

# CMPS 350 Web Fundamentals - Spring 2021

## Lab 8 – Client-side JavaScript

---

### Objective

In this Lab, you will **build a Census App** and a **Todo app 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

### Project Setup

1. Sync “Lab8-Client JS Data Management” from the Lab GitHub Repo and copy it to your repository.
2. The folder contains both the TodoList 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 webstrom and try to implement the following application.

The screenshot shows a web application titled "Population Census" with a light blue background. It features two main sections: "Population Census" and "Countries Census".

**Population Census Section:**

- Labels: "Country Name" and "Population Size".
- Inputs: Two text input fields corresponding to the labels.
- Buttons: "Add" and "Clear" buttons.

**Countries Census Section:**

- Table with 3 columns: "Country Name", "Population", and "Action".
- Table Data:

Country Name	Population	Action
Qatar	120000	<a href="#">✎ Edit</a> <a href="#">🗑 Delete</a>
Germany	140000	<a href="#">✎ Edit</a> <a href="#">🗑 Delete</a>

**Footer:**

"No of rows to show" followed by a dropdown menu currently set to "2".

1. Add the following script tag to your code in order to get the Localbase library  

```
<script
src="https://unpkg.com/localbase/dist/localbase.dev.js">
</script>
```
2. Create the census indexDB database inside the index.js file  

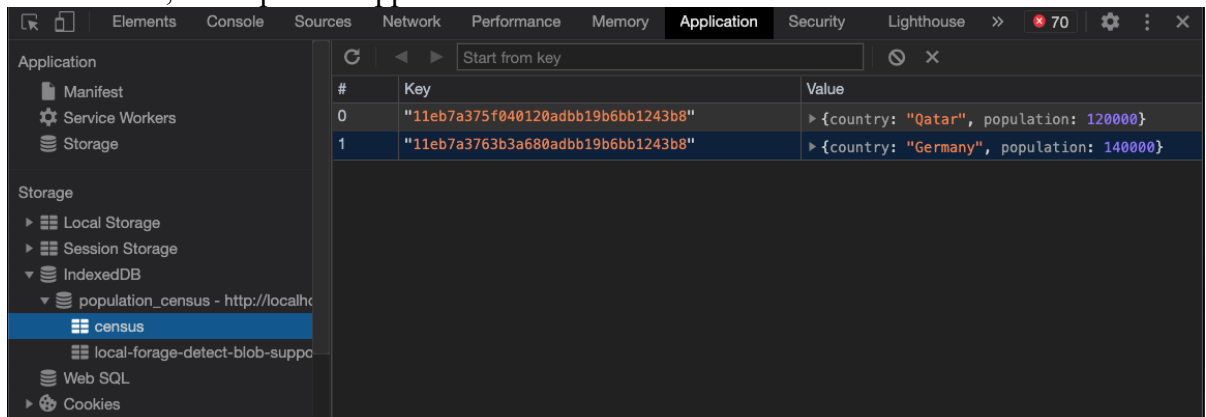
```
const db = new Localbase('census.db')
```
3. Implement the following functionalities
  - a. When the users click on the **add button**, you should
    - i. call a function called addCensus that receives the form,
    - ii. convert the form into a JSON object
    - iii. add the JSON object into a 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, then open the application tab



- c. If the user clicks on the clear, 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	<a href="#">✎Edit</a> <a href="#">🗑Delete</a>
Germany	140000	<a href="#">✎Edit</a> <a href="#">🗑Delete</a>

- 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**

**Population Size**

Add
Clear

**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

**No of rows to show**

✓ 2

5


10

All


## PART B – TodoApp 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**.


# ToDo App




Eat

☐ 

Homework

☐ 

Go Shopping

☐ 

[Clear All](#)