the app to store and retrieve the todo tasks using **IndexedDB**.



PART C – Local Image Gallery

Create an application that allows the user to upload multiple images, stores those images persistently using IndexedDB, and renders those images as a gallery using a RAM layout. Clicking on an image in the gallery should remove it from the list.